SECURED ELECTRONIC HEALTH RECORD MANAGEMENT PROTOCOL

SYEDA FARZANA
ID: 012142014

A Thesis

in

The Department

of

Computer Science and Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of Master of Science in Computer Science and Engineering

United International University

Dhaka, Bangladesh

July 2018

Approval Certificate

This thesis titled " **Secured Electronic Health Record Management Protocol**" submitted by **Syeda Farzana**, Student ID: **012142014**, has been accepted as Satisfactory in fulfillment of the requirement for the degree of Master of Science in Computer Science and Engineering on **14.07.2018**.

Board of Examiners

1.
_____ Supervisor
Dr. Salekul Islam
Professor & Head
Department of Computer Science and Engineering
United International University (UIU)

2.
_____ Head Examiner
Mohammad Mamun Elahi
Assistant Professor
Department of Computer Science and Engineering
United International University (UIU)

3.
_____ Examiner-I
Novia Nurain
Assistant Professor
Department of Computer Science and Engineering
United International University (UIU)

4.
_____ Examiner-II
Mohammad Moniruzzaman
Assistant Professor
Department of Computer Science and Engineering
United International University (UIU)

5.
_____ Ex-Officio
Dr. Mohammad Nurul Huda
Professor & Coordinator - MSCSE
Department of Computer Science and Engineering
United International University (UIU)

Declaration

This is to certify that the work entitled "**Secured Electronic Health Record Management Protocol**" is the outcome of the research carried out by me under the supervision of Dr. Salekul Islam, Professor & Head, Department of Computer Science and Engineering, United International University (UIU), Dhaka, Bangladesh.

_____

Syeda Farzana
Student ID: 012142014
Department of Computer Science and Engineering
Master of Science in Computer Science and Engineering (MSCSE) Program
United International University (UIU)
Dhaka, Bangladesh

In my capacity as supervisor of the candidate's thesis, I certify that the above statements are true to the best of my knowledge.

_____

Dr. Salekul Islam
Professor & Head
Department of Computer Science and Engineering
United International University (UIU)
Dhaka, Bangladesh

Abstract

In our progressive era of technology, digitization is becoming the link between businesses, people, processes, and data. Internet & mobile devices have allowed us the access to almost anything we want to know at the tip of our fingers, connecting us to home, school, work, or local library etc. with just a few clicks. Digitization is evolving with time and profoundly widening access by lowering the impediment to discovering information which is geographically scattered. A correctly digitized collection expands access, protects the safety of the data and provides new options for research. Proper digitization and distribution is essential in the way data are accessed and protected because digitizing allows new infiltrations and discoveries by not just the select few with physical access to the physical data but by any viewer from far away. Appropriate digitization can both directly diminish degradation of the original physical collection, as well as offer the content of collection in the case of any physical loss. The important documents can be stored safely in the repository for the data, shared on cloud or local document management system, and recovered with a simple click. Digitization makes it easier to manage arrangements for all activities within an organization. Application of digitization integrated internally includes sectors such as Health care, Education etc. The users could use self-select options and self-enter data using web-based interfaces allowing them to be able to manage their own data through controlling the structure

Since digitization is now a common practice for storing and retrieving data, Electronic Health Record (EHR) management is becoming very popular. EHR management will bring many benefits including easy to store, cost effective, shareable with health professionals to a remote location, etc. However, EHR stores very sensitive data and thus, a number of security properties including privacy, secrecy, integrity, authenticity of data and availability must be ensured during data transmission, storing and sharing with health professionals. In this study, we have studied symmetric key based technique in designing EHR management protocol. In light of the existing efforts, we have developed a simple protocol for secured EHR management. A simple symmetric key based EHR management protocol that we validated using AVISPA, an industry-strength security protocol validation tool. Even though our proposal is primarily based on symmetric key, it identifies the doctors using their attributes [1][2] AVISPA has confirmed our protocol free from known attacks and confirmed the desired security properties as well.

Acknowledgement

Foremost, I would like to express my sincere gratitude to The Almighty Allah, The most gracious, The most merciful. I could not have reached the place where I am today without His blessings and guidance.

Being a student at United International University (UIU) for my Undergraduate degree as well, I was well aware of their endeavors for "Quest for Excellence" and how each entity that is a part of UIU strives to reach that goal; how that act inspires others to bring out the best in them to try to become a better person. I first came across their sincere cooperative behavior during my first semester in Undergraduate program when due to some error regarding my ID I was halted from receiving a scholarship which I was eligible for. Each and every individual, from administrative staff to my course advisor, who was in charge of such matter, guided me in such an earnest way that I was able to resolve the issue swiftly than I could have expected.

Even still, I was a bit worried during my admission in M.Sc. in Computer Science & Engineering (MSCSE) about whether I could cope up with a Master's degree on a different subject switching from an Electrical and Electronic Engineering Undergraduate background. But soon I realized that my doubts were all unfounded. Even though we were at Master's level, the faculties at the Department of Computer Science & Engineering at UIU paid extra attention, care and patience in explaining every topic in great details just to make it easier and understandable for those who might be coming from a different Undergraduate background. The way the faculties treated us reminded me constantly of how a parent guide their children in every step of their life making sure that they get the supreme lessons and becomes capable to give their best to the world. My genuine unceasing appreciation goes to all the faculties of United International University.

I cannot express in mere words the entirety of the gratefulness I feel for the help and patience my supervisor Salekul Islam, Ph.D. offered me in response to my incessant queries and questions. He imparted his expert knowledge transforming them to a form that was uncomplicated for me to grasp and directed me the best procedures to successfully conclude the aim of my thesis. I admire with gratitude how he broke down any complicated problem into stages and efficiently explained each stages to clear any

doubt in any subject he taught and also during the guidance of my thesis work. I truly feel obliged to have been able to do my thesis under the supervision of such a generous, experienced and compassionate faculty.

I am extremely thankful to Md. Nurul Huda, Ph.D. for guiding and advising me during course selection, letting me know which subjects would be best and easier for me to understand clearly, having lacking the strong base which is built in the Computer Science & Engineering Undergraduate program. His teaching method in classes is a boon that I feel privileged to have received as a student. The way he repeatedly explained a topic with patience if a single doubt is present in any student's mind is truly something I am astonished and grateful for at the same time.

Furthermore, I would like to whole heartedly express my thanks and unpretentious appreciations to my friends. I am beholden to them for painstakingly, sacrificing their time, taking the job upon them to boost and revive my energy by emboldening me to work harder and refocus on my thesis whenever I lacked enthusiasm, inspiration or lost track. I am humbly grateful and feel blessed to have them in my life, for being my strength and for them being there for me whenever I needed it the most.

Last but not the least I would like to thank my parents and each and every member of my family for being my pillar, for always cherishing me, bestowing me with their blessings, love and encouragement throughout every tide and falls of my life. Without their thoughtful teachings regarding ethics, values, ideals and morals concerning right and wrong, I could not have become the person I am now. Nor could I have attained the unwavering concentration needed for completing my thesis without their infinite support.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

Internet is becoming increasingly inexpensive due to services like broadband which are expanding their reach to more users gradually. Adapting to digitization is appearing beneficial thanks to its profound advantages. Features like easy to stay connected with individuals, instant communication from one corner of the world to another, cloud based services (e.g., storing and retrieving data using Cloud computing and Cloud storage), sharing of Information and knowledge, online learning, online shopping, collaboration, work from home, access to a global workforce, video conferencing are just few examples. As Internet is becoming an intricate part of our life, it is now a common practice for using its benefits and advantages to our day to day use to make our life easier. Technology and the digital revolution is bringing benefits in many ways by providing a platform that connects people from across the globe while updating the status of transactions as they happen online. This might require individuals to share their private information demanding the data to be kept secured in a simple yet efficient way.

Digitization is the conversion of data into a digital format with the adoption of technology. This allows saving time and improving the efficiency by capturing documents and data at the point of origin, reducing transcription errors, utilizing security protocols (enhanced security could be applied at the document, folder, person, position, or workgroup level and only certain users can access them that matches with the permission groups maintaining the confidentiality of the document), refining availability to information. A proper digitization process uses hardware, software, and workflow highlighting the safety of the data. The process might use scanning to capture the document or electronic forms to digital capture the information or any other technique. Digitization brings advantages like no physical bounds for storage, easy access by the use of the Internet, 24/7 convenience of access (data are easily accessed through the Cloud or any other system using any device that has Internet, anywhere or anytime), preservation of old scripts or manuscripts, easy recovery of information using keywords, online resource distribution (increasing productivity and

efficiency due to its ability to share, collaborate, exchange and access documents in seconds), linking and networking options, reducing human error etc.

Digital solutions involving security protocols could be applied to strengthen a country's security, intelligence systems, public services, infrastructure etc. Developed countries are increasingly engaging various methods and technology involving digitalization to fully grasp what capability they need and execute strategy accordingly. Its benefits indicate digitization will continue to evolve and have improvement. Digitization has become essential for organizations as it can be used to embark on continuous improvement, allowing elimination of barriers for organizations which are geographically dispersed, permitting transparent sharing of cross-organizational information. This makes governance possible at regional and multinational level rather than at local level only.

Application of digitization includes sectors such as Health care, Education where the users could use various options allowing them to be able to manage their own data including authorization processes, by creating a relationship to other digitized data. This ensures efficiency and transparency between a user and an organization.

Among the many application of digitization in Health care sector, Electronic Health Record (EHR) management is becoming very popular. EHR is any health related information or data of patient stored digitally using electronic methods. EHR involves the transfer of health related data through electronic media, e.g., the Internet [3]. This information can only be accessed instantly by authorized users, e.g., doctors or the patients themselves, maintaining top notch security to assure privacy. EHR contains sensitive information including patients' medical and treatment histories, administrative and billing data, prescriptions provided by the doctors, medical test reports, personal information (e.g., age, weight and height), etc. As EHR contains valuable information, security and privacy is the prime concern. Therefore the need for privacy, confidentiality and security [4][5] is inevitable as essential components between healthcare consumers and providers. Such properties are essential not only for patient health information but also for medical care, investigation, payment, and healthcare policymaking. The EHR system can also be created to go further than typical clinical data collected in a medical institution and can add various other facilities to allow a broader view of a patient's care [6]. For example it can allow access to tools to help doctors to make decisions about a patient's care. As EHRs are stored

digitally they can be easily shared from one place to another through intra connected private networks, e.g., from one branch of a hospital to another distant branch [7]. Since they are stored digitally they are able to reduce the incidence of medical error as a result of refining the correctness and precision of medical data. Moreover the availability of all up to date health related information of a patient in one place prevents duplication of tests, helping to speed up the process of diagnosis or treatment. Even if the patient is unconscious the emergency department can sought out the EHR of the patient to take necessary steps and help the patient to recover from life threatening condition. On the other hand a patient can get motivated to organize his lifestyle accordingly when visually perceiving the trend of the lab results over the medication period by logging on to his own record.

EHR management will bring many benefits including easy to store, cost effective, shareable with health professionals to a remote location, etc. However, extra care must be taken to ensure its proper deployment since EHR stores very sensitive data. Thus, a number of security properties including privacy, secrecy, integrity, authenticity of data and availability must be ensured during data transmission, storing and sharing with health professionals.

Thus various cryptographic protocols that use cryptographic techniques have been developed to achieve secured management of EHR. Two popular methods are found in the literature to secure EHR: symmetric key encryption [8][9] and Attribute-Based Encryption (ABE). Since no public keys are involved, symmetric key based systems are less expensive to implement and maintain. The concept of ABE was first introduced in [10] [11] to store and share encrypted data without using symmetric key. In ABE, if A encrypts data using K_A, B can decrypt this data using K_B, as long as the identities of A and B are close to each other. Here, identities are considered as a set of descriptive attributes, and thus it was termed as Attribute-Based Encryption (ABE). Recently, different efforts have been carried out to use ABE in securing EHR management system [1][2].

In our research, we have developed a simple symmetric key based EHR management protocol. Although our proposal is primarily based on symmetric key, it identifies the doctors using their attributes [1][2]. The security properties of this protocol have been validated by modeling the protocol using Automated Validation of Internet Security

Protocols and Applications (AVISPA) [12]. AVISPA is an automatic tool with industrial strength technology for the investigation of different Internet security protocols and applications. It is being used by the developers of diverse security protocols and by academics equally [13][14]. In our validation, AVISPA testified the proposed protocol free from attacks.

1.1 Online Data Storage

Storing electronic information with a third party provision which can be accessed using the Internet is called “Online data storage”. It can also be termed "hosted storage", "Internet storage" or "Cloud storage" [15][16].

The number of vendors suggesting online data storage has increased recently. Different services store different type of data. Some allow storage of only a particular kind of data, such as photos, music or backup data, while others might allow storage of any type of file. One of the biggest advantages of online storage is the capability to access data from anyplace. Thus syncing or transferring data among devices has become more imperative. Online data storage provides help in not only transferring data between devices, but also the ability to share data among different users easily.

Normally on-site storage (i.e. local storage or portable storage, in the form of tapes and floppy disks, CDs, DVDs, USB thumb drives etc.) is faster than using Internet storage as there is no delay for files to upload or download. But, on-site storage is prone to loss due to theft, natural calamities or device malfunction. It offers limited storage capacity and is also not quite as convenient as online data storage if we want to share files with a large number of users. As it is located off site, online data storage could offer backup during disaster recovery situations e.g., fire, flood, earthquake or similar situation, when on-site backups could be ruined. Most online data storage services offer reliability in the form of enhanced physical security, enhanced data protection, automated backup capabilities, availability as well as easier data transfer and sharing. It has become a service sold on demand, provide elasticity (giving the user as much as they want) and present self-service options.

Many establishments use a combination of on-site and online storage abilities according to their need. For example, in Health care sector, they might use local storage for files they

use often or for personal data and online storage for backup, archive data and all of the patients' data they wish to share with others.

1.2 Literature Review

We studied various existing protocols developed for EHR present today to obtain a clear understanding and come up with materials with which we can contribute in the security techniques for EHR systems that is in par with the current requirements. The vast researches in this field have taken various diverse approaches to ensure the security features of EHR. Different researches have even focused on different specific part of the total EHR system. Some have worked with a small focus modeling and testing a proposed protocol that confirms only secrecy property or authentication property or both or more security property requirements altogether. While others have extended their work with a deeper evaluation study and actually verified the systems and tools within healthcare systems to examine if it supports the real life implementation or integration [17]. We congregated research works that is similar to our pursuit of EHR involving AVISPA for verification. We came to the conclusion that AVISPA has been used in many other study works involving various protocol verifications [14][13][18][19] and EHR research work also has many works [20][10][2] on its own but researches involving the two together are not many.

Work that deals with the secrecy property of EHR which we focused in our proposed work is [21].This paper ensured the secrecy property of EHR to prevent the revelation of highly sensitive health records to unauthorized persons by using pseudonymization method that sustains the patient's privacy and data confidentiality. The basic idea it provided is that many depersonalized (i.e., the health data is separated from identifying information of the patient) medical records alone lacks to uniquely identify the patient. Hence both records and patients' information are given randomly-selected pseudonyms. These pseudonyms act as access tokens which allow relinking of the health data to the matching patients. The pseudonyms are secured by encryption with a user-specific secret key. The authorization model is patient centric. The patient is labeled as the data owner who maintains full control over his or her health data and is able to state access authorizations for trustworthy persons to particular health records specified by the patient. Micro controller smart-cards with integrated crypto chips are used for secured authentication.

Both symmetric key and asymmetric key encryption are used in the model design to provide authentication and authorization layers. Only users with the possession of the correct security tokens are able to transcend each layer to finally decrypt the original data. The shared pseudonyms are encrypted with both the patient's and the health care provider's inner symmetric keys. The complete approach is validated by involving the verification of the correctness of the PIPE pseudonymization protocol using the AVISPA tool and also practically validated by developing a prototype, which is implemented in a medium-sized firm offering predictive genetic testing. The paper concluded that even though pseudonymization is a encouraging technique to fulfill the requirements of data storage, access as well as privacy-preserving use, but in general requires a sufficiently large number of individuals and records to be effective. Moreover it emphasized the fact that successful pseudonymization requires trustworthy depersonalization, which can be quite difficult, for certain kinds of health data. Our work deviated from this paper in ensuring secrecy by utilizing symmetric key encryption solely.

In [22], a secure and useful remote user authentication scheme for connected health care, which achieves uniqueness and anonymity properties is proposed. In this scheme after successful authentication, a symmetric secret session key is established between the user and the server so that they can use that key for their forthcoming secure communications. Security analysis and simulation for the formal security verification is carried out using AVISPA tool to ensure that the scheme is secure against possible passive and active attacks. In our work although symmetric encryption will be used, an indirect authentication of the doctor will be performed by the key server based on the doctor's attributes instead.

1.3 Motivation

Electronic Health Record (EHR) requires extra caution due to its sensitive nature, before its proper deployment physically. A number of security properties including privacy, secrecy, integrity, authenticity of data and availability must be ensured theoretically as well as tested. Various software are available nowadays specifically for this purpose to simulate the scenario and deduce the fault during data transmission, storing and sharing with health

professionals, as well as testing the security aspect to ensure it is working effectively as hypothesized.

As complexity of technology is increasing day by day by adding richer features, demand of more fault proof complex security protocols for EHR is increasing. More and more researches are carried to implement in an EHR to ensure safety and top quality as Hackers may still be able to penetrate EHR system despite various security precautions, harming people by releasing the confidential information to others. Thus, after observing and studying many existing protocols developed for EHR, we want to be a part of this immensely necessary research and contribute in helping EHR to take a step further in its research field. We have come up with developing a symmetric key-based EHR management protocol that has successfully introduced the attribute-based access control in symmetric-key solution. Earlier we came up with a work [23] that is slightly different in incorporating the attribute set than our approach present in this proposed protocol here. [23] only focused on successful data storage and inspired us to delve further with such concept to provide a protocol that involves a complete process of both data storage and data retrieval. Although the protocol we have developed is simple and has not considered many complexities that may arise during deployment in a real-life world, but to our knowledge this is the first AVISPA model of symmetric key-based protocol that also adds attributes. In future, this model can be extended to validate other complex EHR management protocols.

1.4 Thesis Organization

The rest of the thesis is organized as follows:

Chapter 2

In this chapter we described a general Electronic Health Record Management Architecture and proceeded to explain what features or cryptographic protocols are commonly employed in it and are thought crucial for an optimum EHR Management Architecture. In trying to enlighten the features further we have delved into an exploration about the classification or various types of cryptographic protocols that are found in existing EHR Management Protocols, which we came across during our research is presented in this chapter.

Chapter 3

This chapter covers the background information needed about AVISPA (the tool which we would use to validate our researched model) and how it functions.

Chapter 4

This chapter explains our Proposed Electronic Health Record (EHR) Management Protocol with all its requirements. The protocol is broken down in details using message sequences exchange we want for a successfully operational EHR Management Protocol.

Chapter 5

In this chapter we finally implemented our hypothesized Protocol in AVISPA using HLPSL protocol specification and validated the output of our model using a graphical tool, SPAN. Message sequence charts (MSC) provided by SPAN aided us in our analysis to conclude our research by validating our Proposed Electronic Health Record (EHR) Management Protocol.

Chapter 6

In this chapter we summarized the thesis, discussed its findings and contributions, limitations and also outlined guidelines for future research. The chapter is divided into two sections where the first section sum up the thesis along with a discussion about the contribution of the current work. Last section concludes the thesis discussing the future work.

Chapter 2

Background and Related Work

Electronic Health Record (EHR) is any health related information or data of patient stored digitally using electronic methods. This information can only be accessed instantly by authorized users, e.g., doctors or the patients themselves, maintaining top notch security to assure privacy. EHR contains sensitive information including patients' medical and treatment histories, administrative and billing data, prescriptions provided by the doctors, medical test reports, personal information (e.g., age, weight and height), etc. As EHR contains valuable information, security and privacy is the prime concern. Since they are stored digitally they are able to reduce the incidence of medical error as a result of improving the precision and transparency of medical records. Moreover the availability of all up to date health related information of a patient in one place prevents duplication of tests, helping to speed up the process of diagnosis or treatment. EHR involves the transfer of health information through electronic means, including the Internet [3]. Therefore the need for privacy, confidentiality and security [5] is inevitable as essential components between healthcare consumers and providers. Such properties are essential not only for patient health information but also for clinical care, research, payment, and healthcare policymaking. Thus various cryptographic protocols that use cryptographic techniques have been developed to achieve secured management of EHR. Initiatives like Integrating the Health-care Enterprise (IHE) [24] have been formed to establish the definition of standard methods for secure and interoperable EHR exchanges. The European Commission has also issued a mandate for applying the adoption of EHR systems and the US government has also published the Health Insurance Portability and Accountability Act (HIPAA) for the development of standard methods to attain a secure and interoperable EHR exchange among clinics and hospitals similarly. Using the stated guidelines by these initiatives, many extensive projects have been set up to enable healthcare professionals to handle patients' EHRs. Our proposed protocol has followed [25][26] which provided a guideline in compliance to HIPAA standards for designing an EHR system. The goals of HIPAA [27] are to protect health insurance coverage for workers and their families when they change or lose their jobs (Portability) and to protect health data integrity,

confidentiality, and availability (Accountability). These major things are addressed in an arrangement of two parts,

- Part I: Health Care Access, Portability, and Renewability. Protects health insurance coverage when someone loses or changes their job. Addresses issues such as pre-existing conditions.

- Part II: Administrative Simplification.

We focused on the part of HIPAA [26] regulations that cover both security and privacy of protected health information. The physical security of patient's health information in all formats is an element of the Privacy rule in HIPAA. The HIPAA Privacy Rule requires that covered entities (e.g., doctors or medical administrator) apply appropriate administrative, technical, and physical safeguards to protect the privacy of protected health information (PHI), in any form. The HIPAA Security rule on the other hand establishes national standards specifically to protect individuals' electronic personal health information that is created, received, used, or maintained by a covered entity. In short, HIPAA rules aim to protect the confidentiality, integrity, and availability of health information in EHR.

It is provided in [25] that providers and individuals alike must trust that an individual's health information in Electronic Health Records (EHRs) is private and secure and that the confidentiality and accuracy of their electronic health information is not in jeopardy. Poor privacy and security practices increases the exposure of patient information in health information system, strengthening the risk of successful cyber-attack.

In order to encourage patients' trust, the requirements directed in [25] are:

R1: Maintain accurate information in patients' records

R2: Make sure patients have a way to request electronic access to their medical record and know how to do so

R3: Carefully handle patients' health information to protect their privacy

R4: Ensure patients' health information is accessible to authorized representatives when needed.

2.1 EHR Management Architecture

EHR management architecture basically means the model that the system should follow to make the storage and transfer of EHR feasible. Many different approaches have already been carried out and more are being produced to make the system advanced having more security measures and enhanced upgrades in par with the always improving latest technologies. A general and very basic EHR management architecture is shown in Figure 1. The structure basically depicts that the EHRs are stored centrally by the authorization of a patient. Doctor refers to any health professionals. The patient updates or stores the medical records after receiving treatment from doctor-1. The patient could also give authorization to doctor-1 to store the medical records on behalf of the patient. This same patient might be referred to doctor-2 for further medical support. Doctor-2 can view the patient's previous record after the patient grants him access to data at the EHR server. Thus doctor-2 can easily check the patient's past medication history and provide further help without any kind of hassle of missing data or records. The above model is modified in various ways to suit the need of protocols or designs, applied in various works, to enhance the structure and meet the objectives. For example [1] shows the incorporation of encryption-decryption function along with transaction code service (TAC) and private key generator (PKG) to provide flexibility along with the security measures to data which was the objective of that paper.

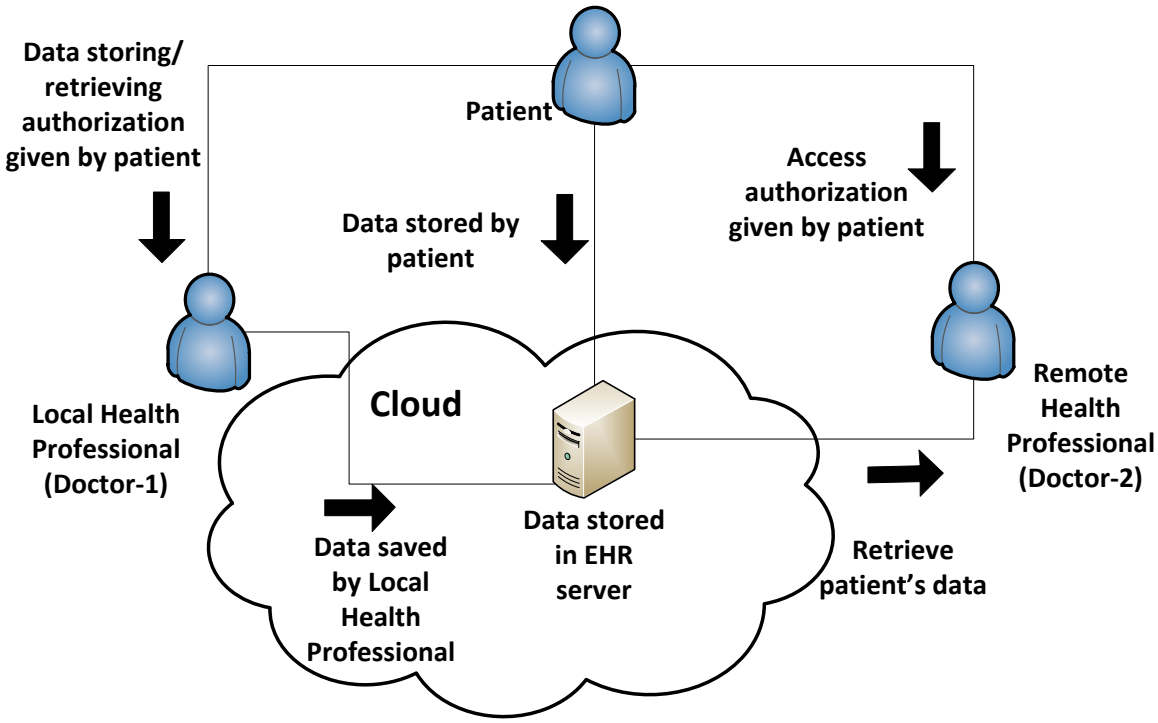

Figure 1: General EHR Management Architecture

2.2 Required Security Properties

We summarize the required security properties that generally a secured EHR management should support.

Privacy and Secrecy: Patient privacy and secrecy is crucial while records on remote servers leave the data open to security exploits and data theft [28]. Encryption is the mechanism, in general, applied to ensure secrecy [1]. Encryption converts the original message or information into encoded text and is only decrypt-able or is readable by authorized persons only. Moreover account security is defined by secure passwords and strict user access levels. Secure password, which allows access to the EHR account, is indispensable to ensure that information does not fall into the wrong hands.

Integrity: Integrity, internal consistency, and accuracy of information in the patient's EHR are imperative to ensure accuracy of the complete health record [29]. It involves information such as patient identification, authorship validation, amendments and record

corrections. As wide variety of data is collected in healthcare it must be collected accurately, completely, and consistently and ensure documentation integrity to avoid wrong information documented on the wrong patient health record. It is important to guarantee that appropriate care and billing activity is subjected to the correct patient.

Access Control: User authentication is required to determine whether someone is, in fact, who it is declared to be. The purpose of authentication is to allow authorship and assign responsibility for an act, event, opinion, or diagnosis made by the doctor [29]. Entries in the healthcare record should be authenticated by the patient [4]. Patient can permit or reject sharing their information with other healthcare practitioners [1]. To implement patient consent in a healthcare system, patient may grant permissions to users on the basis of a role or attributes held by the respective user [5]. The access and sharing of EHRs could be provided by end-to-end source confirmation through signatures and certification.

Availability: It is essential for any EHR system to be available and make the information accessible when it is needed [29]. This means that the computing systems used to store and process the EHR data, the security controls used to protect it, and the communication channels used to access it must be operating properly all the time and are reachable.

2.3 Cryptographic Protocols

Protocols are defined as the algorithmic steps or methods applied to implement a program. Security means to provide protection from risk, danger or crime. Therefore with respect to technology security simply means to take measures using technological processes or approaches to provide protection of data or information or a system against threat or attack from malicious sources. The security is usually provided using cryptography. Thus a security protocol, appropriately termed as cryptographic protocol, basically entails a series of steps based on a set of rules which includes exchanges of various messages, both encrypted and non-encrypted, among multiple parties taking part in the system, to achieve the security goal. Attack from intruder is not limited to data only and can be on the applications, middleware, network and protocol stacks or hardware. Based on the type of security intended to be applied in accordance to the goal or objective, various security properties can be taken into consideration e.g., secrecy, authentication, confidentiality, access control, integrity, non-repudiation, anonymity, fairness, certified delivery etc. The general model usually applied in cryptographic protocol is given in Figure 2 below.

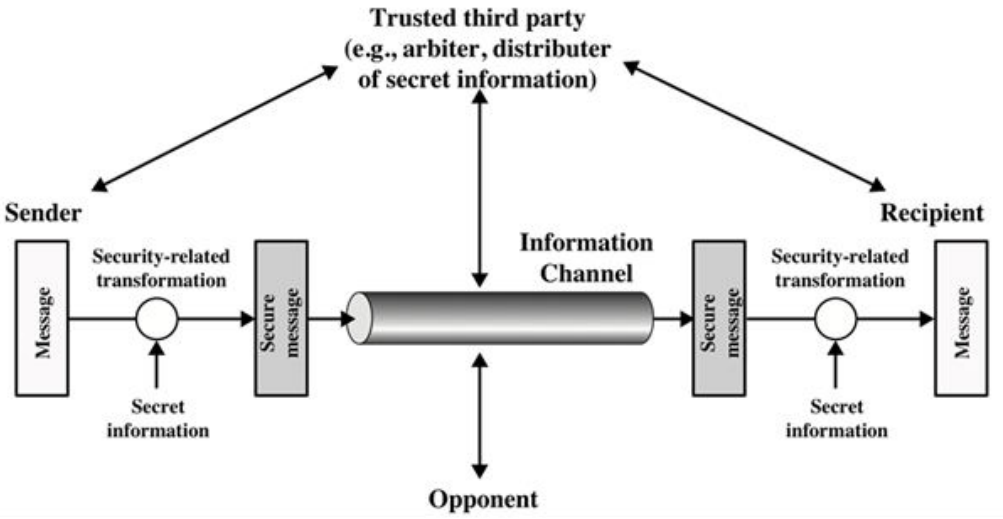

Figure 2: A general model using cryptographic protocol to provide security [30]

The various types of cryptographic protocols that are employed are secrecy protocols, authentication protocols, key distribution protocols, e-commerce protocols etc. to name a few. They have key agreement or establishment, entity authentication, symmetric encryption and message authentication, secret sharing methods etc. features in them.

The popular methods found in the literatures for encryption [31] :

- Symmetric key encryption
- Public key encryption

2.3.1 Symmetric Key Encryption

Symmetric key encryption is a kind of encryption in which the sender and receiver uses the same cryptographic key to encrypt and decrypt the message. Since no public keys are involved, symmetric key based systems are less expensive to implement and maintain. The general structure of symmetric key encryption is given in Figure 3.

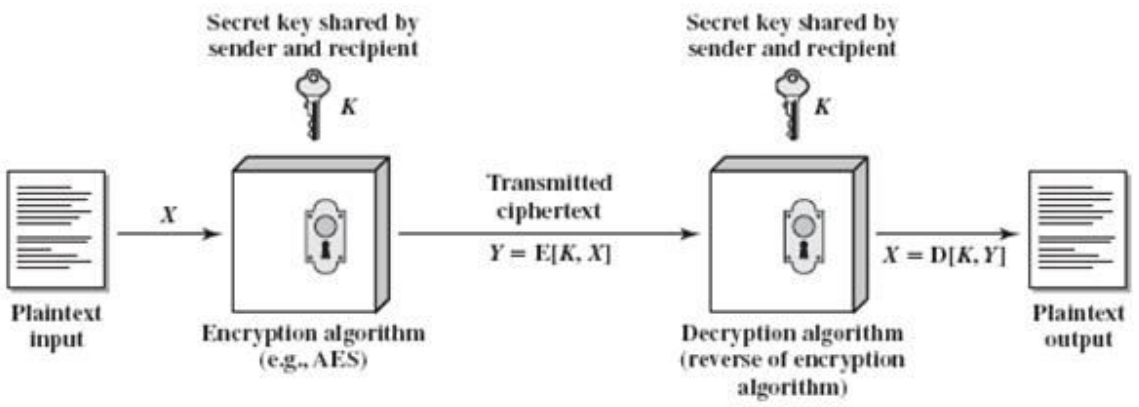

Figure 3 : Symmetric key encryption [32]

where,

X = plaintext message

K = symmetric key used for encryption

E = encryption algorithm applied to obtain ciphertext, a random incomprehensible data

Y = transmitted ciphertext over the network from sender to receiver

D = decryption algorithm applied to obtain original plaintext

2.3.2 Asymmetric Key or Public Key Encryption

In asymmetric key encryption there is a pair of keys involved: a public key and the other a private key. The private key is only known by the sender and is used to encrypt the message. The public key is provided by the sender to the receiver and to anyone in general who are expected to receive and decrypt the message with it. In asymmetric key encryption only a public key can decrypt the message encrypted by the private key. The general structure of asymmetric key encryption is given in Figure 4.

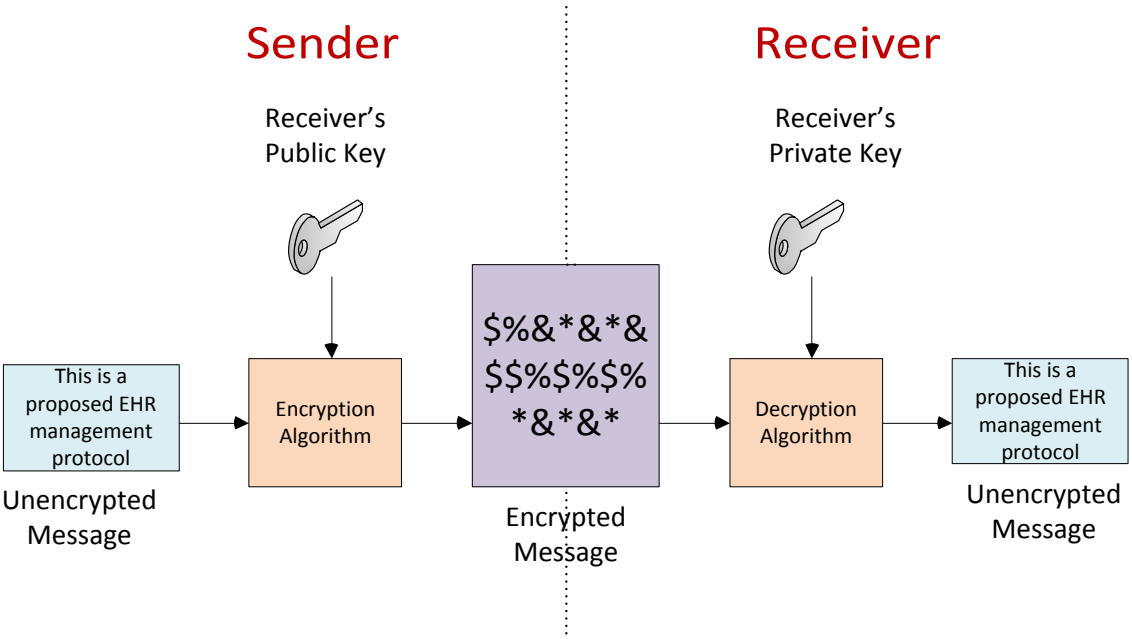

Figure 4: Asymmetric key encryption

2.3.2.1 Public Key Infrastructure (PKI) and Certification

Public key infrastructure (PKI) is a needed set of technical mechanisms, rules, and measures whenever digital certificate come into play.

The PKI allows the use of secure Internet applications activities such as e-commerce, Internet based transactions like internet banking and confidential email. In such activities simple passwords are insufficient authentication method and it is an essential requirement that the identity of the parties involved in the communication and validation of the information being transmitted is confirmed [33].

PKI facilitates an arrangement that binds public keys with respective identities of entities with the help of registration and issuance of certificates by certificate authority (CA). PKI helps to create, manage, distribute, use, store, and revoke digital certificates and manage public-key encryption. PKI consists of two elements; Public Key Cryptography and Certification Authorities [34]. A public key infrastructure (PKI) is a scheme for the creation, storage, and distribution of digital certificates which are used to validate that a particular public key belongs to a certain entity. The PKI creates digital certificates which map public keys to entities, securely stores these certificates in a central repository and rescinds them if needed.

A certificate authority (CA) stores, issues and signs digital certificates. Certification authority (CA) [35] is a trusted (both by the owner of the certificate and by the party trying to verify signatures)third party or entity that issues digital certificates which verifies the ownership of a public key by the named subject of the certificate. A verifier is an entity that seeks out the digital certificates issued by the CA to confirm that a particular public key belongs to a certain entity. The Figure 5 below depicts a general structure of PKI.

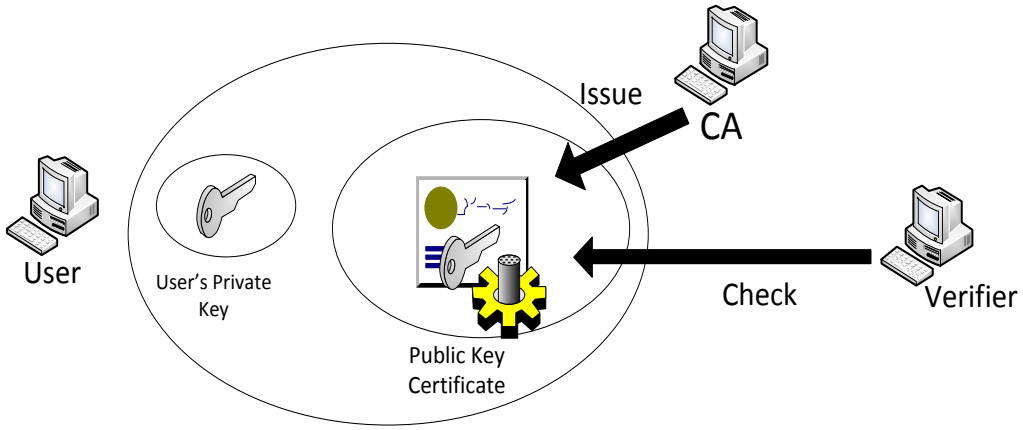

Figure 5: A general structure of Public key infrastructure (PKI)

2.3.2.2 Attribute Based Encryption Using Asymmetric Key

Attribute means property or features of an object, element, or data. It may also be considered as metadata. In Attribute Based Encryption (ABE) using asymmetric key encryption, a pair of cryptographic keys, public and private, are used to encrypt and decrypt the message.

2.3.3 Past Works Involving Symmetric Key or Asymmetric Key Encryption

Many research works have been done using symmetric key encryption and asymmetric key encryption. Below we have discussed two of them as examples.

2.3.3.1 SiRiUS: Securing Remote Untrusted Storage [9]

Several security problems are introduced if the trust in storage server is eliminated. End-to-end security is compulsory, including data secrecy, data integrity, authenticity and access control. SiRiUS is one of the existing solutions that take on these challenges relying greatly on the use of public-key cryptography. Only file or block encryption is done with

symmetric-key algorithm. In this system, the data is encrypted by authors before it is sent to the server, and decrypted by readers after it is received from the server. SiRiUS is a cryptographic file system that enables secure file sharing over untrusted servers. However, disregarding the trust in the server comes at the cost of performance.

This paper presented a secure file system design by using cryptographic storage to be layered over untrusted network to secure block and file-level remote storage. SiRiUS presumes the network storage is insecure and builds a secure file system on top of it that offers its own read-write cryptographic access control for file level sharing. It operates as an add-on that does not change the underlying file system. Confidentiality is obtained by encrypting the contents of data that are stored on untrusted file servers. The server operators deliver the encrypted files without knowing the actual plaintext files themselves.

Since end-users often have no control of the remote server, therefore, no changes to the file server enables to enrich the security of legacy network file systems without changing the existing infrastructure. SiRiUS handles multi-user file systems where users often share files and supports permitting read only or read-write access to files. Cryptographic operations like encryption and signing are done by the client before anything is placed on the file server. All SiRiUS users maintain one key for asymmetric encryption and another for signatures. These are the user's master encryption key (MEK) and master signing key (MSK).

Files stored on the file server are kept in two parts. One part contains the file meta data (referred as *md-file*) and the other contains the file data (referred as *d-file*). The *md-file* holds the access control information while the *d-file* contains the encrypted and signed contents. The file data is encrypted with a symmetric cipher encryption key called the file encryption key (*FEK*) and is signed with a unique key for that file called the file signature key (*FSK*). Each file has its own *FEK* and *FSK* keys that are generated by the owner when the file is created. These keys are encrypted using the public keys of any user to whom the owner provides access permission and are saved as part of the *md-file*. When a user needs an access to a file he reads the relevant *md-file* and decrypts the *FEK* and *FSK* using his private keys. The *md-file* is also signed with the owner's private key, thus the user needs also to get the owner's public key to verify his signature on the *md-file*. The *FEK* and *FSK*

are used to distinguish between read and write access. *FEK* provides read only access to the file while having both the *FEK* and *FSK* allows read and write access.

SiRiUS implements in-band key distribution depending largely secure public-key servers and public-key cryptography in their design. Only file or block encryption is done with symmetric-key algorithm. SiRiUS uses RSA for asymmetric encryption, AES for symmetric encryption, SHA-1 for hashing and DSA for signing. It provides end-to-end encryption of data thus preventing adversaries from accessing files even if they have access to the physical storage device.

2.3.3.2 CRUST: Cryptographic Remote Untrusted Storage without Public Keys [8]

The aim of this paper was to achieve a secure stackable file system layer design over insecure remote untrusted storage systems without making any change to the original system. A system without having a file sharing offer or even a system that does not offer file sharing at all, could also layer CRUST as an extension. Data at rest is kept encrypted. Symmetric key is used to encrypt data along with other cryptographic methods to achieve data integrity, access control and achieve the ability to differentiate between readers and writers to provide flexible control on file access privileges. It included its own in-band key distribution mechanism. A long term key is shared by each user with every other user. The key distribution is carried out using a mechanism that trusts an agent to set up the system and does not involve the users to communicate directly or through any on-line message exchange. Encrypted information, using symmetric key, is shared with file owners and other users. Small number of keys is used to allow file system access securely.

Access privilege for each file is granted based on per–user having file ownership, read only or read-write privileges. Each file has two parts: *data file* and *meta-data file*. The *data file* stores the encrypted form of the data and the *meta-data file* keeps the other information required to achieve successful key managing and authentication. User and their IDs are stored in a user table. The same ID is never reused to enable efficient revocation. The access right of each user to a file is allocated in a meta-data space, called *lockbox*. The distinction between readers and writers is achieved using a MAC-based symmetric-key signature scheme.

It is broadly known that public-key cryptography algorithms are orders of magnitude slower than their symmetric-key counterparts. This fact encouraged the design of this file system that avoids using public-key cryptography and uses symmetric-key replacements instead. The basic design follows the direction of SiRiUS and is especially useful in situations where the users have no control over the underlying file system. It is also useful for sharing files between users that are rarely online, because direct communication between the users is not necessary. An additional insight to this approach is that the servers are not required to carry out cryptographic operation.

2.3.4 Encryption Mechanisms Used in EHR

The symmetric key encryption [20][36][37][38][39] and asymmetric key encryption [40][41][42][43] procedures have been included further in various EHR protocol or system strengthening. Patient Controlled Encryption: Ensuring Privacy of Electronic Medical Records [20], a hybrid public key infrastructure solution (HPKI) for HIPAA privacy/security regulations [36], Strategies for health data exchange for secondary, cross-institutional clinical research [38] are some of the examples.

2.3.5 Attribute Based Encryption Used in EHR

The concept of ABE was first introduced in [10][11] to store and share encrypted data without using symmetric key. Since attribute means property or features of an object, element, or data in Attribute Based Encryption (ABE) the private key and ciphertext is associated with attributes (e.g., the country he lives, gender etc.) of the user. Access structures are specified based on the attributes so decryption of the ciphertext is possible only if the set of attributes matches. In ABE, if A encrypts data using K_A, B can decrypt this data using K_B, as long as the identities of A and B are close to each other. Here, identities are considered as a set of descriptive attributes, and thus it was termed as Attribute-Based Encryption (ABE). An important feature of ABE is collusion-resistance i.e.an attacker in possession of multiple keys should only be able to access data if at least one individual key grants access. Further variations to ABE encryption are found to enhance the basic structure [1][2]. Recently, different efforts have been carried out to use ABE in securing EHR management system [1][2]. Some of the various researches done using Attribute-Based Encryption are accounted below as examples.

2.3.5.1 Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data [10]

This paper come up with a cryptosystem in which distribution of private key to another user is not necessary to decrypt encrypted data, thus achieving a fine-grained sharing. This is done by assigning sets of descriptive attributes to cipher texts by the encryptor and access structures are linked with private keys. Thus making a Key-Policy Attribute-Based Encryption (KP-ABE) in which the access structures holds the rights to allow which cipher text a user could decrypt.

Usually in an ABE system, a user's key is able to decrypt a ciphertext if the attributes labeling the key and the ciphertext matches. But this research developed a richer type of cryptosystem by matching the tree-access structure associated with a key and the attribute linked with a ciphertext for a user, to be able to decrypt different parts of an encrypted data stored on the server. A tree-access structure is a construction in which the set of descriptive attributes labeling the ciphertexts are associated with the leaves of a tree-access structure identifying the private keys.

Four algorithms are used in this KP-ABE scheme:

- Setup - A randomized algorithm that takes implicit security parameter as input and provides PK (public parameters) and MK (master key) as outputs.
- Encryption - A randomized algorithm that takes m (a message), γ (a set of attributes), PK (the public parameters) as inputs and provides E (ciphertext) as output.
- Key Generation - A randomized algorithm that takes $\mathbb{A}$ (an access structure), MK (the master key), PK (the public parameters) as inputs and provides D (a decryption key) as output.
- Decryption - An algorithm that takes E (ciphertext), D (the decryption key), PK (the public parameters) as inputs and M (the message), if $\gamma \in \mathbb{A}$, as output.

2.3.5.2 Privacy Preserving EHR System Using Attribute-Based Infrastructure [2]

This paper focuses on allowing a patient to be able to provide access to their health data based on the attribute of the user receiving the access, using type-identifier (e.g., if the user is a doctor or a medical official or a pharmacist etc.), with their characteristics (e.g., name, ID, specialization, location etc.) and what portions of their medical data the patient wants to share with them. Adaptive chosen ciphertext (CCA-2) secure broadcast ciphertext-policy attribute –based encryption, consisting of five algorithms (Setup, Extract, Encrypt, Decrypt and Delegate), is used to allow direct revocation of user access when necessary.

2.4 Motivation and Objective of Our Work

Many existing network-based storage systems depend on the remote file server. The data in these solutions is often stored unencrypted, and the users rely on the server's access control [44][45]. This means that users efficiently trust the file server's administrators and concentrate on defending against malevolent users accessing the network. The data in these cases may be uncovered from backup copies or stolen hard-disks. As the world advances, it is becoming more complicated to secure, yet more vulnerable to attacks. Hence such methods cannot be applied in our model since EHR deals with very sensitive data and requires maximum confidentiality and secrecy. [21]

Advanced file systems are designed for securing remote storage systems and for allowing more flexible file sharing between users applying various data integrity and access control techniques. The strictest trust model present in related work avoids trusting the entire storage infrastructure. In such systems, the data is encrypted by authors before it is sent to the server, and decrypted by readers after it is received from the server. Access control is achieved by cryptographic means, and does not depend on the file server's mechanisms.

Our work follows the model of untrusted server storage as in the system like SiRiUS. We use in-band key distribution and replaced public-key cryptography with symmetric key algorithms to improve the performance. The key server performs checks before issuing a data encryption key as an access control measure.

The main object we addressed in our work is developing a symmetric key-based EHR management protocol that has successfully introduced the attribute-based access control in symmetric-key solution. Although symmetric encryption will be used, an indirect authentication of the doctor will be performed by the key server based on the doctor's attributes. Thus the access control resides with the patient. The patient, being the owner of their file, controls providing the access permission to the EHR. This is similar to distributing the file signature public-key in SiRiUS.

Using symmetric key methods data integrity and cryptographic access control is applied instead of public-key signatures. The design completely avoids the use of public-key cryptography in order to achieve better performance than existing systems because:

- Symmetric key encryption is more secure (since the key or shared secret has to be distributed to the sender and receiver using a secure channel before communications starts) compared to public-key cryptography which is usually used between two entities who do not share a secret or key before the communication session starts)

- Symmetric key encryption is faster than public-key encryption as it does not require as many CPU cycles as public-key encryption. But then again, public key has the advantage of being used in digital signatures to guarantee that a message was created by a particular entity or authenticate remote systems or users. But since an indirect authentication of the doctor will be performed by the key server based on the doctor's attributes and non-repudiation was not our focus, we can avoid aiming for this advantage.

We believe our design will exhibit a lower security overhead than earlier systems. This paper introduces the basic ideas and architecture with recommendations.

Chapter 3

AVISPA

The full form of AVISPA is "Automated Validation of Internet Security Protocols and Applications". AVISPA is the tool that is used frequently nowadays for verification of cryptographic protocols. Figure 6 shows the whole AVISPA system architecture. AVISPA [12] is a push-button tool with industrial-strength technology for the analysis of diverse Internet security protocols [13][14] and applications. AVISPA is being used by the developers of different security protocols and by academics as well due to the level of scope and robustness it permits while allowing good performance and scalability. The ability to test same protocol specification by applying different verification techniques makes this tool highly desirable. The AVISPA community has demonstrated a number of Internet Engineering Task Force (IETF) security protocols, and numerous protocols have been verified. [46]

AVISPA is a specialized model checker for security protocols to verify that any proposed protocol is free from security threats. AVISPA does not evaluate the performance of the protocol under differing conditions and stresses that might help to conclude the efficiency or analyze performance of the protocol. AVISPA focuses solely on a protocol to deduce if it is totally secure or not. Ensuring the security of cryptographic protocols is the crucial function. It is a security protocol verification tool using fully automatic processes applying various analysis methods, permitting the user to use different tools for a single protocol modeling. Security protocol designs are further aided with the use of a graphical user interface SPAN [47][48] (Security Protocol Animator).

AVISPA uses modular and expressive formal language to specify a protocol and its security properties [35][12][42][43]. AVISPA depends on its back-ends, which use an automatic analytic technique to discover any flaws if they exist. The architecture of AVISPA is shown in Figure 6. In AVISPA, protocol roles are modeled in the High-Level Protocol Specification Language (HLPSL for short [49]) as state transition systems. AVISPA uses this common input language HLPSL for its four back-end tools. Designing

such a common language sometimes raises some problems e.g., making the protocol description language inapt for unbounded verification etc.

The HLPSL2IF tool transforms a HLPSL specification into an Intermediate Format (IF), a transition system of infinite state. This IF specification is being scrutinized by any of the four back-end tools (i.e. the various verification tools embedded in AVISPA that uses an automatic analytic technique to detect any flaws if they exist):

- On-the-Fly Model-Checker (OFMC - uses different symbolic methods to delve into the state space in a demand-driven way) [50]
- Constraint-Logic-based Attack Searcher [51] (CL-AtSe - offers the conversion from any security scheme specification inscribed as transition relation in IF into a set of restraints, which are used to find if there are attacks present
- SAT-based Model Checker [52] (SATMC - generates a propositional formulae which is inserted into a state-of-the-art SAT solver and any model found is converted back into an attack)
- Tree Automata-based Protocol Analyzer [53] (TA4SP - in charge for estimating the intruder knowledge by using regular tree languages).

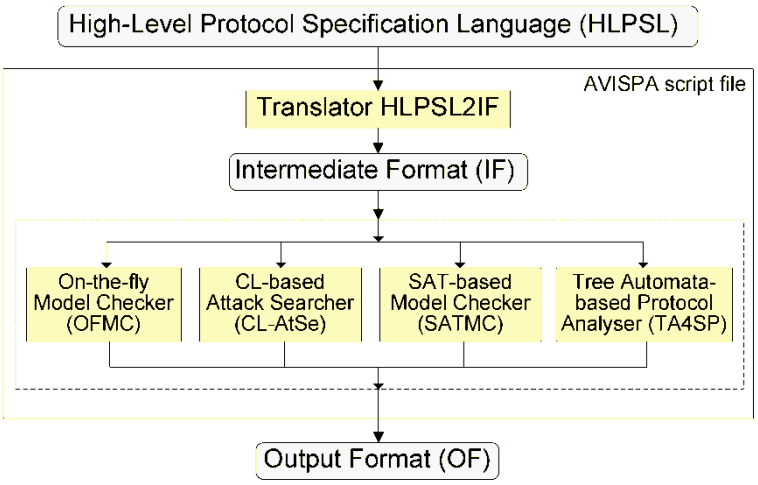

Figure 6: Architecture of the AVISPA tool [14]

HLPSL specification contains the following parts:

- Roles - When modeling a protocol, it is easier to begin with an sketch of the flow of messages in `A-B` notation, and then progress with the specification of the basic roles. HLPSL is a role-based language which means that the actions of each kind of participant are detailed. There are two kind of roles:
  - Basic role – For every type of participant in a protocol, there is basic role stating his sequence of actions.
  - Composed role – Multiple basic roles are joined together into a `composed role` that instantiates the basic roles and specifies about how the resulting participants relate with one another by executing them together, usually in parallel (with interleaving semantics). Composed roles describe sessions of the protocol. There is no transition section in composed roles but rather a composition section in which the basic roles are instantiated.

- Transitions – A set of transitions are present in a HLPSL specification. Each one denotes the receipt of a message (`RCV`) and the sending of a reply message (`SND`). A transition comprises of a trigger, or precondition, and an action to be executed when the trigger event occurs.

- Session - After a basic role is defined, composed roles are needed to be defined which describe sessions of the protocol. A `composed role` instantiates one instance of each basic role thus describing one whole protocol session. By convention, such a `composed role` is termed `session`. In the `session` role, all the channels used by the basic roles are stated.

- Environment – This is a top-level role containing global constants, a formation of one or more sessions (where the intruder may play some roles as a genuine user) and also a statement describing the knowledge an intruder initially might have. E.g., typically, the names of all agents, all symmetric keys, all public keys, his or her own private key, any keys he or she shares with others, and all publicly known functions are included. The final declaration in a specification is always an

instantiation of the top level role i.e. `environment()`.In the environment role, a number of sessions are instantiated corresponding to the composed `role` session.

Figure 7 [49] is shown below for understanding a valid representation of role instantiation. There are three agents (or principals) taking part in this scenario, namely, `a`, `b` and `i`. In both the sessions, `a` plays role `alice` and these two instances are called `alice 1` and `alice 2`. In the first session, the role of `bob` is played by `b` (instance termed `bob 1`), while in the second session, it is played by the `intruder` (instance termed `bob 2`).

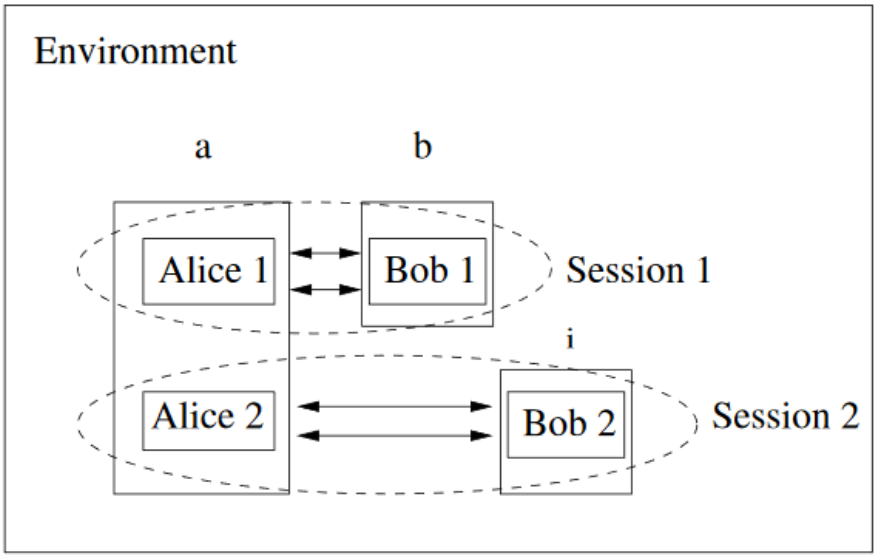

Figure 7: A valid representation of role instantiation [49]

3.1 Security Goal Specification

By means of the back-end tools of AVISPA, secrecy and various forms of authentication goals could be validated. AVISPA supports four types of goal predicates: `witness`, `wrequest` (for weak authentication), `request` (for strong authentication) and `secret`. Below we have explained how the predicates are used to lay down security goals, namely, `secrecy` and `authentication`.

Secrecy is tested with the help of goal predicate `secret (E, id, S`, which confirms that the secret information `E` should be known only by the agents of set `S`. The label `id` (of type `protocol_id`) is used to identify the `goal`. In the HLPSL goal section, the

statement `secrecy_of id` should be given to refer to it. An intruder making an effort to break a secret launches different attacks. The intruder learns a value, which he is not allowed to know and is considered as secret, if an attack is successful. Thus, the secrecy property is violated.

Authentication is confirmed using diverse goal predicates: `witness(A, B, id, E)`, `request(B, A, id, E)` (for strong authentication) and `wrequest(B, A, id, E)` (for weak authentication). The `witness` predicate is used for (both strong and weak) authentication of `A` by `B` on `E`. This is verification that `A` is a witness for the information `E`. The `request` predicate is used for the strong authentication property (and `wrequest` is used for weak authentication property) of `A` by `B` on `E`, which states that `B` requests an inspection of the value `E`. The authentication property is conveyed in the HLPSL goal section, which is written as `authentication_on id` (likewise, `weak_authentication_on id` for weak authentication). In this expression, `id` is a label (of type `protocol_id`) to mark the goal. If any of the back-end tools discovers a trace in which the request event is preceded by a witness event initiated by an agent other than A, an attack will be informed. Furthermore, an attack trace will also be reported if no valid witness is found for a request.

Some features of protocols sometimes might not be possible to be modeled effectively e.g., probability, timestamps etc. Also subtle syntax errors might be present, resulting in models that produce incorrect results in AVISPA. Since AVISPA uses a common language for its input, this also contributes in creating some problems. For example, the link between agents and their keys is often hard-coded, resulting in making the protocol description language incompatible for unbounded verification. Tests might not also detail the interactive restrictions that were used.

3.2 Security Protocol Animator (SPAN)

HLPSL is an expressive language compared to the basic `Alice & Bob` notation. Protocol specifications in HLPSL are separated into roles. The basic roles, define the actions of principals in an execution of the protocol. Other roles e.g., `composed role`, instantiate several of these basic roles to develop sessions of the protocol. Lastly, the `environment role` states the operational principals and sessions whose execution is to

be considered. Thus writing a HLPSL specification requires protocols to be defined as `role` by `role` rather than as message by message (e.g., `Alice & Bob` notation specification messages). Thus an HLPSL specification becomes far less vague but more challenging to read. Therefore, to avoid the difficulty for the protocol designers to understand if the HLPSL specification matches to the `Alice & Bob` protocol originally thought of, SPAN [47][48] (Security Protocol Animator), Figure 8, a protocol animator aimed at helping protocol developers in writing AVISPA specifications [49], is used which symbolically executes an HLPSL protocol specification.

SPAN Figure 8 is a security protocol animator for HLPSL and CAS+ specifications. HLPSL is the language used for stating cryptographic protocols for the AVISPA toolset and CAS+ is a light development of the CASRUL language [54]. New Version now permits to translate a CAS+ specification into an HLPSL specification. SPAN is a tool that animates the HLPSL specifications and helps in producing `Message Sequence Charts (MSC)` which can be viewed as an `Alice & Bob` trace from an HLPSL specification [49].

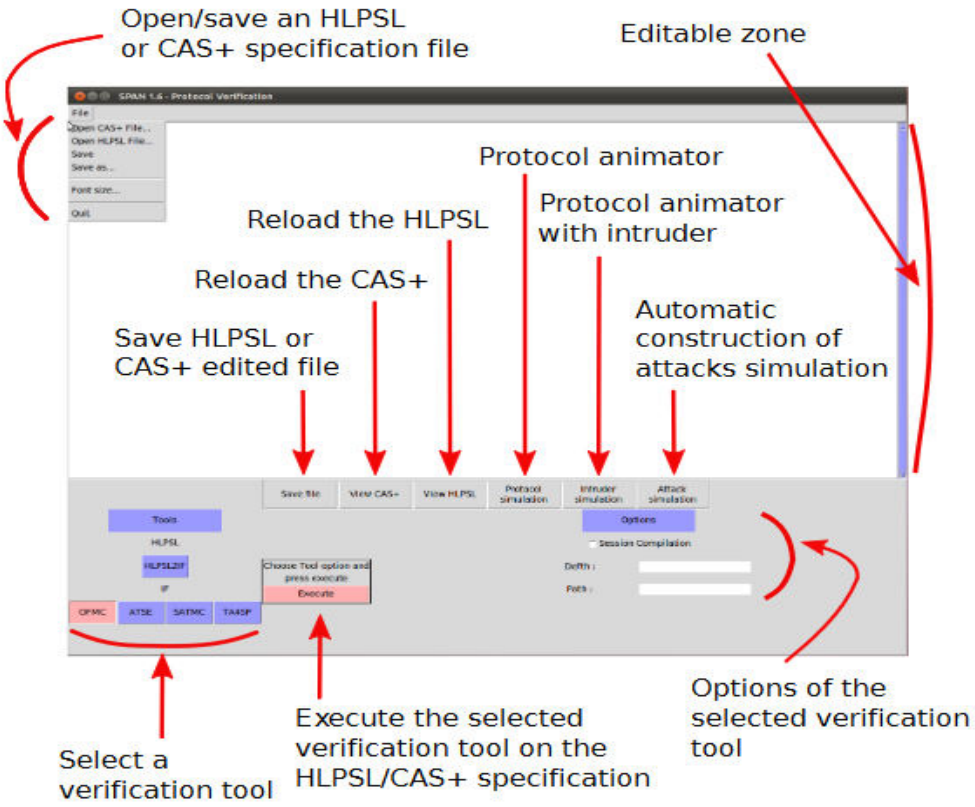

Figure 8: The full SPAN main graphical interface. [55]

3.3 Detection of Attacks in AVISPA

Security goals are stated in HLPSL by supplementing the transitions of basic roles with goal facts and then allocating them a meaning by describing, in the HLPSL goal section, what conditions (i.e. what combination of such facts) designate an attack. For example the goal facts state which values should be secret between which entities (i.e. which agents are permitted to know such secrets). The goal statement in the goal section defines that anytime the intruder acquires a secret value, which he or she was not given permission to know and was intended to be kept a secret, it should be considered an attack. Internally, the attack conditions are detailed in terms of temporal logic, but suitable and brief macros are provided for the two most commonly used security goals, authentication and secrecy.

If the secret is a mixture of constituents from several roles, and if the intruder plays a role in one session, he or she can validly learn the secret and the attack that follows cannot be detected. In other words this means that if in a certain session the intruder plays the role of an honest agent who is permitted to know the secret value, then the intruder is accepted to know it and no attack is reported for this violation. In this attack after learning the secret, the intruder can re-use this value later in some new session (where he does not play the role of an honest agent) masquerading as one of the honest agents, while the other agents believe that the value is a shared secret between honest agents only. However, since it is suggestive of an authentication attack, it could be detected all the same.

The initial states and a transition relation in AVISPA together describe an infinite-state transition system. The goals of the protocol in Intermediate Format (IF) can be created by specifying attack states. A protocol is considered secure for an attack state g if there is no reachable state s such that matches g [56]. All back-ends of the AVISPA tool have the same output format which can be based to graphically represent an attack as a sequence of message exchanges.

AVISPA library [46] contains a collection of specifications of security protocols and problems written in the HLPSL which have been analyzed by the AVISPA Tool. They are taken from a set of protocols described in a deliverable [57]. The deliverable provided selected protocols, presented their security properties (goals) and verified if the protocol satisfied the desired security property (at least in a certain configuration setting). Otherwise

the deliverable found a counter example to prove that the property is breached. Known common attacks, e.g., Man-in-the-Middle attack (MitM), replay attack, breach of secrecy or authentication, etc. along with examples of various security goals a protocol could be designed to achieve which are taken into account in AVISPA are given in Table 1below. Here, Man-in-the-Middle Attack relates directly to the Authentication properties hence it is not mentioned separately in the table.

Table 1: List of Security Properties (or Security Goals) in AVISPA used to detect an Attack

Security Properties (or Security Goals)	
Confidentiality (Secrecy)	Secrecy of session key `k`
Authentication	Authentication of a peer in role `A` with strong agreement on nonce `n`
	Authentication (unicast)
	Entity authentication (Peer Entity Authentication)
	Message authentication (Data Origin Authentication)
	Replay Protection
	Authentication in Multicast or via a Subscribe / Notify Service
	Implicit Destination Authentication
	Source Authentication
Authorization	Authorization (by a Trusted Third Party)
Key Agreement Properties	Key authentication
	Key confirmation (Key Proof of Possession)
	Perfect Forward Secrecy (PFS)
	Fresh Key Derivation
	Secure capabilities negotiation (Resistance against Downgrading and Negotiation Attacks)
Anonymity	Identity Protection against Eavesdroppers
	Identity Protection against Peer

Denial-of-Service (DoS) Resistance	(Limited) Denial-of-Service (DoS) Resistance
Sender Invariance	
Non-repudiation	Accountability
	Proof of Origin
	Proof of Delivery
Safety Temporal Property	

Chapter 4

Proposed Electronic Health Record (EHR) Management Protocol

Many different approaches have already been carried out in EHR management architecture and more are being produced to make the system advanced with more added security measures and enhanced upgrades in par with the always enhancing up-to-date technologies. A general and very basic EHR management architecture is already shown in section 2.1, Figure 1. By extending that EHR management architecture we have structured our proposed EHR architecture.

The structure of our proposed EHR architecture is given in Figure 9 depicts that the EHRs are stored centrally in a Data store (DS) by a doctor after receiving the authorization permission from a patient. The patient can store data himself or herself or give permission to a local health professional (Doctor-1) or a remote health professional (Doctor-2) to store. In our model, we have introduced a key server (KS) that shares or provides the encryption key (K) to the doctor, if the doctor has been approved by the patient (P). This key, (K) is used to encrypt the patient's data before storing in the Data store. We have developed a very simple architecture that only focuses on storing data and retrieving data by the patient authorized doctors, incorporating attribute-based access control in symmetric-key solution. Then we validated our concept using AVISPA. A complicated system, or the system in Figure 1, will take into account a lot of other features e.g., stored data viewing or retrieval by the patient, administrative and billing data viewing or retrieval etc. In our design validation we have solely validated the mechanism of storing and retrieving data by patient authorized doctors rather than the patient himself or herself (though our EHR architecture fully support the process) as the patient is the principal entity that controls authorization to his or her data storage here.

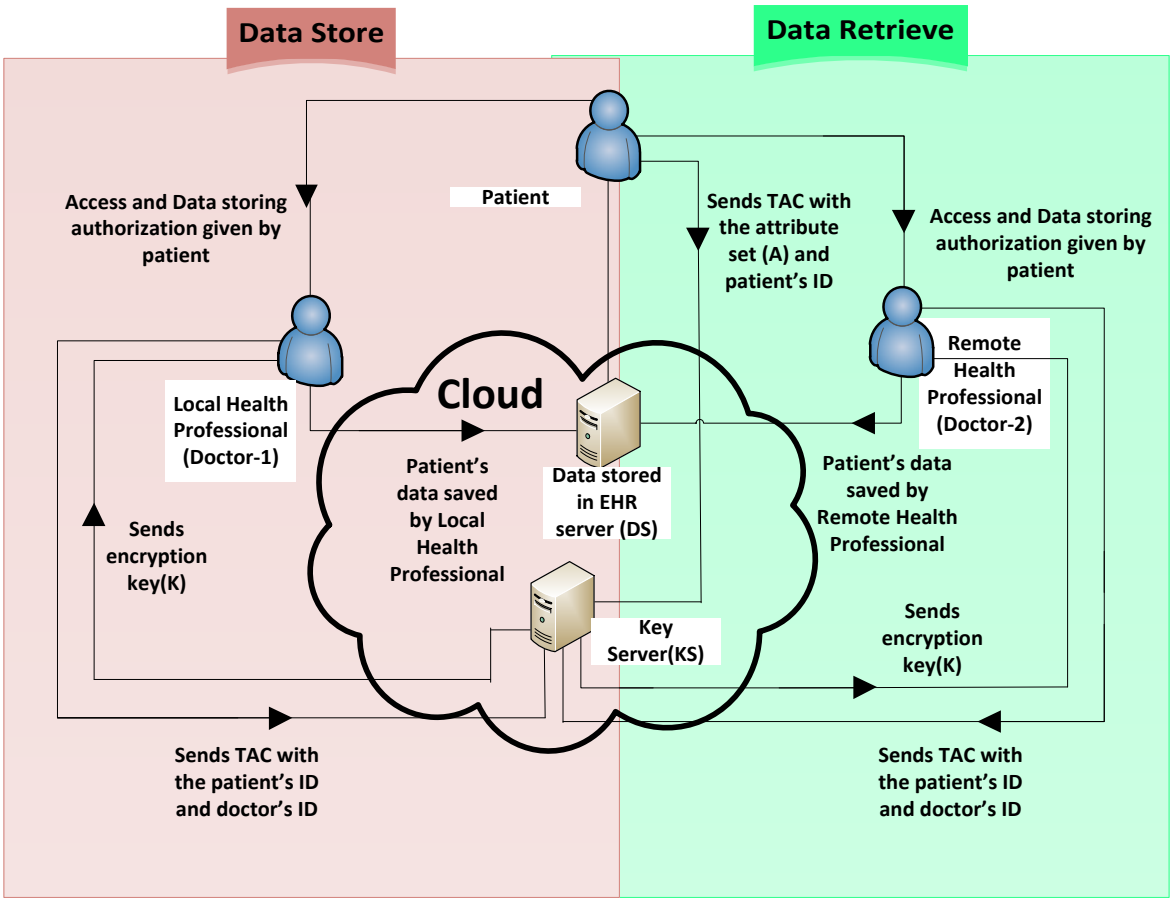

Figure 9: Proposed EHR Management Architecture

4.1 Threat Model

Our goal is to protect the confidentiality of the patient's data. Moreover patient is provided the privilege to be the entity to grant access rights to his or her data to users on the basis of a role or attributes held by the respective user. The key server (KS) and the data store (DS) is considered as trusted entities. Hence the intruder is not granted to play the roles of these two. Attackers or intruders play the roles of the doctor or the patient or those that gain access to the communication channels between patient and the KS or DS. Security should hold for all files. In other words, we seek semantic security, leaking only equality of files to attackers.

4.2 Security Requirements

The proposed protocol has been aimed to maintain the following security requirements which ensures the requirements (**R1**, **R2**, **R3** and **R4** given in chapter 2) directed in [25] to maintain patients' trust:

SR1: Patient will control the access of his or her Electronic Health Record (EHR). Only patient-authorized health professional (e.g., a doctor) will have access to EHR. A patient can define a set of attributes (A). Any doctor with such attribute will have access to EHR of that patient. [**R2**, **R4**]

SR2: Access control will be both doctor and health record specific, i.e., doctor D has granted access to patient's health record with a record-specific transaction code (TAC). [**R1**]

[Note: **R1** is ensured with this security requirement because only an authenticated doctor with granted access to the specific health record is allowed to make changes to the data. Hence integrity is indirectly maintained as the data changes will be valid and accurate.]

SR3: Although symmetric encryption will be used, an indirect authentication of the doctor will be performed by the key server based on the doctor's attributes.

SR4: Data encryption key will be generated by the key server and will be sent to the authenticated doctor only. This key will be known by the key server and the doctor only. [**R3**]

SR5: EHR will be known by the patient and the doctor only. [**R3**]

4.3 Proposed Protocol

The following four entities or principals are involved in this proposed protocol:

- Patient (P): Carries a health card with secret key chip.
- Doctor (D): Generates Electronic Health Record (EHR) and stores encrypted EHR to the Data Store (DS).

- Key Server (KS): Authenticates the doctor (D) based on attributes set (A) provided by the patient. Generates encryption key to encrypt EHR. Forwards the key to the authenticated and authorized doctor (D).

- Data Store (DS): Stores encrypted EHR corresponding to each transaction code (TAC) and patient ID (P), however does not understand the content of the EHR.

The proposed protocol for storing data is shown in Figure 10 and for retrieving data in Figure 11. The specification deals with a list of exchanged messages. They describe the model as an un-attacked run of the protocol. The form `P → D: TAC` means that role `P` sends the message `TAC` to role `D`. Whereas `D → KS: {D.P.TAC}_K`$_{dks}$ means that role `D` sends a message to `KS` after encrypting using the symmetric key K_{dks}. Here, "`{}`" means encryption and "`.`" is used for concatenation. The recipient of a message must be the dispatcher of the next one. Note that `TAC` is not encrypted during the transfer from `P` to `D`. Hence anyone including any intruder can receive or collect the `TAC`. The reason is that merely collecting `TAC` will not be sufficient to request an encryption or decryption key `K` from `KS`. Nor will a `TAC` alone can help in reaching a decrypted form of the stored `EHR` after receiving the encrypted data from `DS` by requesting with a `P.TAC`. The encryption or decryption key is issued by `KS` to only those entities that match or are given access permission by the patient in the attribute set `A` provided by `P` to `KS` in `Message 1`. Moreover the transfer of the key `K`, after its generation, from `KS` to `D` is encrypted using a symmetric key, present between `KS` and `D`, to ensure secrecy is present. An intruder cannot obtain `K` from its encrypted version without the symmetric key used to encrypt it. The communication is made between authenticated entities only. The desired message sequence chart for data storing and data retrieval is given in Figure 12 and Figure 13 respectively. Table 2 shows the notations being used in this proposed protocol.

Message 1: P → KS : {P.A.TAC}_K_{pks}

% % TAC = genTAC(K_{chip}, timestamp), KS stores A (set of attributes)
%% against the identity of the patient (P).

Message 2: P → D : TAC

Message 3: D → KS : {D.P.TAC}_K_{dks}

Message 4: KS → D : {K}_K_{dks} %% K = h(P. TAC. K_{pks})

Message 5: D → DS : {P.TAC.{EHR}_K}_K_{dds}

Figure 10: The proposed protocol for storing data

Message 1: P → D : TAC

Message 2: D → KS : {D.P.TAC}_K_{dks}

Message 3: KS → D : {K}_K_{dks}

Message 4: D → DS : P.TAC

Message 5: DS → D : {P.TAC.{EHR}_K}_K_{dds}

Figure 11: The proposed protocol for data retrieval

Table 2: The notations being used in this proposed protocol

Notations	Meaning
P, D, KS, DS	Principals: Patient, Doctor, Key server and Data store
K_{chip}	Secret key written in the patient's chip card
timestamp	Time of an instance
genTAC(), h()	One way function used to generate TAC and hash values
TAC	Transaction Code
K_{dks}	A symmetric key between Doctor and Key server
K_{pks}	A symmetric key between Patient and Key server
K_{dds}	A symmetric key between Doctor and Data store

K	Data encryption key
A	Attribute set selected and shared by patient
EHR	Electronic Health Record to be stored

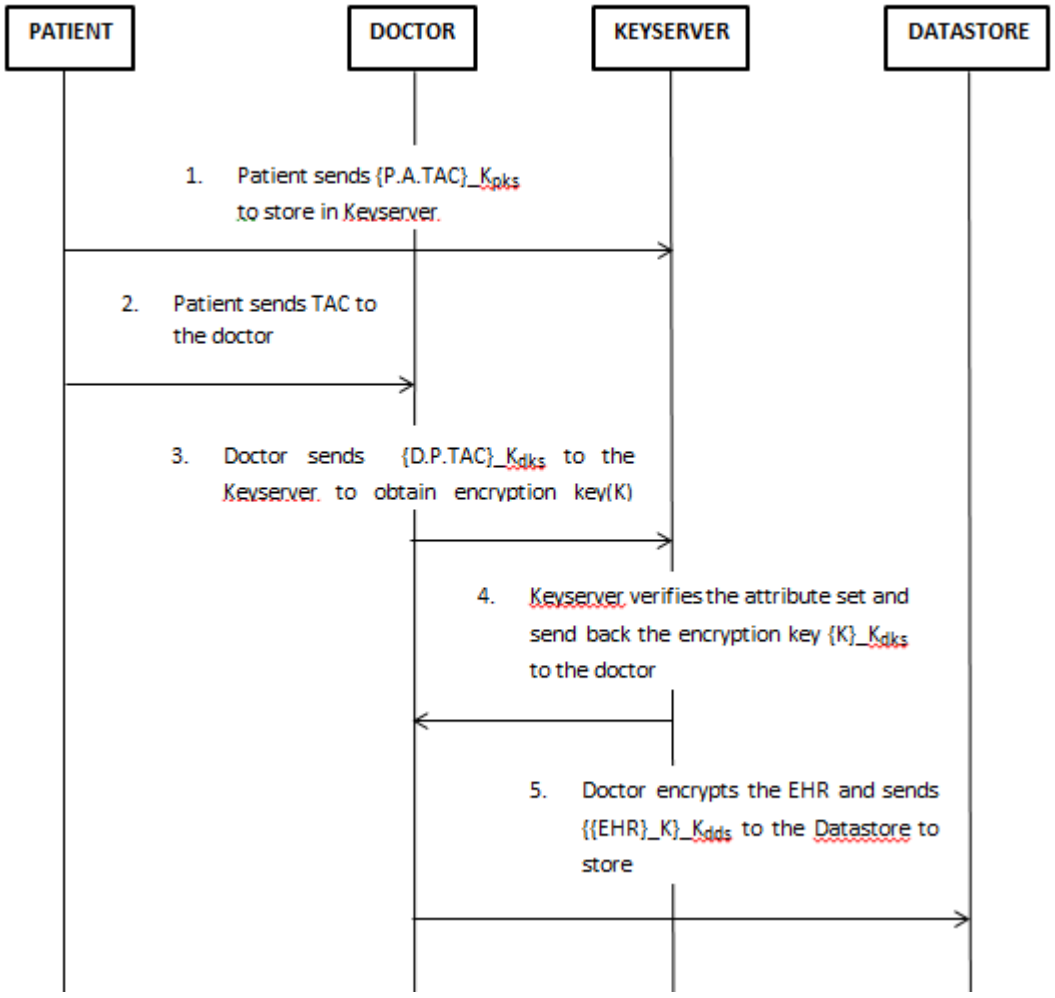

Figure 12: Message sequence chart for data storing

Below we have explained each message and how the message ensures the Security Requirements (SR) presented in section 4.1. For example, the first message warrants SR1, SR2 (i.e., the first and second security requirement).

Message 1: The patient generates the TAC where $TAC=genTAC(K_{chip}, timestamp)$ and then sends the TAC to the key server with the patient's ID, P and attribute set(A) selected by the patient. Before sending the whole message is encrypted with a symmetric key, K_{pks}, existing between the patient and the key server. The attribute set will consist of entities the patient wants to give access permission or authorization access to his or her

data. Here, due to the use of genTAC() function, K_{chip} and timestamp, the TAC value will be unique for every transaction (storing or retrieving data). (SR1, SR2)

Message 2: The patient forwards this TAC to the doctor. We termed him or her as doctor-1 for referencing in our proposed model. (SR1, SR2)

Message 3: The doctor sends the patient's ID, P doctor's ID, D and the TAC to the key server. The whole message is encrypted with the symmetric key K_{dks}, present between the doctor and the key sever, before sending to the key sever. The key server performs the following two tasks:

1) The key server decrypts the received message using K_{dks} and matches the received TAC with the one sent by the patient earlier. (SR2)

2) Checks if the tuple P.D.TAC is a member of its attribute set (as well as if D is a member of attribute set A). It is assumed that this attribute set had been defined by the patient earlier. (SR1, SR3)

Message 4: The key server generates the data encryption symmetric key K where $K = h(P.TAC.K_{pks})$ and then forwards K to the doctor encrypted with K_{dks}. (SR4)

Message 5: Finally, the doctor encrypts EHR with the data encryption key, K and sends the encrypted EHR to the data store to be saved in. The whole message is encrypted by a symmetric key K_{dds} shared between the doctor and the data store before sending to the data store. (SR5)

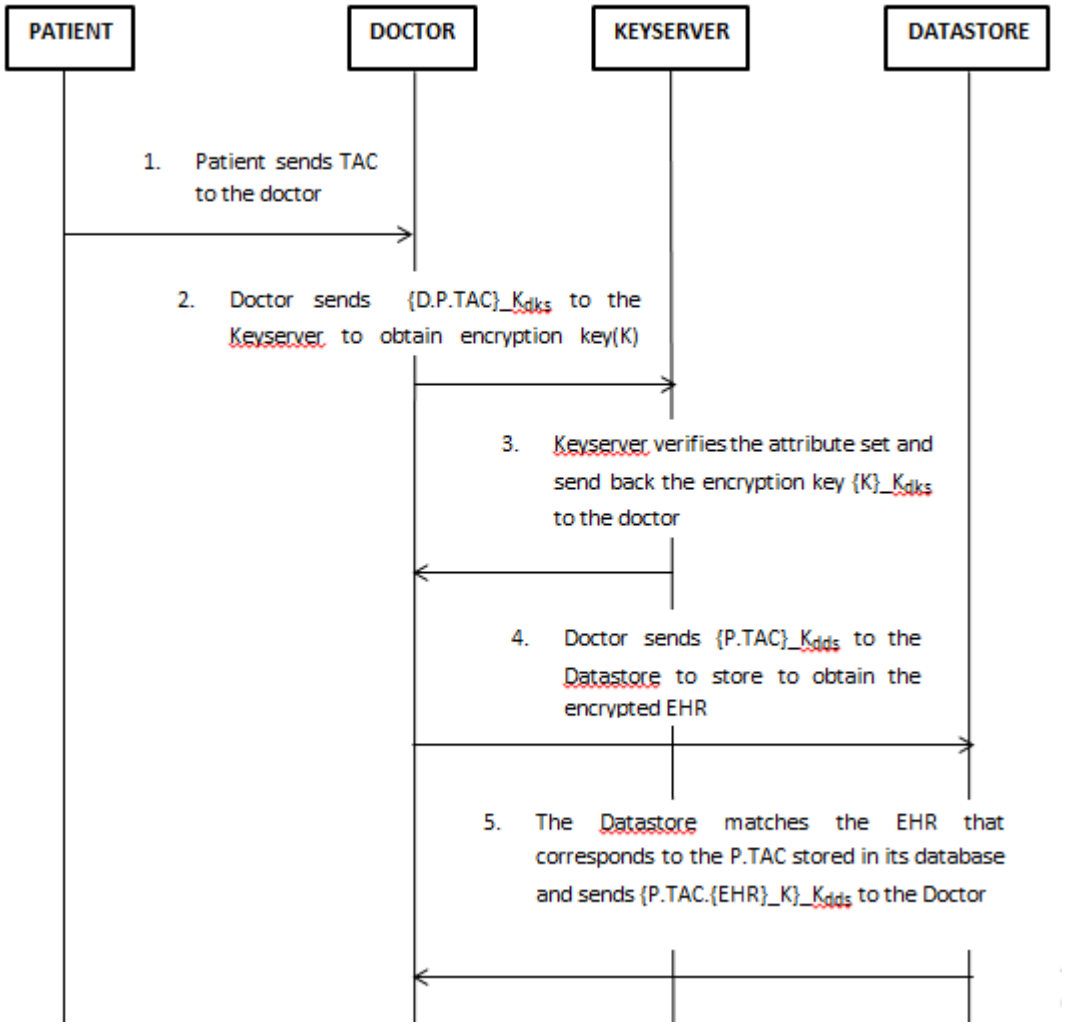

Figure 13: Message sequence chart for data retrieval

Below we have explained each message and how the message ensures the Security Requirements (SR) presented in section 4.1. For example, the first message warrants SR1, SR2 (i.e., the first and second security requirement).

Message 1: The patient sends the `TAC` to the doctor, `D` selected by the patient. The doctor in this step could be same doctor (e.g., doctor-1) from the data storing stage or could be a new doctor selected by the patient (e.g., doctor-2). Here, `TAC = genTAC(`K_{chip}`, timestamp)` is the same `TAC` generated and stored by the patient during data storing by the doctor-1. `TAC` value will be unique for every transaction (storing or retrieving data). (SR1, SR2)

Message 2: The doctor sends the patient's ID, P doctor's ID, D and the TAC to the key server. The whole message is encrypted with the symmetric key K_{dks}, present between the doctor and the key sever, before sending to the key sever. The key server performs the following two tasks:

1) The key server decrypts the received message using K_{dks} and matches the received TAC with the one sent by the patient earlier. (SR2)

2) Checks if the tuple P.D.TAC is member of its attribute set (as well as if D is a member of attribute set A). It is assumed that this attribute set had been defined by the patient earlier. (SR1, SR3)

Message 3: The key server generates the data encryption symmetric key K where $K = h(P.TAC.K_{pks})$ and then forwards K to the doctor encrypted with K_{dks}. (SR4)

Message 4: The doctor sends the patient's ID, P and the TAC to the data store. The whole message is encrypted by a symmetric key K_{dds} shared between the doctor and the data store before sending to the data store. The data store checks if the encrypted data corresponding to the tuple P.TAC is present in its database. (SR5)

Message 5: The data store sends the encrypted EHR to the doctor. The whole message is encrypted by a symmetric key K_{dds} shared between the doctor and the data store before sending. (SR5)

Since any data and message is encrypted using symmetric key (assuming the key is not compromised and kept a secret) in all stages of communication between the participants, an intruder cannot decipher the content of a hijacked message nor can he/she encrypt the message using valid symmetric key before forwarding it to someone faking his/her identity.

Chapter 5

Security Model Development

We have developed and verified our proposed model using AVISPA tool. The procedure used in the development model, steps with which we conducted the verification, the model and its output is described in details in this chapter. We have run two separate AVISPA models for data storing and data retrieval scenario to test the target secrecy property in our proposed EHR management protocol.

5.1 Modeling Attributes in AVISPA

According to ABE-based authentications, the patient should have provision to select the attributes of the doctor or hospitals, upon which the doctor will be allowed to access the patient's EHR. The patient may select a number of attributes including the identity of the doctor, the field of specialization, the location of the doctor, location of the hospital etc.

While modeling matching of the attributes, we use compound data type set, whose elements are of the compound type, e.g., `agent.agent.public_key`. The identities selected by the patients will be values of the compound type. HLPSL provides a set membership function, `in()` to check if a tuple is already member of the set.

5.2 AVISPA Model

In the AVISPA model we have developed, there are four agent roles: patient (`P`), doctor (`D`), keyserver (`KS`) and datastore (`DS`). Figure 14 and Figure 15 demonstrates the HLPSL specification that we have developed to model these four roles for data storing and retrieval. AVISPA is able to validate the secrecy of any message component. As we have mentioned in section 4.1, data encryption key (`K`) and `EHR` should be known by specific agents. To validate the secrecy of these two parameters, the HLPSL goal `secrecy_of` and the following two secret facts have been added.

For data storing:

- `secret(K', sec_k,{KS,D1})`
- `secret(EHR, sec_ehr,{D1})`

Here, the labels `sec_k` and `sec_ehr` identify the goals and `{KS,D1}), {D1}` are the sets of agents that are allowed to learn the value `K` and `EHR` respectively.

For data retrieval:

- `secret(K', sec_k,{KS,D2})`
- `secret(EHR', sec_ehr,{D2})`

Here, the labels `sec_k` and `sec_ehr` identify the goals and `{KS,D2}), {D2}` are the sets of agents that are allowed to learn the value `K` and `Edata` respectively.

In the AVISPA model for data storing, Figure 14, we have assumed the following:

- K_{pks}: Symmetric key between the patient (`P`) and the key server (`KS`) already present or established earlier between them.
- K_{dks}: Symmetric key between the doctor (`D`) and the key server (`KS`) already present or established earlier between them.
- K_{dds}: Symmetric key between the doctor (`D`) and the data store (`DS`) already present or established earlier between them.
- `TAC`: Generated by the patient (`P`) using the secret key chip written in the patient's chip card and a timestamp.
- `D1`: Local doctor authorized by the patient (`P`) and thus present in the attribute set (`A`) provided by `P`. This doctor is storing the patient's data to the data store (`DS`).
- `K`: Symmetric key generated by the key server (`KS`) and send to the doctor (`D`) to encrypt `EHR`.

- `D2`: Remote doctor authorized by the patient (`P`) and thus present in the attribute set (`A`) provided by `P`. This entity is not taking part in data storing of our simulation and is merely present as a member of the set `A`.

- `C1`: Doctors of a clinic1 authorized by the patient (`P`) and thus present in the attribute set (`A`) provided by `P`. This entity is not taking part in data storing of our simulation and is merely present as a member of the set `A`.

- `C2`: Doctors of a clinic2 authorized by the patient (`P`) and thus present in the attribute set (`A`) provided by `P`. This entity is not taking part in data storing of our simulation and is merely present as a member of the set `A`.

HLPSL supports numerous basic types, some of them are provided below that are used in our developed model:

`agent`: It indicates the principal names. The exclusive identifier i indicates the intruder.

`const`: It denotes the constants.

`symmetric_key`: It means the key for a symmetric-key cryptosystem.

`text`: It is often used for nonces and sometimes for messages.

`nat`: It denotes the natural numbers in non-message contexts.

Moreover,

- We used associative "`.`" operator for concatenation.

- The declarations "`played_by P`" is used to indicate that the agent named `P` plays the role

- `intruder_knowledge` denotes the initial knowledge of intruder.

- Transitions are of the form `X =|> Y`, relates an event `X` and an action `Y`.

- The goal `secrecy_of sec_ehr` means the variable `sec_ehr` is to remain secret forever. Consequently, a security violation will occur if `sec_ehr` is ever derived or found by the intruder.

```
%%% Secured Electronic Health Record Management Protocol showing data  storing of patient(P)by
%%% remote doctor(D1)

%%%--------------------------------PATIENT-----------------------------------
role patient(
            P, D1, KS, DS: agent,
            Kchip: symmetric_key,
            GenTAC: hash_func,
            A: (agent)set,
            Kpks:symmetric_key,
            SND, RCV: channel (dy)
      )
played_by P
def=
      local
            State: nat,
            Tstamp: text,
            TAC1: hash(symmetric_key.text) %%TAC=genTAC(Kchip.tstamp)
      init
            State := 0

     transition
            1. State =0 /\ RCV(start) =|> State':= 1
               /\ Tstamp':= new()
               /\ TAC1':= GenTAC(Kchip.Tstamp')
               /\ SND({P.A.TAC1'}_Kpks)
               /\ SND(TAC1')
end role

%%%-------------------------------DOCTOR-------------------------------------

role doctor(

            P,D1,KS,DS: agent,
            GenTAC : hash_func,
            EHR : text,
            Kdds,Kdks:symmetric_key,
            SND,RCV: channel (dy)
      )
played_by D1
def=
      local
            State:nat,
            TAC1:hash(symmetric_key.text),
            K:message

      init
            State := 0

      transition
            1. State = 0 /\ RCV(TAC1') =|> State':= 1
                        /\   SND({D1.P.TAC1'}_Kdks)
```

```
            2. State = 1 /\ RCV({K'}_Kdks)  =|> State':= 2
                         /\ SND({P.TAC1.{EHR}_K'}_Kdds)
                         /\ secret(EHR, sec_ehr,{D1})
end role

%%%------------------------------KEYSERVER-----------------------------

role keyserver(
            P, D1, KS, DS: agent,
            H : hash_func,
            Kpks, Kdks : symmetric_key,
            Attribute: (agent.agent.message)set,
            SND, RCV: channel (dy)
      )
played_by KS
def=
      local
            State : nat,
            TAC1 : hash(symmetric_key.text),
            K: message,
            A:(agent)set

      init
            State := 0

      transition
            1. State = 0 /\ RCV({P.A'.TAC1'}_Kpks) =|> State':= 1

            2. State = 1 /\ RCV({D1.P.TAC1}_Kdks)
                         /\in(D1,A)
                         /\ in(D1.P.TAC1, Attribute) =|> State':= 2
                         /\ K':=H(P.TAC1.Kpks)
                         /\ SND({K'}_Kdks)
                         /\ secret(K', sec_k,{KS,D1})

end role

%%%---------------------------------DATASTORE------------------------------
role datastore(
            P, D1, KS, DS: agent,
            Kdds : symmetric_key,
            SND, RCV: channel (dy)  )
played_by DS
def=
      local
            State : nat,
            TAC1 : hash(symmetric_key.text),
            Edata : {text}_message
      init
            State := 0

      transition
            1. State = 0 /\ RCV({P.TAC1'.Edata'}_Kdds) =|> State':= 1

end role
```

%% Each SND on one side corresponds to a RCV on the other side. In order to put these building %% blocks together, we first defined the composition in a further role the session

```
%%%-----------------------------SESSION-------------------------------
role session(
            P, D1, KS, DS: agent,
            K_chip, Kpks,Kdds, K_dks: symmetric_key,
            Gen_TAC, H_h : hash_func,
            E_HR : text
      )
def=
      local
            A:(agent)set,
            Attributes: ( agent.agent.message ) set,
            SP, RP, SD1, RD1, SKS, RKS, SDS, RDS : channel (dy)

      init
            A := {d1, d2,c1,c2}
         /\ Attributes := {d1.p.tac1,d2.p.tac1, c1.p.tac1, c2.p.tac1}

      composition
               patient(P,D1,KS,DS,K_chip,Gen_TAC,A,Kpks,SP,RP)
            /\ doctor(P,D1,KS,DS,Gen_TAC,E_HR,Kdds,K_dks,SD1,RD1)
            /\ keyserver(P,D1,KS,DS,H_h,Kpks,K_dks,Attributes,SKS,RKS)
            /\ datastore(P,D1,KS,DS,Kdds,SDS,RDS)

end role
```

%%% The environment consists of the attacker ,composition, and session instances

```
%%%---------------------------ENVIRONMENT--------------------------

role environment()
def=

    const
        p, d1, d2,c1,c2,ks,ds: agent,
        kchip,kichip,kpks,kipks,kdds,kidds,kdks,kidks : symmetric_key,
        gentac, h : hash_func,
        ehr : text,
        tac1: message,     %hash_func(symmetric_key.text),
        sec_ehr, sec_k : protocol_id

  intruder_knowledge = {p,d1,ks,ds,gentac,kichip,kipks,kidds,kidks}

    composition
             session(p,d1,ks,ds,kchip,kpks,kdds,kdks,gentac,h,ehr)
          /\ session(p,d1,ks,ds,kchip,kpks,kdds,kdks,gentac,h,ehr)
          /\ session(p,i,ks,ds,kchip,kpks,kidds,kidks,gentac,h,ehr)
          /\ session(i,d1,ks,ds,kichip,kipks,kdds,kdks,gentac,h,ehr)

end role

%%%-------------------------------GOAL-------------------------------
goal
secrecy_of sec_ehr, sec_k
end goal

%%%---------------------------ENVIRONMENT--------------------------
environment()
```

Figure 14: Roles of the four agents, session, environment and goal roles for data storing by a doctor (D1)

In the AVISPA model for data retrieval, Figure 15, we have assumed the following:

- K_{pks}: Symmetric key between the patient (P) and the key server (KS) already present or established earlier between them.
- K_{dks}: Symmetric key between the doctor (D) and the key server (KS) already present or established earlier between them.
- K_{dds}: Symmetric key between the doctor (D) and the data store (DS) already present or established earlier between them.
- TAC: Generated by the patient (P) using the secret key chip written in the patient's chip card and a timestamp.
- D2: Remote doctor authorized by the patient (P) and thus present in the attribute set (A) provided by P. This doctor is retrieving the patient's data from the data store (DS).
- K: Symmetric key generated by the key server (KS) and send to the doctor (D) to encrypt EHR.
- S: It is a table for database the data store (DS) maintains for encrypted EHR against the corresponding patient (P) and transaction code (TAC) number. This is required for DS to provide the correct encrypted EHR when a doctor requests with a particular P and TAC. The data store searches its table to see if it has the requested data before sending it.
- D1: Local doctor authorized by the patient (P) and thus present in the attribute set (A) provided by P. This entity is not taking part in data retrieval of our simulation and is merely present as a member of the set A.
- C1: Doctors of a clinic1 authorized by the patient (P) and thus present in the attribute set (A) provided by P. This entity is not taking part in data retrieval of our simulation and is merely present as a member of the set A.

- C2: Doctors of a clinic2 authorized by the patient (P) and thus present in the attribute set (A) provided by P. This entity is not taking part in data retrieval of our simulation and is merely present as a member of the set A.

```
%%% Secured Electronic Health Record Management Protocol showing data  retrieval of patient(P)by
%%% remote doctor(D2)

%%%--------------------------------PATIENT--------------------------

role patient(
            P, D2, KS, DS: agent,
            Kchip,Kpks: symmetric_key,
            GenTAC: hash_func,
            A: (agent)set,
            Tstamp:text,
            TAC1:text,        %%TAC1=genTAC(Kchip.tstamp)
            SND, RCV: channel (dy)
      )
played_by P
def=
      local
            State: nat

      init
            State := 0

      transition
               1. State =0 /\ RCV(start) =|> State':= 1 /\ SND(TAC1)

end role

%%%--------------------------------DOCTOR---------------------------

role doctor(

            P,D2,KS,DS: agent,
            GenTAC : hash_func,
            Kdds,Kdks:symmetric_key,
            SND,RCV: channel (dy)
      )
played_by D2

def=
      local
            State:nat,
            TAC1:text,
            K:message,
            EHR:text


      init
            State := 0


      transition
```

```
            1. State = 0 /\ RCV(TAC1') =|> State':= 1
                         /\  SND({D2.P.TAC1'}_Kdks)

            2. State = 1 /\ RCV({K'}_Kdks)  =|> State':= 2
                         /\ SND({P.TAC1}_Kdds)

            3. State= 2 /\ RCV ({P.TAC1.{EHR'}_K}_Kdds) =|> State':=3
                        /\ secret(EHR',sec_ehr,{D2})

end role

%%%------------------------------KEYSERVER----------------------------

role keyserver(

            P, D2, KS, DS: agent,
            H : hash_func,
            Kpks, Kdks : symmetric_key,
            Attribute: (agent.agent.text)set,
            A:(agent)set,
            SND, RCV: channel (dy)
      )
played_by KS
def=
      local
            State : nat,
            TAC1 : text,
            K: message

      init
            State := 0

   transition
        1. State = 0 /\ RCV({D2.P.TAC1'}_Kdks)
                     /\in(D2,A)
                     /\ in(D2.P.TAC1', Attribute) =|> State':= 1
                     /\ K':=H(P.TAC1.Kpks)
                     /\ SND({K'}_Kdks)
                     /\ secret(K', sec_k,{KS,D2})

end role

%%%------------------------------DATASTORE----------------------------

role datastore(

              P, D2, KS, DS: agent,
              Kdds : symmetric_key,
              Edata:{text}_message,
              S:(agent.text)set,
              SND, RCV: channel (dy)
      )
played_by DS
def=
      local
            State : nat,
            TAC1 : text

      init
            State := 0
```

```
      transition
            1. State = 0 /\ RCV({P.TAC1'}_Kdds)
                         /\ in(P.TAC1',S)  =|> State':= 1
                         /\ SND ({P.TAC1'.Edata}_Kdds)
                         /\ secret(ehr,sec_ehr,{D2})

end role

%%%------------------------------SESSION------------------------------

role session(
            P, D2, KS, DS: agent,
            K_chip, Kpks, Kdds, K_dks: symmetric_key,
            Gen_TAC, H_h : hash_func,
            Tstamp:text,
            TAC1: text,
            EHR:text
      )
def=
   local
            Edata : {text}_message,
            A:(agent)set,
            S:(agent.text)set,
            Attributes: (agent.agent.text)set,
            SP, RP, SD2, RD2, SKS, RKS, SDS, RDS : channel (dy)

   init
          Edata := {EHR}_H_h(P.TAC1.Kpks)
       /\ A := {d1, d2, c1, c2}
       /\ S:= {p.tac1,  p.tac2}
       /\ Attributes := {d1.p.tac1, d2.p.tac1, c1.p.tac1, c2.p.tac1}

  composition
       patient(P,D2,KS,DS,K_chip,Kpks,Gen_TAC,A,Tstamp,TAC1,SP,RP)
    /\ doctor(P,D2,KS,DS,Gen_TAC,Kdds,K_dks,SD2,RD2)
    /\ keyserver(P,D2,KS,DS,H_h,Kpks,K_dks,Attributes,A,SKS,RKS)
    /\ datastore(P,D2,KS,DS,Kdds,Edata,S,SDS,RDS)

end role

%%%---------------------------ENVIRONMENT---------------------------

role environment()
def=

   const
        p, d1, d2,c1,c2,ks,ds: agent,
        kchip,kichip,kpks,kipks,kdds,kidds,kdks,kidks : symmetric_key,
        gentac, h : hash_func,
        tstamp :text,
        tac1,tac2:text,
        ehr: text,
        sec_ehr, sec_k: protocol_id


  intruder_knowledge = {p,d2,ks,ds,gentac,kichip,kipks,kidds,kidks}
```

```
  composition
    session(p,d2,ks,ds,kchip,kpks,kdds,kdks,gentac,h,tstamp,tac1,ehr)
  /\session(p,d2,ks,ds,kchip,kpks,kdds,kdks,gentac,h,tstamp,tac1,ehr)
  /\session(p,i,ks,ds,kchip,kpks,kidds,kidks,gentac,h,tstamp,tac1,ehr)
  /\session(i,d2,ks,ds,kichip,kipks,kdds,kdks,gentac,h,tstamp,tac1,ehr)

end role

%%%------------------------------GOAL-------------------------------

goal
secrecy_of sec_k, sec_ehr
end goal

%%%---------------------------ENVIRONMENT-------------------------
environment()
```

Figure 15: Roles of the four agents, session, environment and goal roles for data retrieval by another doctor (D2)

5.3 Freshness, Replay and MitM Attacks

AVISPA supports a new() function that produces a fresh value at runtime. To ensure the freshness of time, we use the new() function inside the patient role that generates the timestamp (Tstamp), as shown in Figure 14 since AVISPA does not provide any notion of time. Note that in a transition a primed variable (e.g., Tstamp') denotes the new value of the variable (Tstamp). This new value has been either acquired in the left-hand side or assigned in the right-hand side of the transition. An active intruder, playing concurrently the role of any communicating node, has been introduced. First the intruder_knowledge has been added in the environment() role.

- To identify the existence of any replay attack, we instantiate two identical session()s of the model in parallel.

- Since the key server and the data store have been considered as trusted entities, the intruder is not allowed to play the roles of these two. Thus, the intruder is playing the roles of the doctor and the patient in the last two session()s, as shown in Figure 14 and Figure 15. Hence, the agent identities p and d have been replaced by the intruder identity, i. This will also detect Man-in-the-Middle (MitM) or connection hijack attacks if any such attack exists.

5.4 Model Validation and Analysis

We use SPAN (Security Protocol Animator) [48], a graphical tool that performs an HLPSL protocol specification. SPAN provides the message sequence charts (MSC) that shows the simulations steps of all messages. Figure 16 shows the MSC of the model we have developed for storing data in Data Store and Figure 21 shows the MSC of the model we have developed for retrieving data from the Data Store. During protocol simulation no intruder's role has been added. The sequences of the MSC confirm that the model we have developed is successfully exchanging all messages. AVISPA supports Dolev–Yao channels [58] (denoted by `channel(dy)` in the roles) for message transmission, where the intruder has complete control over message transmission. The intruder can overhear, intercept and synthesize any message, and his actions are only restricted by the restraints of the cryptographic methods used. As we have already mentioned, we have added an active intruder who can perform the roles of the patient or the doctor simultaneously. We have verified our model using all four back-ends of AVISPA to discover an attack if any exists. AVISPA cannot concurrently identify several attacks during a single run. Therefore, we confirm the security goals one-by-one.

The HLPSL specification of our protocol is first converted into a lower level specification by a translator, called the hlpsl2if. After that a specification in an intermediate format is generated, called the Intermediate Format (IF). The output format (OF) of AVISPA is created using one of the back-ends. The analysis of the OF is as follows:

- `SUMMARY`: Informs whether the tested protocol is secure, insecure, or the test is indecisive.
- `DETAILS`: It states what conditions of the test were taken into consideration to declare the protocol is safe or liable to get attacks or the test to be indecisive.
- `PROTOCOL`: It is the name of the protocol.
- `GOAL`: It indicates the goal of the test.
- `BACKEND`: It stands for the back-end name used in the test.

- COMMENTS & STATISTICS: It displays the trace of an attack (if any present) printed in a standard Alice-Bob format.

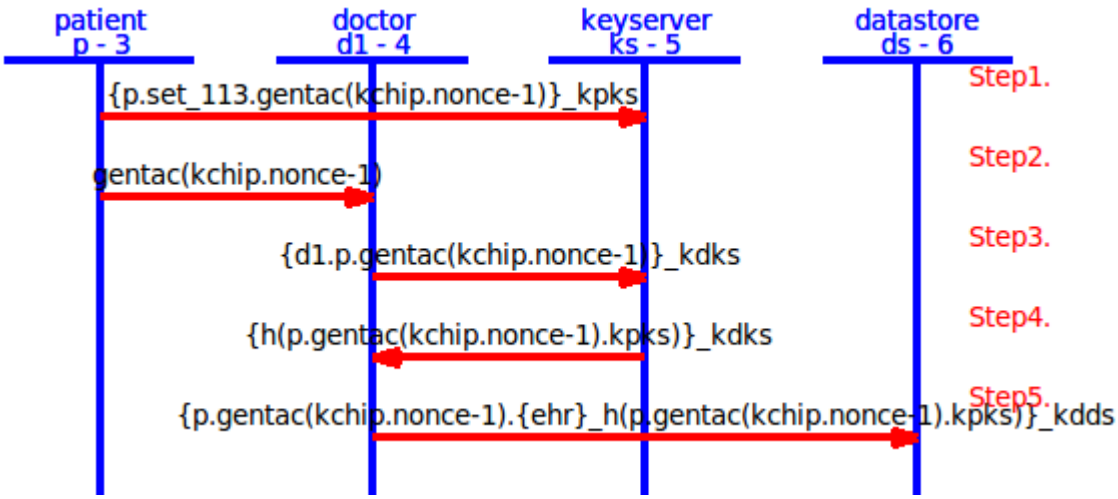

Figure 16: Protocol simulation of the model for data storing

Output Results for Storing Data:

The first two back-ends, OFMC, Figure 17, and CL-AtSe, Figure 18, for BOUNDED_NUMBER_OF_SESSIONS have reported SAFE. The other two, SATMC, Figure 19, and TA4SP, Figure 20, have reported NOT_SUPPORTED and gave INCONCLUSIVE results. The validation output of OFMC, CL-AtSe, SATMC, TA4SP are given in Figure 17, Figure 18, Figure 19, and Figure 20 respectively. Due to the intricacy of the model, we have to run OFMC with a bounded depth. The STATISTICS section of the OFMC output gave us the time needed to execute our protocol specification by the tool and the number of the visited nodes or states during the execution. The terms and their meaning present in the STATISTICS section is given below:

- parseTime: time to parse the input file
- searchTime: time for the analysis of the space of symbolic states (SSS)

- visitedNodes: node visited in SSS
- depth: number of layers visited the SSS (tree)

In `OFMC` backend output, the depth for the search provided is 10, the total number of nodes searched in this case is 9318, which took 21.00s. In `CL-AtSe` backend output, 28 states were analyzed and 22 states were reachable. Moreover, `CL-AtSe` backend took 0.03s for translation and 0.01s for computation.

We can conclude from the outputs that the AVISPA model which we have established is free from the attacks listed in in Table 1 that AVISPA is able to find so the `secrecy` security goal of the model that we have aimed to achieve in our protocol have been validated.

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/Final_Store.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 21.00s
  visitedNodes: 9318 nodes
  depth: 10 plies
```

Figure 17: Validation output of OFMC

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

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

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 28 states
  Reachable  : 22 states
  Translation: 0.03 seconds
  Computation: 0.01 seconds
```

Figure 18: Validation output of CL-AtSe

```
SUMMARY
  INCONCLUSIVE

DETAILS
  ERROR

PROTOCOL
  Final_Store.if

BACKEND
  SATMC
```

Figure 19: Validation output of SATMC

```
SUMMARY
  INCONCLUSIVE

DETAILS:
  NOT_SUPPORTED

PROTOCOL:
  /home/span/span/testsuite/results/Final_Store.if

GOAL:
  SECRECY

BACKEND:
  TA4SP

COMMENTS:
  Sorry, TA4SP does not support set up to now

STATISTICS:
  Translation: 0.00 seconds
```

Figure 20: Validation output of TA4SP

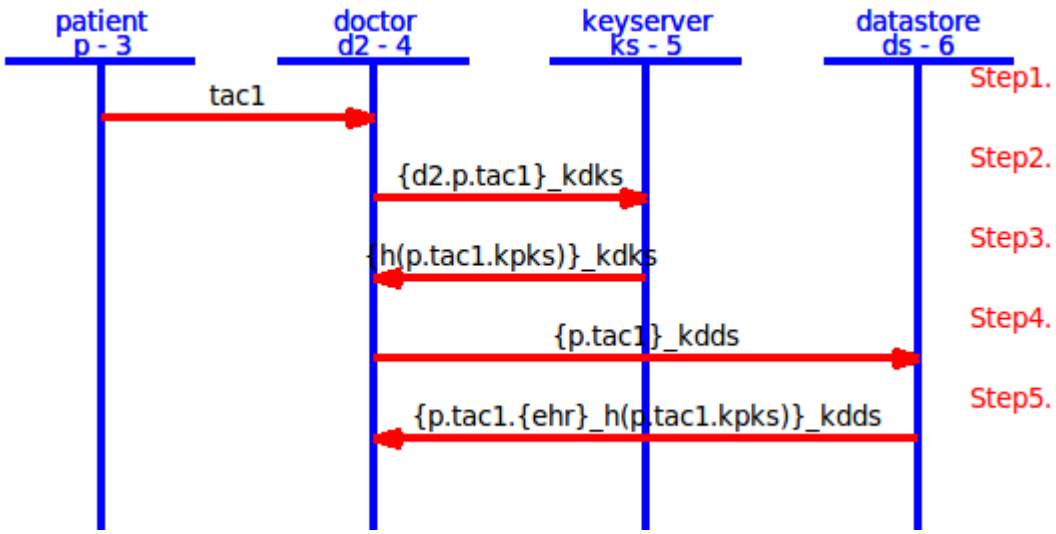

Figure 21: Protocol simulation of the model for data retrieval

Output Results for Data Retrieval:

The output results for the data retrieval are analyzed in the same way as storing data is done. The first two back-ends, `OFMC`, Figure 22, and `CL-AtSe`, Figure 23, for `BOUNDED_NUMBER_OF_SESSIONS` have reported `SAFE`. The other two, `SATMC`, Figure 24, and `TA4SP`, Figure 25, have reported `NOT_SUPPORTED` and gave `INCONCLUSIVE` results. The validation output of `OFMC`, `CL-AtSe`, `SATMC`, `TA4SP` are given in Figure 22, Figure 23, Figure 24 and Figure 25 respectively. Due to the difficulty of the model, we have to run `OFMC` with a `bounded depth`. The STATISTICS section of the OFMC output gave us the time essential to perform our protocol specification by the tool and the number of the visited nodes or states during the execution. We can conclude from the outputs that the AVISPA model which we have developed is free from the attacks listed in Table 1 that AVISPA is able to find so the `secrecy` security goal of the model that we have aimed to achieve in our protocol have been validated.

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/Final_Retrieve.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 24.82s
  visitedNodes: 0 nodes
  depth: 1000000 plies
```

Figure 22: Validation output of OFMC

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

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

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed  : 1302922 states
  Reachable  : 184694 states
  Translation: 0.10 seconds
  Computation: 21.07 seconds
```

Figure 23: Validation output of CL-AtSe

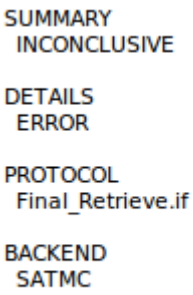

Figure 24: Validation output of SATMC

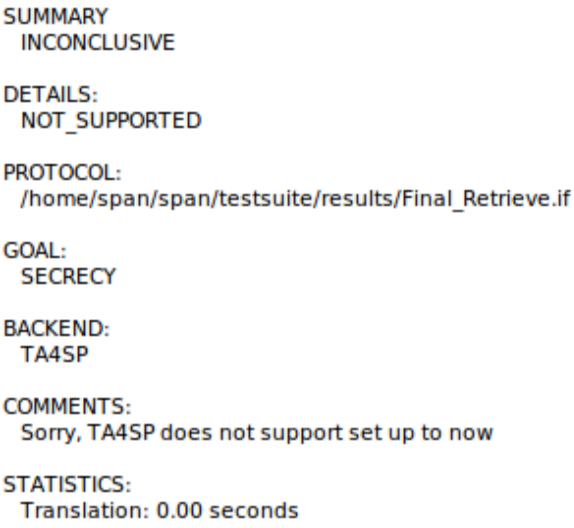

Figure 25: Validation output of TA4SP

5.5 Observations from the Model Validation and Analysis

We have observed that even though we can protect against the existence of any replay attack by using the `new()` function that produces a fresh value at runtime confirming the freshness of time; an active intruder, performing the roles of the doctor and the patient could still be found. Thus agent identities `p` and `d` have been replaced by the intruder identity, `i`, in the last two `session()` s to detect Man-in-the-Middle (MitM) or connection hijack attacks if any such attack existed.

Chapter 6

Conclusion

We conclude our work by reflecting the contributions we hope our research provided in this constantly evolving field of EHR and considering the further works that could be done to enhance it in the future.

6.1 Contribution

The most important contribution of the paper is developing a symmetric key-based EHR management protocol. We have successfully introduced the attribute-based access control in symmetric-key solution. Although the protocol we have developed is simple and has not considered many complexities that may arise during deployment in a real-life world, but to our knowledge this is the first AVISPA model of symmetric key-based protocol that also adds attributes. In future, this model can be extended to validate other complex EHR management protocols.

In section 2.1 and 2.2 we described a general Electronic Health Record Management Architecture and proceeded to explain what features or cryptographic protocols are commonly employed in it and are thought crucial for an optimum EHR Management Architecture. In trying to enlighten the features further we delve into an exploration in section 2.3 about the classification or various types of cryptographic protocols that are found in existing EHR Management Protocols, which we came across during our research. From there we moved on to provide a background idea about AVISPA (the tool which we would use to validate our researched model) and how it functions, in section 3.1 and 3.2.

Having laid out all the pre-requisite information, we progressed on to explain our Proposed Electronic Health Record (EHR) Management Protocol with all its requirements in section 4.1 and 4.2. The protocol is broken down in details using message sequences exchange we want for a successfully operational EHR Management Protocol.

Subsequently we implemented our hypothesized Protocol in AVISPA in section 5.2 using HLPSL protocol specification and validated the output of our model in section 5.4 using a graphical tool, SPAN that executed the HLPSL protocol specification. SPAN provided the message sequence charts (MSC) that helped in our analysis by showing the simulations steps of all messages we were theorizing, Thus aiding us to conclude our research by validating our Proposed Electronic Health Record (EHR) Management Protocol.

6.2 Limitations

In our proposed protocol model design we did not verify authentication property directly of the doctor or the patient and thus intruder could fake the identity of these two communicating entities. Moreover we have not carried out any performance study. We have also not deduced how much overhead will be added. Furthermore our work was limited to formal theoretical verification simulating the proposed protocol in AVISPA tool. We have not developed any prototype or practically test our protocol in real life scenario.

6.3 Future Work

In our model validation we have focused mainly to ensure secrecy property and provide authorization but future works could include other cryptographic features validation e.g., authentication. We have observed that in our proposed electronic health record (EHR) management protocol even though we can protect against the existence of any replay attack, an active intruder, playing the roles of the doctor and the patient could still be found. Future work could upgrade the protocol, by providing fine-grained access control or strengthening it by ensuring the roles of the doctor and the patient are authenticated before any message exchange occurs. This would lead to the elimination of the possibility of any active intruder faking their roles.

Other works could include validation of authentication with the help of trusted third party or could incorporate public key encryption to use digital signatures for authentication.

References

[1] T. Hupperich, H. Löhr, A. Sadeghi, and M. Winandy, "Flexible patient-controlled security for electronic health records," in *Proceedings of the 2nd ACM SIGHIT symposium on International health informatics - IHI '12*, 2012, p. 727.

[2] S. Narayan, M. Gagné, and R. Safavi-Naini, "Privacy preserving EHR system using attribute-based infrastructure," in *Proceedings of the 2010 ACM workshop on Cloud computing security workshop - CCSW '10*, 2010, p. 47.

[3] K. T. Win, "A review of security of electronic health records," *Management*, vol. 34, no. 1, pp. 13–19, 2005.

[4] B. Katt, R. Breu, M. Hafner, T. Schabetsberger, R. Mair, and F. Wozak, "Privacy and access control for IHE-based systems," *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering*, vol. 1 LNICST, pp. 145–153, 2009.

[5] N. Shang, M. Nabeel, F. Paci, and E. Bertino, "A privacy-preserving approach to policy-based content dissemination," in *Proceedings - International Conference on Data Engineering*, 2010, pp. 944–955.

[6] B. Blobel, "Comparing approaches for advanced e-health security infrastructures," *International Journal of Medical Informatics*, vol. 76, no. 5–6, pp. 454–459, 2007.

[7] T. D. Gunter and N. P. Terry, "The Emergence of National Electronic Heath Record Architechtures in the United States and Australian: Models, Costs, and Questions," *Journal of Medical Internet Research*, vol. 7, no. 1. JMIR Publications Inc., Toronto, Canada, p. e3, 2005.

[8] E. Geron and A. Wool, "CRUST: Cryptographic remote untrusted storage without public keys," *International Journal of Information Security*, vol. 8, no. 5, pp. 357–377, 2009.

[9] E. Goh, H. Shacham, N. Modadugu, and D. Boneh, "SiRiUS: Securing remote untrusted storage," in *Proceedings of the The 10th Annual Network and Distributed*

System Security Symposium - NDSS '03, 2003, no. 121481, pp. 131–145.

[10] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based Encryption for Fine-grained Access Control of Encrypted Data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security*, 2006, pp. 89–98.

[11] A. Sahai and B. Waters, "Fuzzy Identity-Based Encryption," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2005, pp. 457–473.

[12] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuellar, P. H. Drielsma, P. C. Heám, O. Kouchnarenko, J. Mantovani, S. Mödersheim, D. von Oheimb, M. Rusinowitch, J. Santiago, M. Turuani, L. Viganò, and L. Vigneron, "The AVISPA Tool for the Automated Validation of Internet Security Protocols and Applications," in *CAV'05: Proceedings of the 17th international conference on Computer Aided Verification*, 2005, pp. 281–285.

[13] S. Islam, "Security Analysis of LMAP Using AVISPA," *International Journal of Security and Networks.*, vol. 9, no. 1, pp. 30–39, 2014.

[14] S. Islam, "Security Property Validation of the Sensor Network Encryption Protocol (SNEP)," *Computers*, vol. 4, no. 3, pp. 215–233, 2015.

[15] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, and A. Rabkin, "A View of Cloud Computing Clearing the clouds away from the true potential and obstacles posed by this computing capability," in *Communications of the ACM*, 2010, vol. 53, no. 4, pp. 50–58.

[16] M. Armbrust, A. Fox, R. Griffith, A. Joseph, and RH, "Above the clouds: A Berkeley view of cloud computing,"in *University of California, Berkeley, Technical Report UCB*, pp. 07–013, 2009.

[17] E. Palm, "Rehabilitation at Home of Patients with Neglect Using a Telemedical Intervention: a Security Perspective," 2016.

[18] M. Pura, V. Patriciu, and I. O. N. Bica, "Formal verification of secure ad hoc routing protocols using AVISPA : ARAN case study 2 Related Work Framework Formal

Analysis," in *Proceedings of the 4th EUROPEAN COMPUTING CONFERENCE*, pp. 200–206.

[19] M. Pura, V. Patriciu, and I. O. N. Bica, "Modeling and formal verification of implicit on- demand secure ad hoc routing protocols in HLPSL and AVISPA," *INTERNATIONAL JOURNAL OF COMPUTERS AND COMMUNICATIONS.*, vol. 3, no. 2, pp. 25–32, 2009.

[20] J. Benaloh, M. Chase, E. Horvitz, and K. Lauter, "Patient controlled encryption: ensuring privacy of electronic medical records," in *CCSW '09 Proceedings of the 2009 ACM workshop on Cloud computing security*, 2009, pp. 103–114.

[21] J. Heurix and T. Neubauer, "Privacy-preserving storage and access of medical data through pseudonymization and encryption," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6863 LNCS, pp. 186–197, 2011.

[22] A. K. Das and A. Goswami, "A secure and efficient uniqueness-and-anonymity-preserving remote user authentication scheme for connected health care," *Journal of Medical Systems*, vol. 37, no. 3, pp. 1–16, 2013.

[23] S. Islam and S. Farzana, "Secured Electronic Health Record Management Protocol," in *ICISPC 2017: Proceedings of the International Conference on Imaging, Signal Processing and Communication*, 2017, pp. 158–162.

[24] "Integrating the Healthcare Enterprise (IHE)." [Online]. Available: https://www.ihe.net/.

[25] HealthIT.gov, "Guide to Privacy and Security of Health Information," no. April, pp. 27–40, 2013.

[26] United States. Department of Health and Human Services, "HIPAA administrative simplification regulation text," vol. 164, pp. 1–115, 2013.

[27] "Health Information Privacy." [Online]. Available: https://www.hhs.gov/hipaa/for-professionals/index.html.

[28] J. J. P. C. Rodrigues, I. De La Torre, G. Fernández, and M. López-Coronado, "Analysis of the security and privacy requirements of cloud-based electronic health records systems," *Journal of Medical Internet Research*, vol. 15, no. 8, pp. 1–9, 2013.

[29] R. C. Barrows and P. D. Clayton, "Privacy, confidentiality, and electronic medical records.," *Journal of the American Medical Informatics Association : JAMIA*, vol. 3, no. 2, pp. 139–48, 1996.

[30] W. Stallings, "A Model For Network Security," in *Network Security Essentials: Applications and Standards*, FOURTH., Prentice Hall,1 Lake Street, Upper Saddle River, NJ: Pearson Education, Inc, 2011, p. 19.

[31] J. L. Fernandez-Aleman, I. C. Senor, P. A. O. Lozoya, and A. Toval, "Security and privacy in electronic health records: A systematic literature review," *Journal of Biomedical Informatics*, vol. 46, no. 3, pp. 541–562, 2013.

[32] W. Stallings, "Symmetric Encryption Principles," in *Network Security Essentials: Applications and Standards*, FOURTH., Prentice Hall,1 Lake Street, Upper Saddle River, NJ: Pearson Education, Inc, 2011, p. 29.

[33] U. Maurer, "Modelling a Public-Key Infrastructure," in *European Symposium on Research in Computer Security (ESORICS 96)*, Berlin, 1996, pp. 1–26.

[34] "What is Public Key Infrastructure (PKI)." [Online]. Available: https://www.thalesesecurity.com/faq/public-key-infrastructure-pki/what-public-key-infrastructure-pki.

[35] R. P. Fulare and A. V Sakhare, "Secure Authentication Technique in Wireless Integrated Sensor Network : Virtual Certificate Authority," vol. 3, no. 3, pp. 501–508, 2014.

[36] J. Hu, H. H. Chen, and T. W. Hou, "A hybrid public key infrastructure solution (HPKI) for HIPAA privacy/security regulations," in *Computer Standards and Interfaces*, 2010, vol. 32, no. 5–6, pp. 274–280.

[37] L. Huang, H. Chu, C. Lien, C. Hsiao, and T. Kao, "Privacy preservation and

information security protection for patients' portable electronic health records," in *Computers in Biology and Medicine*, 2009, vol. 39, no. 9, pp. 743–750.

[38] B. S. Elger, J. Iavindrasana, L. Lo Iacono, H. Müller, N. Roduit, P. Summers, and J. Wright, "Strategies for health data exchange for secondary, cross-institutional clinical research," in *Computer Methods and Programs in Biomedicine*, 2010, vol. 99, no. 3, pp. 230–251.

[39] W. D. Yu and M. A. Chekhanovskiy, "An electronic health record content protection system using SmartCard and PMR," in *HEALTHCOM 2007: Ubiquitous Health in Aging Societies - 2007 9th International Conference on e-Health Networking, Application and Services*, 2007, pp. 11–18.

[40] R. Zhang and L. Liu, "Security models and requirements for healthcare application clouds," in *Proceedings - 2010 IEEE 3rd International Conference on Cloud Computing, CLOUD 2010*, 2010, pp. 268–275.

[41] J. Sun and Y. Fang, "Cross-domain data sharing in distributed electronic health record systems," in *IEEE Transactions on Parallel and Distributed Systems*, 2010, vol. 21, no. 6, pp. 754–764.

[42] S. Sucurovic, "An approach to access control in electronic health record," *Journal of Medical Systems*, vol. 34, no. 4, pp. 659–666, 2010.

[43] M. Jafari, R. Safavi-Naini, C. Saunders, and N. P. Sheppard, "Using digital rights management for securing data in a medical research environment," in *Proceedings of the tenth annual ACM workshop on Digital rights management - DRM '10*, 2010, p. 55.

[44] W. K. Hon, C. Millard, and I. Walden, "The problem of 'personal data' in cloud computing: what information is regulated?--the cloud of unknowing," in *International Data Privacy Law*, 2011, vol. 1, no. 4, pp. 211–228.

[45] F. Daryabar, A. Dehghantanha, B. Eterovic-Soric, and K. K. R. Choo, "Forensic investigation of OneDrive, Box, GoogleDrive and Dropbox applications on Android and iOS devices," *Australian Journal of Forensic Sciences*, vol. 48, no. 6, pp. 615–642, 2016.

[46] T. AVISPA, “The AVISPA Library.” [Online]. Available: http://www.avispa-project.org/library/avispa-library.html.

[47] Y. Glouche, T. Genet, O. Heen, and O. Courtay, *A Security Protocol Animator Tool for AVISPA*. IRISA/Université de Rennes 1: Rennes, France.

[48] Y. Glouche, T. Genet, and E. Houssay, *SPAN: Security Protocol ANimator for AVISPA*, User Manua. IRISA/Université de Rennes 1: Rennes, France, 2008.

[49] T. The AVISPA, “HLPSL Tutorial-Beginner’s Guide to Modelling and Analysing Internet Security Protocols,” 2006. [Online]. Available: http://www.avispa-project.org/package/tutorial.pdf.

[50] D. Basin and M. Sebastian, “OFMC : A symbolic model checker for security protocols,” 2004.

[51] M. Turuani, “The CL-Atse Protocol Analyser,” in *Proceedings of the 17th international conference on Term Rewriting and Applications.RTA*, Loria-INRIA, Vandoeuvre-lès-Nancy, France, 2006, Vol 4098., pp. 277–286.

[52] A. Armando, R. Carbone, and L. Compagna, *SATMC : a SAT-based Model Checker for Security-critical Systems*, Volume 322. Lisbon, Portugal: Springer-Verlag, 2004.

[53] Y. Boichut, P.-C. Heam, O. Kouchnarenko, and F. Oehl, “Improvements on the Genet and Klay Technique to Automatically Verify Security Protocols,” in *Proceedings of International Workshop on Automated Verification of Infinite-State Systems (AVIS’2004), joint to ETAPS’04*, 2004, no. March 2015, pp. 1–11.

[54] F. Jacquemard, M. Rusinowitch, and L. Vigneron, “Compiling and verifying security protocols,” in *Proceedings of the 7th international conference on Logic for programming and automated reasoning, LPAR 2000*, 2000, vol. 6397, pp. 131–160.

[55] T. Genet, *A Short SPAN + AVISPA Tutorial*. IRISA/Universit´e de Rennes 1, 2015.

[56] T. T. AVISPA, “AVISPA v1. 1 user manual,” *2013*, vol. 1, p. 20, 2006.

[57] T. The AVISPA, “Deliverable D6 . 1 : List of selected problems Deliverable details,” 2001.

[58] D. Dolev, “On the Security of Public Key Protocols,” in *IEEE Transactions on Information Theory*, 1983, vol. 29, no. 2, pp. 198–208.