BOTSWANA INTERNATIONAL UNIVERSITY
OF SCIENCE & TECHNOLOGY

SECURITY ENHANCEMENT IN HEALTHCARE CLOUD COMPUTING

BY

GORATA MOLAMOGANYI
REG NO:14100115
SUPERVISOR: Dr ADAMU MURTALA ZUNGERU

2018

SECURITY ENHANCEMENT IN HEALTHCARE CLOUD COMPUTING

By

Gorata Molamoganyi
Reg. No: 14100115
MEng Computer and Telecommunications Engineering

Department of Electrical, Computer and Telecommunications Engineering,
Faculty of Engineering,
Botswana International University of Science and Technology (BIUST)
Gorata.molamoganyi@studentmail.biust.ac.bw

A Dissertation/Thesis Submitted to the Faculty of Engineering in Partial Fulfilment of the Requirements for the Award of the Degree of Master of Engineering in Computer and Telecommunications of BIUST

Supervisor: Dr. Adamu Murtala Zungeru
Department of Electrical, Computer and Telecommunications Engineering,
Faculty of Engineering, BIUST
Zungerum@biust.ac.bw, +2674931655
Signature: _____ Date: 12/12/2018

Supervisor: Dr. Mmoloki Mangwala
Department of Electrical, Computer and Telecommunications Engineering,
Faculty of Engineering, BIUST
Mangwalam@biust.ac.bw, +2674931666
Signature: _____ Date: 12/12/2018

July, 2018

DECLARATION AND COPYRIGHT

I, **GORATA MOLAMOGANYI**, declare that this dissertation/thesis is my own original work and that it has not been presented and will not be presented to any other university for a similar or any other degree award.

Signature .....

This dissertation/thesis is copyright material protected under the Berne Convention, the Copyright Act of 1999 and other international and national enactments, in that behalf, on intellectual property. It must not be reproduced by any means, in full or in part, except for short extracts in fair dealing; for researcher private study, critical scholarly review or discourse with an acknowledgement, without the written permission of the office of the DVCAA, on behalf of both the author and the BIUST.

CERTIFICATION

The undersigned certifies that they have read and hereby recommends for acceptance by the Faculty of Engineering a dissertation/thesis titled: **Security enhancement in healthcare cloud computing**, in fulfilment of the requirements for the degree of Master of Engineering in Computer and Telecommunications Engineering of BIUST.

..... 12/12/2018

Dr. Adamu Murtala Zungeru

(Supervisor)

Date: .....

.....

Dr. Mmoloki Mangwala

(Supervisor)

Date: 12/12/2018

DEDICATION

I would like to dedicate this thesis to the almighty God. Without him I would not have had the strength and wisdom to complete this thesis, and I would also like to dedicate it to my family, for their support and understanding when I could not spend time with them.

ACKNOWLEDGEMENT

Firstly, I would like to express my sincere gratitude to my supervisors, Dr Adamu Murtala Zungeru and Dr Mmoloki Mangwala, for their unwavering support and encouragement to complete this thesis and taking time out of their busy schedule to review and give guidance on my work. Without them my thesis would not have be completed. To all the staff and students of the Botswana International University of Science and Technology, I thank you.

I also thank the Human Resources Development Council (HRDC) Botswana for assisting me with some funds to buy the laptop and to carry out data collection.

Last but not least, I would like to thank my family for their support, more especially my mother for her encouragement whenever I felt like giving up and also my son for understanding when I could not give him attention.

ABSTRACT

Cloud computing has brought so many benefits to the healthcare industry. Information about patients can now be stored in a central place where it can be shared amongst all health organisations and easily accessed, hence saving time and costs. However, there are issues with sharing such information on the cloud since it is highly sensitive information. Security, availability and scalability are major factors in the cloud computing environment. This study presents the current state-of-the-art research in this field by focusing on several shortcomings of current healthcare solutions and standards. This work proposed a system (self-encrypting system) that will enhance security at the client's side by encrypting data before it is being sent to the cloud. This system uses random generated numbers to give access to the user at the cloud's side. The system is intended to be linked to the cloud in such a way that, before the client submits the data to the cloud, the data will go through that system for encryption. The research presents algorithms on how the system will work. A prototype of the system was developed using NetBeans platform. The results of the prototype show encryption and decryption of a file using Triple des encryption (3DES). An asymmetric encryption standard (RSA) algorithm was used for key distribution to safeguard the file against man-in–the–middle attacks. Different encryption algorithms were also evaluated using different parameters. Three encryption algorithms (3DES, DES and RSA) were compared and Triple DES (3DES) proved to be the fairest among them all since it's more secure and efficient as compared to RSA which is slow and DES which is not secure. However, the availability of information and nodes at the cloud cannot be overlooked, hence this work further proposed a mathematical model for measuring the availability of data and nodes. In future the system will be tested in a real working environment to test its effectiveness against vulnerabilities. Lastly, an input function for the graph and weight of graph will be improved to achieve the complete availability model.

Table of Contents

DECLARATION AND COPYRIGHT i

CERTIFICATION ii

DEDICATION iii

ACKNOWLEDGEMENT iv

ABSTRACT v

LIST OF FIGURES x

LIST OF TABLES xi

LIST OF ABBREVIATIONS xii

CHAPTER 1 1

INTRODUCTION 1

1.1 BACKGROUND OF THE STUDY 1

1.2 HIERARCHICAL ARCHITECTURE OF CLOUD COMPUTING 1

1.2.1 Service delivery models 2

1.2.2 Deployment models 2

1.3 SECURITY AND PRIVACY ISSUES IN HEALTH-CLOUD 5

1.3.1 Confidentiality 5

1.3.2 Privacy 6

1.3.3 Multi-tenancy 6

1.3.4 Integrity 6

1.4 MOTIVATION 6

1.5 STATEMENT OF THE PROBLEM 7

1.6 AIM 7

1.7 OBJECTIVES 7

1.8 RESEARCH QUESTIONS 7

1.9 CONTRIBUTIONS 7

1.10 RESEARCH PUBLICATIONS 8

1.11 THESIS OUTLINE 9

CHAPTER 2 10

SECURITY AND AVAILABILITY IN HEALTHCARE CLOUD 10
2.1. INTRODUCTION 10
2.2. SECURITY ISSUES 10
2.2.1 Data security 10
2.2.2 Network security 12
2.2.3 Application Security 12
2.2.4 Virtualisation Security 13
2.2.5 Identity Management 14
2.3 AVAILABILITY ISSUES 14
2.3.1 Flooding Attack via Bandwidth Starvation 15
2.3.2 Denial of service attacks 15
2.4 SECURITY ISSUES ACCORDING TO THE CLOUD SERVICE MODELS 15
2.4.1 Infrastructure as a Service (IaaS) 15
2.4.2 Software as a Service (SaaS) 16
2.4.3 Platform as a service (PaaS) 17
2.5 MODELS EMPLOYED TO ADDRESS SECURITY ISSUES 17
2.6 MODELS EMPLOYED TO ADDRESS AVAILABILITY ISSUES 20
2.7 ENCRYPTION 22
2.7.1Evaluation parameters for encryption algorithms 22
2.8 ENCRYPTION STANDARDS 22
2.8.1 Rivest Shamir Adelman (RSA) 22
2.8.2 Advanced Encryption Standard (AES) 22
2.8.3 Data Encryption Standard (DES) 23
2.9 CONCLUSION 23
CHAPTER 3 24
SECURITY AND AVAILABILITY MODELS IN HEALTHCARE CLOUD 24
3.1 INTRODUCTION 24
3.2 SECURITY REQUIREMENTS FOR THE HEALTHCARE CLOUD 24
3.2.1 Data Confidentiality 24

3.2.2 On-demand Revocation 24
3.2.3 Write Access Control 24
3.2.4 Scalability, Efficiency and Usability 25
3.2.4 Detection and Prevention of Security Attacks in the System 25
3.3 DATA SECURITY MODELS 25
3.3.1 Electronic Health Record Security Reference Model 25
3.3.2 Patient-centric, secure and scalable PHR sharing model 26
3.3.3 A Novel Framework with Cloud-Based Privacy-Aware Role Based Access Control (CPRBAC) Model 27
3.3.4 Secure Dynamic Access Control Scheme of PHR in Cloud Computing 28
3.3.5 Active Data Cube Model for EHR Protection(ADCu) 29
3.4 ANALYSIS OF THE MODELS 30
3.5 PROPOSED SECURITY ENHANCEMENT MODEL FOR HEALTHCARE CLOUD (SELF-ENCRYPTING SYSTEM) 32
3.5.1 An Algorithm of the Proposed Model 32
3.5.2 Algorithm for Uploading and Encrypting 34
3.5.3 Algorithm for Requesting Data 35
3.6 AVAILABILITY IN CLOUD COMPUTING 37
3.6.1 Information Availability in Cloud Computing 37
3.7 Node Availability 38
3.8 CONCLUSION 42
IMPLEMENTATION OF THE SELF ENCRYPTING SYSTEM 43
4.1 INTRODUCTION 43
4.2 DEVELOPMENT TOOLS AND ENVIRONMENT 43
4.3 THE SELF-ENCRYPTING SYSTEM 43
4.3.1 Encryption of the Original File 43
4.3.2 Decryption of the File 44
4.3.3 How the System Addresses Privacy and Confidentiality Issues 45
4.4 CONCLUSION 47

RESULTS AND ANALYSIS 48
5.1 INTRODUCTION 48
5.2 RESULTS FROM AVAILABILITY MODEL 48
5.3 RESULTS FROM THE PROPOSED SELF -ENCRYPTION SYSTEM 49
5.3.1 Operation of Triple DES (3DES) 49
5.3.2 Random Key Generation 51
5.3.3 Key Distribution 51
5.3.4 Securing the Channel for Transmission of Keys 52
5.3.5 Encryption 53
5.3.6 Decryption 54
5.4: COMPARISON OF THE TRIPLE DES WITH OTHER ALGORITHMS 54
5.4.1 Encryption Time 55
5.4.2 Decryption Time 56
5.4.3 Comparative analysis of results from other systems with the result from the self-encrypting system 57
5.5 CONCLUSION 60
CHAPTER 6 61
CONCLUSIONS AND FUTURE WORKS 61
6.1 CONCLUSIONS 61
6.2 FUTURE WORKS 61
6.2.1Homomorphic encryption 61
6.2.2 Honey encryption 62
6.2.3implementing the algorithm in parallel on a multicore GPU using open CL 62
REFERENCES 63

LIST OF FIGURES

Fig. 1.1: Hierarchical view of service models in cloud computing.....3

Fig. 1.2: Global health village model for continental health cloud 3

Fig. 1.3: Global health village model for hybrid health cloud 4

Fig. 1.4: Information gathering in health cloud system.....4

Fig. 3.1: Electronic Health Record Security Reference Model System Components.....28

Fig. 3.2: Patient Centric, Secure and Scalable PHR Sharing Model.....29

Fig. 3.3: Components of CPRBAC model.....30

Fig. 3.4: Access Environment for Secure Dynamic Access Control Scheme.....31

Fig. 3.5: Structure of an Active EHR (ADCU).....32

Fig. 3.6: System architecture of the proposed system.....35

Fig. 3.7: Flowchart of the proposed system.....36

Fig. 3.8: Flowchart for uploading and encrypting data.....37

Fig. 3.9: Flowchart for requesting data.....38

Fig. 3.10: Performance and availability of machines in the system.....41

Fig. 3.11: The Gb _node graph 44

Fig. 5.1: Probability of at least j machines in operarion.....52

Fig. 5.2: Operation of Triple DES.....54

Fig. 5.3: Random generated keys.....54

Fig. 5.4: Key Exchange.....55

Fig. 5.5: Key distribution in RSA.....56

Fig. 5.6: Encryption Results.....56

Fig. 5.7: Encryption Results in Binary.....57

Fig. 5.8: Decryption results.....57

Fig. 5.10: Encryption time.....59

Fig. 5.11: Decryption time.....59

LIST OF TABLES

Table 3.1: Analysis of Security models.....34

Table 4.5: Code description for encryption and decryption.....49

Table 5.1: Computational overhead encryption in Triple DES 56

Table 5.2: Computational overhead for encryption in DES.....57

Table 5.3: Computational overhead encryption in RSA 57

Table 5.4: Comparative analysis of other system with the self-encrypting system.....59

LIST OF ABBREVIATIONS

ABE: Attribute Based Encryption

CBIHS: Community-Based In-Home Services

CDO: Healthcare Delivery Organisation

CDS: Cloud Data Service

DES: Data Encryption System

DO: Data Owner

DSS: Data Sharing Service

EHR: Electronic Health Card

ESPAC: Enabling Security and Patient-Centric Access Control for E-health Cloud Computing

GUI: Graphical User Interface

IaaS: Infrastructure as a Service

IDC: Internet Data Centres

MFC: Mean Failure of Cost

MTBF: Mean Time between Failures

MTTF: Mean Time to Failure

MTTR: Mean Time to Recover

PaaS: Platform as a Service

PHI: Protected Health Information

PHR: Patient Health Record

PK: Private Key

PKI: Public Key Infrastructure

RBAC: Role-Based Access Control

SaaS: Software as a Service

SECHA: Secured E-health in cloud using homomorphic encryption and access control

TPM: Trusted Platform Module

TTP: Trusted Third Party

TVD: Trusted Virtual Domain

TVDC: Trusted Virtual Data Centre

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND OF THE STUDY

Healthcare industry is going through the pressure of delivering the best service to patients and other healthcare professionals all over the world. The conventional way of storing healthcare records has proved to be inefficient. It is time consuming and costly. Introduction of cloud computing in the healthcare has brought new opportunities like better ways to access healthcare data easily, opportunities for new businesses in the healthcare industry [**1**].

Cloud computing is used for sharing resources over the internet using virtual machines rather than physical machines [**1**]. These resources can be provided quickly with little need for interaction with the service provider. These shared resources can be servers, storage applications and services [**2**].

The application of cloud computing in the health sector is called healthcare cloud computing. A healthcare cloud is made up of computers and servers which work together to meet the needs of a healthcare industry [1]. However, as much as the healthcare cloud brings many benefits, they also come with some challenges. Because the health sector deals with sensitive information, there is a need for increased security and privacy levels so that this information does not fall into the wrong hands. The availability of the information to the users is also very important. Information needs to be accessible whenever it is needed. Security needs to be implemented on the cloud and the client side. Less research has been done on the security at the client's side. As much as information might be accessed by unauthorized users in the cloud, this can also happen at the client's side.

1.2 HIERARCHICAL ARCHITECTURE OF CLOUD COMPUTING

This defines the cloud computing model which consists of five characteristics, three service delivery models and four deployment models. These five characteristics of cloud computing are, location–independent resource pooling, on-demand self-service, rapid elasticity, broad network access and measured service. These characteristics represent the first layer in the cloud environment. These five characteristics represent the first layer in the cloud environment structure [**3**].

1.2.1 Service delivery models

Cloud computing is made up of three service delivery models. These models offer different services to the cloud users and they are Infrastructure as a service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) [3].

1.2.1.1 Infrastructure as a Service (IaaS)

This is the lowest layer which is concerned with providing the support services to the cloud infrastructure [4]. In this layer, the consumer does not have control over the core cloud infrastructure; instead, they can control the operating systems, storage and deployed applications and run the software. In addition to these, they have limited rights to control the networking components [5].

1.2.1.2 Platform as a Service (PaaS)

This is the middle layer which offers the user the flexibility to develop applications using tools provided by the PaaS provider. In this layer the consumer is given the rights to control the deployed applications and to change the configurations for the environment where applications are hosted. Moreover, the consumer is not allowed to control the network, servers, operating system and storage which forms the core cloud infrastructure [6].

1.2.1.3 Software as a Service (SaaS)

This is the upper layer where applications are held by a vendor or service provider and availed to customers over a network, typically the internet. In this model the consumer has the freedom of using the service provider's applications which are running on the cloud. These applications can be accessed from different devices or client interfaces such as browsers and program interfaces. Fig. 1.1 shows the various applications a consumer can use [6].

1.2.2 Deployment models

Cloud computing also consists of deployment models. Companies have the freedom of choosing a deployment model suitable for their business, operational and technical requirements. There are four primary cloud deployment models: Private, country, continent and hybrid model [7]. In this study, we shall look at cloud computing in the healthcare domain and classify them on how they are related. Fig. 1.2 and 1.3 shows private, country, continent and hybrid deployment models.

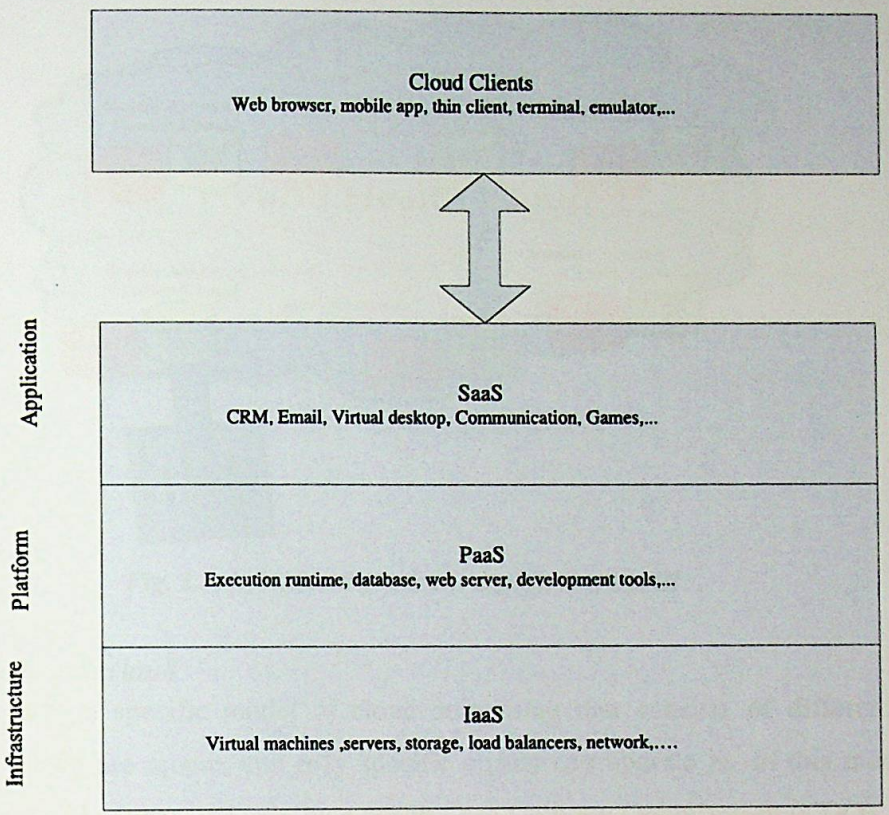


Fig. 1.1: Hierarchical view of service models in cloud computing

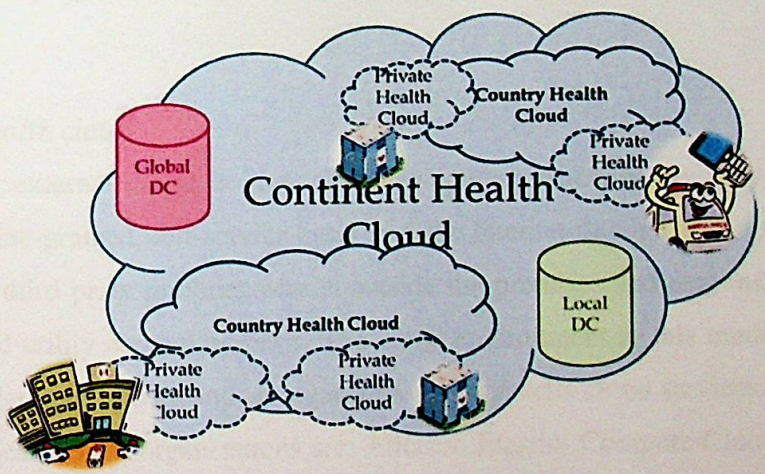


Fig. 1.2: private and country deployment model

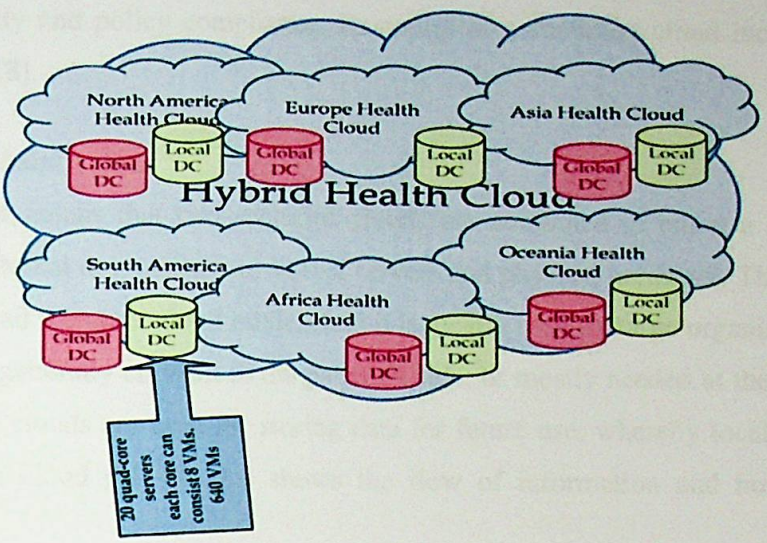


Fig. 1.3: hybrid and continent deployment model

1.2.2.1 Private health cloud

A private cloud is a specific model of cloud computing that consists of different cloud environments which are secure, and only specific clients can operate in. In this model, the cloud infrastructure is controlled only by a health Care Delivery Organization (CDO). Private cloud provides computing power in an environment populated with virtual machines by using a group of physical resources just like with other deployment models. Nonetheless, only one organization has the rights to access the resources thus having control over the resources and privacy [**8**].

1.2.2.2 Public health cloud

Public cloud or external cloud defines cloud computing whereby the way resources are provided on a fine-grained, self-service basis over the Internet through web applications/web services from a third party provider who is outside the premises and responsible for billing on a fine-grained utility computing basis. The cloud environment in this model is accessible to the public or large industries and an organization which sells cloud services is the one who owns it. Examples of such organizations are: Amazon Elastic- Compute Cloud, IBM's Blue-Cloud, Sun Cloud, and Google App Engine [**8**].

1.2.2.3 Community health cloud

In this model the cloud infrastructure is owned by several healthcare delivery organizations (CDO'S) and supports a particular group of people [**9**]. It contains both features of public and private cloud models [**7**]. A community cloud may be established whereby several organizations have the same desires and want to share the infrastructure so as to benefit from cloud computing. Although the community health cloud is more expensive than other deployment models because the costs are distributed to fewer users, it offers a high level of

privacy, security and policy compliance. Examples of community cloud includes Google's "Gov Cloud" [8].

1.2.2.4 Hybrid health cloud

A hybrid cloud means that two secluded clouds are combined in either a public, private, internal and external or a mixture of virtual servers and physical hardware. The model has all the features of all the other cloud models and it is mostly used by large organizations [7]. The information is generally eminent at the patient's side, or mostly needed at the patient's ends. Hybrid storage clouds are used for storing data for future use, whereby local data is backed up to a public cloud [8]. Fig 1.4 shows the flow of information and how the data are collected.

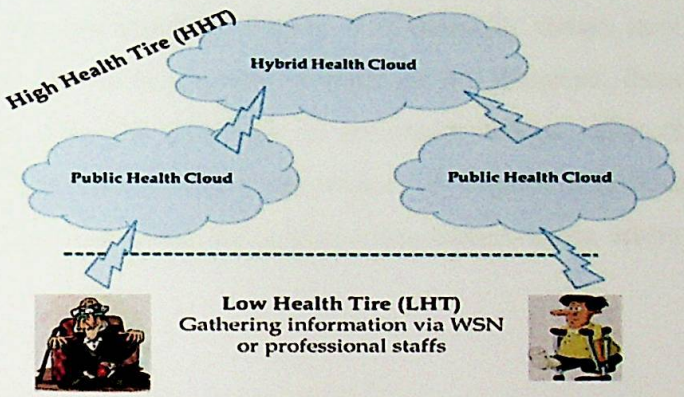


Fig.1.4: Information gathering in health cloud system

1.3 SECURITY AND PRIVACY ISSUES IN HEALTH-CLOUD

As much as cloud computing has some beneficial factors, it also comes with some security risks. The healthcare deals with some sensitive information about the patient which is needed to be handled with greater care so as not to fall on wrong hands. The main issues in healthcloud security are; confidentiality, privacy, multi-tenancy and integrity.

1.3.1 Confidentiality

Confidentiality refers to only authorized parties of the system having the ability to access protected data. The risk of data being compromised in the cloud is increasing due to a high number of parties, devices and applications using the cloud. The more the number of parties, the high risks of unauthorized parties accessing data. According to Thilakanathan in [10], the threats to confidentiality can be divided into three; insider attacks, outsider attacks and data leakage. Inside attacks can be in the form of malicious cloud service provider accessing data in the cloud, malicious cloud customer and the malicious third-party user gaining access to data. Outsider attacks may be in the form of remote software attacks on the cloud infrastructure and remote hardware attack against the cloud. Data leakage may be due to

failure of security access rights across multiple domains and failure of electronic and physical transport systems for cloud data and backups [11].

1.3.2 Privacy

Privacy is the need to regulate the exposure of personal information. The cloud poses so many challenges concerning the privacy of information segregated in various locations in the cloud, hence increasing the likelihood of confidentiality and privacy breaches and the legal issues [10].

1.3.3 Multi-tenancy

Multi-tenancy refers to sharing resources in a cloud environment, e.g., multiple users use the same resources at the network level, host level and application level. Multitenancy presents a number of privacy and confidentiality threats. Confidentiality threats may be in the form of data leakages which may occur because of hardware not detached. Reusable entities must also be carefully controlled in case they do not create serious vulnerability through data remanence. This is whereby a customer can request for a space to store data from the cloud provider and then they will search for sensitive information about others through scanning [12].

1.3.4 Integrity

The most important aspect of information security is integrity. Integrity refers to the modification of data by only people who have the right to do so and in an acceptable/correct way. It means guarding of information against illegal changing, deleting and even fabrication [10]. The integrity of data can be compromised when the resources are not well separated especially in complicated cloud hosting environments like SaaS which is configured to share resources amongst customers. Data segregation can be due to security parameters which are not defined correctly and virtual machines and hypervisors which are not configured properly. Poor deployment of the control procedures can pose risks to the cloud infrastructure [13].

1.4 MOTIVATION

The healthcare sector as one of the most important sectors needs to advance with technology. However, every technology has its benefits and limitations. Technologies such as cloud computing are no exception. Problems such as security and availability of information are some of the limitations of healthcare cloud. Various researchers in [14] [5] [7] have conducted research on how to tackle these problems but there are still some open issues more especially on securing the data on the client side, and when it's on transit to the cloud. This

then calls for an efficient solution to help in solving the problem. This research proposes a model to help in solving the problem. The proposed solution aims at improving the security of information on the client's side before it can be sent to the cloud. Information is encrypted at the client side so that even if it is on transit, it is still secure and furthermore calculations of the storage size are done when the data is still encrypted.

1.5 STATEMENT OF THE PROBLEM

The health sector holds very crucial information about a patient's health hence the highest level of security should be employed. Currently, there are security issues in the healthcare cloud. Patient's privacy is compromised since information can be accessed by unauthorised users on its way to the cloud [**15**]. Methods which have been employed have proven not to be strong enough hence security is compromised together with the availability of information whenever it is needed. The problem this research addresses is that of the availability and security in healthcare cloud.

1.6 AIM

The main aim of this research is to design and implement a system that will enhance security in healthcare cloud.

1.7 OBJECTIVES

1. To identify security and availability issues in healthcare cloud computing
2. To evaluate existing security models in healthcare cloud and identify the gaps
3. To come up with a mathematical model to address the availability issues
4. To design and implement a system that will address the security issues at the patient's side

1.8 RESEARCH QUESTIONS

Question 1: What are the current security issues in healthcare cloud, which hinders the successful transmission of healthcare information?

Question 2: What are the problems with the current models being employed in the healthcare cloud?

Question 3: How can these problems with the current models employed be addressed to ensure secure transmission of healthcare information?

Question 4: what are availability issues in healthcare cloud?

1.9 CONTRIBUTIONS

Security and availability are very important in cloud computing. Information should always be there when it is needed and prevented from unauthorised access. The research gives an insight on the various security and availability issues found in healthcare cloud computing

together with an analysis of different models implemented to solve these problems. Secondly, the research provides a platform to enhance security on the client's side, by proposing a system which will encrypt information before it can be routed to the cloud, hence preventing unauthorised access. Thirdly, a mathematical model is proposed to measure availability to make better dissension for the optimum of the system. All this will help to ensure a secure and available system.

1.10 RESEARCH PUBLICATIONS

To date two publications of this work has been published in international journal and conference. They are as follows;

1. **G. Molamoganyi**, A.M. Zungeru, M. Mangwala, J.M. Chuma, "Design and Implementation of Security in Healthcare Cloud Computing", Journal of Computer Sciences, Science Publications, vol. 13(2), pp. 34-47, 2017. **(Scopus Indexed)**

2. A. M. Zungeru, **G. Molamoganyi**, M. Mangwala, J. Chuma, "Healthcare Cloud Computing for Underground Wireless Sensor Networks", in: IEEE 2nd International Conference on Frontiers of Sensors Technologies (ICFST 2017), pp. 136-140, 2017, ISBN: 978-1-5090-4860-1 ©2017 IEEE, April 14-16, 2017, Shenzhen-China. **(ISI/Scopus Indexed)**

1.11 THESIS OUTLINE

1.11.1 Chapter 2:

This chapter presents a literature survey on security and availability issues in healthcare cloud computing. This chapter further presents a comparison of different models proposed by researchers in this area to solve security issues in health care cloud computing.

1.11.2 Chapter 3:

This chapter describes the overall methodology for solving the current problems in the healthcare cloud together with the models used in arriving at the solution.

1.11.3 Chapter 4:

This chapter discusses the implementation of the proposed system

1.11.4 Chapter 5:

This chapter presents the results from the availability and the security model.

1.11.5 Chapter 6:

This chapter summarises the main ideas of the thesis together with the recommendations for future work.

CHAPTER 2

SECURITY AND AVAILABILITY IN HEALTHCARE CLOUD

2.1. INTRODUCTION

The slow adoption of healthcare cloud is mostly due to security and availability. This is because cloud computing encompasses many technologies; hence issues of security from these systems and technologies are applicable to cloud computing. These issues are also inherited into the healthcare cloud [**15**]. These issues need to be tackled to put at ease the fears associated with moving to the cloud and benefitting from the improvements it brings [**16**]. This review focuses on the security issues and availability on healthcare cloud computing specifically looking arguments, methodologies, gaps and proposed solutions.

2.2. SECURITY ISSUES

In a cloud computing framework, a cloud provider creates, deploys and manages the resources and applications in the cloud. Many users' access services from the cloud provider through resource sharing hence there are bound to be security issues. Kumar et al., in [**3**] states that the main security issues are related to confidentiality, integrity and availability also known as the CIA triad. Below are the major categories of security threats in the cloud [**17**].

2.2.1 Data security

The cloud is shared by many users at different times; hence their data is also stored in that shared environment, controlled by the service provider. This makes it susceptible to malicious attacks. Data is segregated among different servers which are in geographically different countries which makes it difficult to control the transparency and the regulatory issues. This makes data protection and privacy very important issues in a cloud environment. There is need for data to be protected from the outside world to ensure that it is not corrupted. Also, there is a risk of vulnerabilities when data is shared between different systems in the cloud [**17**].

Juare [**18**] describes issues concerning data security as; how data will be secured when in transit, data –eminences, how authentication and integrity will be provided, how user's data stored in the cloud servers will be protected from attackers, and how will cloud users be able to switch from one cloud provider to another and how will they facilitate the transmission of data from one cloud provider to another [**18**].

2.2.1.1 Data Security Issues

2.2.1.1.1 Data Privacy

Privacy is the ability of an individual or group to separate themselves or information about themselves from the public. The nature of information source, whereby data comes from different sources poses a risk to the privacy of data in the cloud. Therefore, data need to be controlled, and only made available to authorised users.

2.2.1.1.2 Data Integrity

Integrity refers to the assurance that information which is stored in the cloud is not tempered with by unauthorised users, this also applies to when data is in transit. Integrity means retaining the correctness, exactness and the credibility of data throughout its lifetime [**18**]. Issues regarding data integrity are modification/fabrication, unlawful computation usually in remote servers and if data is changed while in transit by attackers. Since cloud users keep large amounts of data on cloud servers, data may be lost or modified either accidentally or maliciously. Also, rivals may initiate attacks thus making data owners to lose control of their own data.

2.2.1.1.3 Data Confidentiality

Confidentiality refers to only people who have the rights to the system having the ability to access protected data in cloud environments. In a cloud environment all the customers' activities should not be disclosed to both the cloud provider and other customers. Confidentiality is one of the major issues in the cloud environment. This is because customers carry out their data and computation activities on cloud servers, which are managed and controlled by cloud providers which are often untrustworthy [**19**]. Xiao [**19**] discussed the confidentiality threats within the cloud as cross-VM attack via Side Channels and malicious Sysadmin. Cross-VM attack is whereby the nature of multi-tenancy is exploited, by enabling VMs belonging to different customers to stay together in the same physical machine. Malicious sysadmin is whereby a system administrator (sysadmin) of the cloud provider, who has the rights to the system, can perform attacks by gaining access to the memory of virtual machines belonging to customers.

2.2.1.1.4 Data Segregation

Data within the cloud is kept in separate cloud servers in several locations; hence it is difficult to protect it from the outside world. Encryption cannot be the only solution to this problem because other users may not want to run the risk of losing their data because of encryption going wrong. Therefore, encryption need to be made available at all the stages, and the

encryption schemes needs to be designed and tested by experienced professionals so that they can be effective [**15**].

2.2.1.1.5 Data Locality

Locality of data is very important in cloud computing. Data in the cloud is segregated to different locations which are governed by different laws; hence if the data location is not safe that means the privacy and confidentiality of data is at risk because malicious insiders and external hackers can gain access into the system. The cloud users do not have any information about the location of their data hence they have no control over who can access their data or not [**15**].

2.2.1.1.6 Multi-Tenancy

Multi-tenancy allows different users to be hosted in the same environment hence if there is some corrupt activity on that environment, it will affect the rest resulting in data breaches [**14**].

2.2.2 Network security

The cloud machines residing within a cloud platform are linked through a network which also links to other machines which are not part of the cloud platform; hence, intruders can gain access into the cloud through the network thus compromising data privacy and confidentiality. The technology of virtual machines affects the security of the network severely. It is often difficult to control and monitor the network because of the Invisible network created by virtual servers [**17**]. According to [**20**] network based attacks can fall into the following categories; port scanner whereby a port on a server may be analysed to check for the vulnerabilities of the target machine thus resulting in services not being offered to the user, Botnets, whereby a botnet is used to gain access into a host machine and illegally acquire information, which it sends to the bot-master. With the establishment of a command and-control system through a bot-master, many systems can be entry points for acquiring personal information illegally [**20**]. Authors in [**21**] also explained spoofing attacks as network impersonate entities for malicious purposes. An IP spoofing attack may substitute the existing IP address in a network packet with a fake source IP address. A virtual network may be a victim of ARP spoofing, thereby causing an attacker VM to access packets of other VMs.

2.2.3 Application Security

Threats to application security can be openness, flexibility and public availability of the cloud infrastructure. The application software which are running on the cloud platforms can be an entry point for most attacks. Vulnerabilities like presence of trap doors, overflow problems

and poor-quality code are gateways for various attacks. The main issues with using cloud applications are: not being able to have direct control over the cloud environment, how to access data by the cloud vendor and housing many users in a single environment (multitenancy) [17].

2.2.4 Virtualisation Security

Virtualisation can be described as a method of running several virtual machines' operating systems which are not dependent on each other in a physical machine. The main aim of virtualisation is to assign the workload to different physical machines to attain resource sharing [22]. Virtualisation technology comes with its own share of new attacks through the hypervisor and other management components. Virtual machines may be created and returned to previous instances whenever there is a need in a cloud environment because of this movement it is difficult to achieve and sustain security which is stable [17]. Virtual technology can introduce the following risks to the cloud environment;

2.2.4.1 Risks of Resource Access Control

When resources are combined in the same logical storage, there is no isolation and security. The cloud user can only see the logical storage location and does not know the exact location of information which might be stored in another server location thus being vulnerable to leakage. Also, when the virtual platform is attacked, the administrative rights might be stolen to be used maliciously [22].

2.2.4.2 Risks of DOS Attack

Just like in a traditional network, virtualisation services can slow down or even stop due to many applications using it for processing. Distributed Denial of Service (DDoS) is a major attack to the virtualised environment. The virtual platform offers cloud users the necessary services, thus when many users or malicious applications request for services, they will use up more resources which will result on the virtualised platform not being able to run effectively hence affecting all cloud users. A malicious application using more services is very destructive to the virtualised environment; it introduces the denial of service attack.

2.2.4.3 Risks of Virtualization Platform in Building Network

Network stack sharing can be a major problem in virtualised security. The network connection between a client and a server is based on the software hubs. If two clients belong to the same network and share the same virtual interface, they can see the server and all the traffic between the networks which are connected by a virtual switch through the same physical card. Hence if all the clients and the server share the same stack, the hacker can be able to access the whole stack by attacking the client computer.

2.2.4.4 Risks of Virtualization Platform's Security Management

The virtualisation platform is built in a private cloud; hence it is physically isolated from the internet. The internet produces new and updates viruses everyday hence segregation of the virtualised platform from the internet makes it vulnerable to attacks since antiviruses cannot be updated rapidly and the virtualised platform cannot be repaired in time. Another risk is that of management. There is a change in the management of a virtualised platform and the traditional network. The manager of a virtualised network can no longer use the ordinary tools used in a traditional network to manage the network. Because the virtualised technique hides the Graphical user interface for management, kernel module and also the binary systems. Therefore, an ordinary network administrator won't see them, thus increasing the cost and difficulty of managing a virtualised environment [**22**].

2.2.5 Identity Management

A cloud service provider generates identities for each cloud user to access cloud services. Unauthorised access is still a major problem in the cloud since a malicious entity can pretend to be an authentic user to access the cloud hence impacting on the availability of services for the real user [**17**]. Identity management issues include openness, identity theft, least privileges, elevated privilege control, availability, confidentiality, integrity and trust management. Attacks to identity are; Brute-force attack whereby attackers gain access to sensitive identity credentials of the customers which are stored in the server by trying out passwords and usernames. Cookie-replay attacks whereby the attacker will steal a cookie with the correct information about the session together with the credentials to make the identity manager believe that the session is ongoing.

Data Tampering Attack which means that unauthorised users can change data related to identification of CSC in an identity data-store at Cloud [**23**].

2.3 AVAILABILITY ISSUES

The main goal of availability for cloud computing systems is to make sure that cloud users can use the cloud services at any time and at any place. Availability is very important as it forms the integral of cloud computing existence, which is to provide services whenever they are needed (on-demand service). If a certain service can no longer be provided the quality of service depreciates resulting in customers losing faith in the cloud system. A cloud computing system allows its users to have access to its applications and services from anywhere. However, there are issues with the availability of data [2]. Padhy et al., [15] explained data availability is one of the major concerns in an organization. Data owners may experience system failures because they keep data at remote systems which are owned by

others. If the cloud is no longer functioning, data will no longer be available since it is provisioned by a single service provider. The cloud application needs to ensure that enterprises are provided with full-time service and protected from resiliency to hardware and software failures, as well as denial of service attacks. Threats to availability are flooding attack via bandwidth starvation and DOS attacks [19].

2.3.1 Flooding Attack via Bandwidth Starvation

Flooding attack via bandwidth starvation (BW-DDoS) is type of denial of service attack whereby the attacker populates the victim with unwanted traffic to prevent the acceptable traffic from reaching the victim's system. The flooding attacks use different types of protocols to flood the target system. Some strong attacking agent will include privileged zombie that has complete control over its host, with the ability of sending spoofed IP packets. Furthermore, some BW-DDoS attacks create more complicated attacks using reflection and amplification techniques, which have more devastating effects on the victim and are harder to deal with [**24**].

2.3.2 Denial of service attacks

A denial of service (DOS) attack is described as an attack designed to restrict services offered by the server in the network. This attack occurs when a computer user is blocked, and it is due to malicious act by another user. Efficiency can go down if a computer hardware, software and data are not available even if nothing is destroyed [**25**]. The common DOS attacks are those that target the computer network or connectivity. The bandwidth attacks populate the network with large volumes of traffic so that authorised users cannot be able to use network resources. Hence, the computer no longer accepts request from users [**26**].

2.4 SECURITY ISSUES ACCORDING TO THE CLOUD SERVICE MODELS

Security issues in the cloud can also be categorized according to the three cloud service delivery models.

2.4.1 Infrastructure as a Service (IaaS)

In a virtualized environment the IaaS provide a collection of resources like servers, storage, networks and other computing resources. In this model users are flexible to run any software and controlling it fully as well as managing the given resources, hence security can also be controlled much better [**26**].

Virtual Machine security–the virtual machine is affected by the same security threats as the traditional servers hence they need to be protected either by using the commonly used methods or the cloud-based methods. Threats like malware and viruses are common in the

virtual machines. In this layer securing the virtual machine is solely the responsibility of the cloud consumers, hence they can apply their own security controls according to what they need and risk levels.

Securing VM images repository–virtual machines can still be at risk even if they are offline as compared to physical. This results in VM images being attacked by introducing malicious codes in their files or even stealing the file itself. Securing the virtual machine repository is solely the responsibility of the cloud service provider.

Virtual network security–Network VM attacks are a result from sharing the network infrastructure with different occupants using vSwitch.

Hypervisor security–a hypervisor controls any access to the physical servers by the virtual machines. It maps from the physical resources to virtualized resources and vice versa. If a hypervisor is compromised the security of the VM's is violated because all the operations of the VM's are traced. Any threat to the hypervisor disturbs the security of the VMs because all the activities carried out by the VM's become traced unencrypted. Providing security of the hypervisor is the responsibility of cloud providers and the service provider [**27**].

2.4.2 Software as a Service (SaaS)

SaaS is responsible for providing application services whenever they are needed such as email, conferencing software and business applications. The SaaS model is built on top of PaaS and IaaS and thus it inherits all the security issues pertinent to both. These issues include data management issues like data locality, integrity, access, confidentiality and backups. In this model both the cloud provider and the service providers share the responsibility of maintaining and enforcing security. The users of this model have less control over the security of the model [**26**].

Web application vulnerability scanning – those web applications which are going to run on the cloud infrastructure should be checked for malicious threats and validated though the use of web application scanners. And those scanners should be updated so that they can be able to catch the recently discovered threats and their attack paths which are stored in the National Vulnerability Database (NVD) and the Common Weaknesses Enumeration (CWE). Firewalls should be implemented to deal with the discovered threats. The requests and responses from applications should be examined to check for those threats. The most critical web applications vulnerabilities are found to be injection and cross site scripting (Input validation) weaknesses.

Web application security miss-configuration and breaking –security in the web applications is an important issue in the SaaS environment. Not configuring the security properly or

applying weak security measures in applications is also a problem, more especially in a multi-tenant environment where each tenant can have their own security controls, which may not work well with others thus creating security holes. To be able to control and manage the security consistently, the cloud provider should be the one responsible for all the security controls [27].

2.4.3 Platform as a service (PaaS)

Platform as a service makes it possible for the applications which operate on the cloud to be deployed without incurring the cost of buying and maintenance of the underlying hardware or software layers. PaaS application security is made up of two layers namely the security of customer applications running on PaaS and the security of the PaaS system itself. In PaaS the service providers have the responsibility of securing the platform software stack [28].

SOA related security issues-because PaaS is built on the service-oriented Architecture model (SOA) all the security issues that occurs in the domain are inevitable. Those security threats are XML-related attacks, replay attacks, man-in-the-middle attacks, dictionary attacks and input validation attacks. In a PaaS layer, the security is the responsibility of both the cloud providers, service providers and consumers. It is important to secure the cloud services through authentication, authorization and WS-Security standards.

API Security-the Platform as a Service offers application programming interfaces that provide management functions such as business functions, security functions and application management. It is necessary to separate the APIs in memory to enforce consistent authentication and authorization on calls to APIs. This matter is the responsibility of the cloud service provider [27].

Underlying infrastructure security-in PaaS the responsibility of securing the underlying infrastructure lies with the providers. The developers do not have access to the underlying layers. The fact that user's data in PaaS is stored in cloud servers can be a security concern.

2.5 MODELS EMPLOYED TO ADDRESS SECURITY ISSUES

Lohr et al [1] and [7] discusses security in e-health cloud on the cloud platform. They propose the use of privacy domains for a patient's medical data as a technical measure to support the enforcement of privacy and data protection policies. The system must be able to divide execution environments for applications into distinct domains that are isolated from each other. Moreover, data leakages from the domain are prevented by the security architecture and the domain infrastructure. Lohr et al., [1] extended the concept of domains by focusing on the client platform security, there is a security kernel running on top of the hardware, thus

providing secluded virtual machines for application and conventional operating systems. Furthermore, there is a trusted virtual domain (TVD) proxy for each TVD, which manages the TVD on the client and configures the security kernel according to the TVD policy. Moreover, the architecture also has a secure graphical user interface (GUI) which provides input and output capabilities, thus making sure that a user can be able to identify the compartment they are interacting with. To verify the integrity of software on the client, there is a trusted hardware component. This hardware component is called the trusted platform module (TPM). It is usually implemented as a separate module which is integrated on the mainboard of a computer. TPM provides security and cryptographic functions such as public key encryption, digital signatures, and secure key storage. The trusted platform module (TPM) is usually implemented as a separate chip integrated on the mainboard of a computer. Authors in [**1**]and [**7**] used a systematic literature review to address specific questions like how privacy and confidentiality of data will be ensured in the cloud. Although their research is meant to solve the security issues in cloud computing, there are still some questions which are unanswered such as how data is protected on the user's side and when it is on transit. Authors in [**7**] mainly focuses on the security on the cloud thus neglecting security on the client platform and other channels which data take to reach the cloud. Although [**1**] came up with a better solution in terms of security and cryptographic functions, there are still issues like inefficiency, of which a client has to use a certain desktop with all those software included in it. If the client is not at the desktop, they can't use that system and another issue will be the cost implications of installing all software in client machines.

On the other hand, [**15**] also discussed various security issues in health cloud computing. The author explained that among all security issues, the main issue is of data encryption. Data needs to be encrypted whether it's on transit or stationary. Currently in the cloud, most of the data is unencrypted during processing, hence it poses a risk for attacks like man-in-the-middle and sniffer attacks. Authors in [**15**] proposed the use of homomorphic encryption on data, which allows data to be processed without unencrypting it. Although the author proposed this method, it is difficult to implement as the research is still at an early age in this area and many challenges have not yet been addressed. The author didn't come up with suggestions on how to address some of the challenges in this area. Authors in [**5**] proposed the ESPAC scheme (enabling security and patient –centric Access control) for e-health cloud computing. The ESPAC scheme is made for those who are requesting data and it allows data requesters to have different privileges according to their specific roles and assigns different attribute sets to them. At first the patient's health information is detected using body sensors and made ready to be transmitted to the trusted e-health care service provider. Furthermore,

to safely transmit the collected protected health information (PHI) to the e-health care service provider, a public key cryptography is used. The service provider is authorized to build-up the access tree. After receiving the Protected Health Information securely, the service provider categorizes the PHI according to a set of characteristics chosen by the patient. It then sets different privacy levels of those requesting data according to their roles and assigns different sets of attributes to these different levels. After classification, encrypted data is securely transferred to the cloud storage. The authors in [**5**] focus more on the safe transfer of data to the cloud but does not elaborate on what happens at the client's side. Onik et al., in [**29**] Proposed h-cloud architecture. This is whereby wireless sensor networks are integrated with cloud computing for storing patient's health data in the cloud. The proposed h-cloud has a security mechanism inside to guarantee data confidentiality, integrity, security and fine-grained access control. To achieve the fine-grained access control an Attribute-based encryption (ABE) technique to encrypt data before storing on the cloud is proposed. On the other hand, [**4**] proposed cipher text policy Attribute-Based Encryption (CP-ABE scheme). In this scheme, one public key for encryption is shared by healthcare providers and this is done to avoid public key infrastructure (PKI). The secret key of a healthcare provider can change the encrypted text to plain text only if the attribute set of the healthcare provider's key satisfies the access policy associated with cipher text. Kaur [**30**] proposed mechanisms at multiple levels and to provide a role-based control to ensure the protection of critical medical information about patients. To achieve this, they proposed to combine the use of symmetric cryptosystem for authentication together with the role-based access control (RBAC) mechanisms for authorization. Users of Community-Based In-home Services (CBIHS) are given a unique username and password. Rather than storing the user password in plaintext it is encrypted with a private key (pk) issued by a trusted third party (TTP). Once the users are authenticated according to the username and password they provided, the authorization is implemented based on user roles. Thilakanathan et al., [**31**], proposes a secure data sharing model and protocol that allows a group of users specified by the owner of data to share data. This proposed model improves the e-health monitoring system through the addition of a security layer which enables secure information sharing and efficiency. The original web service now represents the cloud data service (CDS) and added to it, is another web service called data sharing service (DSS) that handles the data sharing aspects of the system. The model works thus; each user in the group, including the data owner (DO) has a key database (DKDB). However, their keys are split into (n+1) part where n parts are saved in each proxy and the user keeps the remaining part. By doing so no user will know the complete key y which is needed to decrypt the keys in the data key database. When the user wants to have access to data, they call the DSS then decrypt the key in the DKDB using all the key pieces in

the proxy database corresponding to the calling user. This key is then used to decrypt the encrypted data in the cloud. Whenever the data owner wants the user to be prevented from accessing data, the key pieces are simply deleted hence they won't be any need for data to be encrypted again or for the keys to be distributed to the remaining users. In the secure data sharing model, the authors only talk about security on the cloud, there isn't anywhere the security at the user's side is mentioned. Ikuomola in [32] proposes an architecture of a secured e–health in cloud using homomorphic encryption and access control (SECHA). It consists of fine basic components (patient, PHR, access control module, user or subject and cloud). Homomorphic encryption allows patients to store encrypted PHR files in a public cloud and take advantage of the cloud provider's analytic services. The scheme prevents malicious insiders from accessing private information and prevents accidental leakage of such information.

Authors in [28] also carried systematic review on security vulnerabilities and threats in the cloud. The authors made a clear distinction between threats and vulnerabilities and how to address them. The review was based on security flaws at different levels of the cloud model. They explained that the first layer (software as a service) is the most vulnerable as it solely lies on the cloud platform hence security must be enforced. To mitigate these risks three solutions were suggested. Firstly, there is digital signatures which will secure data while it is on transit. Then secondly homomorphic encryption which allows processing and calculations to be done on data without being decrypted. Thirdly, the Trusted Virtual data centre (TVDC) which ensures isolation and integrity in the cloud environments. Although the authors suggested homomorphic encryption as the best encryption method for data in the cloud, they also highlighted that it requires a huge processing power which might affect the availability of data. Furthermore, [33] also reviewed different security and privacy issues and measures in place to solve them. Issues like confidentiality, trust, Integrity, authentication, encryption and recourse sharing issues were discussed. Confidentiality is preserved by designing a secure cloud storage service and using cryptographic techniques, authenticity is solved by using digital signatures, encryption by using several cryptographic methods instead of one and two-level encryption to manage encryption and decryption keys in the cloud.

2.6 MODELS EMPLOYED TO ADDRESS AVAILABILITY ISSUES

High availability is one of the important features in the cloud. Lack of availability has adverse effects on the cloud. Data could replicate among different data centres by two general methods. It can replicate using full data redundancy algorithms or a partial one. In full data redundancy algorithms, all data are expected to be replicated to one or more data centres based on the selected algorithm. To achieve high availability in the cloud Hernandez-Ramirez

et al [34] suggested the following redundancy techniques; no replication, mirroring, total replication and IDA based redundancy. In the no-replication method, only the original version of the file is stored on the disk pool. This method uses a round robin allocation policy which protects all the files which are stored in a single server thus providing minimal fault tolerance. In mirroring every time, a file is stored on a disk, DAM makes a copy of it to be stored in a different disk. This is the simplest method of ensuring higher availability with less resource consumption. In total replication all the file servers will have a copy of the file. Unlike the mirroring, this method has a high resource consumption, and it is the method which offers high availability. IDA based redundancy method uses an alternative approach which is based on information dispersal techniques. In this technique when a file is needed to be stored using IDA, the file is broken into pieces which are stored in different disks. This technique is used to provide better availability with less resource consumption [35].

Deshmukh et al. in [36] proposed system which ensures data storage security and availability using a distributed scheme. Servers are divided into two levels which are master and slave servers. Users who want to upload a file and modify it perform these operations on master servers and they have no connection with slaves which store chunks of data. Therefore, in case of failure or misbehaviour appearance of master servers, chunks of data will be recovered from slaves. Dispersing and recovery of data is performed based on the token keys which are generated through a token generation algorithm. This method supports both security and availability with the assumption that the administrators are responsible for authenticating users as well as chunking data. In fact, with chunking data and storing them in the separate slave servers, the errors are localized.

Authors in [37] proposed the Secured Virtual Diffused File System (SVDFS) which provide data security, integrity and availability which is suitable for deployment in a public cloud environment. A secured virtual diffused file system is a distributed file system which its main aim is to address the problem of confidentiality, integrity and availability. By allowing users to place a secured virtual file system on top of existing public cloud infrastructure. The main idea of this model is the separation between storage and information owners and it allows data storage outsourcing, but the control of these data is not outsourced. All these are achieved by using a file system interface and registry server that the interface could locate in the cloud or outside the cloud while the registry server is in the data owner site and keeps the metadata that are used for gathering or distributing the data.

2.7 ENCRYPTION

Encryption is the process of changing information in plaintext into an unreadable form in cipher text to protect it against unauthorized access. Usually an encryption algorithm is used to encrypt and decrypt data using either a public or private key. For someone to have access to such information they need to have a key.

2.7.1Evaluation parameters for encryption algorithms

2.7.1.1 Encryption time

Encryption time is the time an encryption algorithm takes to convert plaintext into cipher text. It is used to calculate the throughput of an encryption scheme. It is calculated as the total plaintext in bytes encrypted divided by the encryption time [1]. Different encryption algorithms have different encryption times. The algorithms in section 2.8 are the ones being evaluated to find the different encryption times. RSA being the one which takes the longest time to encrypt.

2.7.1.2 Decryption Time

Decryption time is the time an encryption algorithm takes to change cipher text back into plaintext. Decryption time is calculated as the total plaintext in bytes decrypted divided by the decryption time [2].

2.8 ENCRYPTION STANDARDS

2.8.1 Rivest Shamir Adelman (RSA)

This is one of the best-known algorithms and named after its inventors (Rivest, Shamir and Adelman). It is an asymmetric algorithm which uses a variable size block and key. It uses two keys, public key which is known by all for encryption and the private key which is only known by the end user for decryption. The sender encrypts the message using a public key and then the receiver will decrypt using the private key [**38**] [**39**].

2.8.2 Advanced Encryption Standard (AES)

This is a symmetric key block cipher that has been recommended by National Institute of Standard and Technology (NIST) as an upgrade of DES. It does its computations by using bytes rather than bits of Symmetric Encryption of the Original File. Its 128 bits of plaintext are treated as 16 bytes which is arranged into columns to be processed in the form of a matrix. It provides high security and faster speeds and it can be done on a wide range of platforms [**40**] [**41**].

2.8.3 Data Encryption Standard (DES)

This is a symmetric key-block cipher which uses a private key for both encryption and decryption. It provides a standard method for protecting confidential data. It accepts an input 0f 64-bit long plaintext and 56-bit key to produce an output of 64-bits ciphertext. It uses many permutations throughout in order to make it difficult for the cipher text to be analysed [40] [41].

2.9 CONCLUSION

From the review of the literature in this area, security and availability are burning issues in healthcare cloud computing. Many authors looked at different issues regarding security and availability in the cloud, they also came up with different algorithms to address this issue. But still they are not addressed fully, there are still some gaps on the research because the research is still at an early stage on this topic hence further research is needed to fully address this issue. Most of the proposed models addresses security on the cloud itself, the question arising might be "what about at the client's side and when the data is in transit?" This is an issue which also needs to be addressed. Chapter 3 presents algorithms and models to help solve security issues in healthcare cloud computing.

CHAPTER 3

SECURITY AND AVAILABILITY MODELS IN HEALTHCARE CLOUD

3.1 INTRODUCTION

Current models used in the healthcare cloud have proven not to be effective enough to address all the issues inherent with the healthcare cloud. This chapter addresses the research question; how can the problems in the current models used in healthcare cloud be addressed? First, the security requirements for healthcare cloud are presented and then followed by an evaluation of data security models used in cloud computing. Lastly, a new model is proposed to improve on the security of the current models.

3.2 SECURITY REQUIREMENTS FOR THE HEALTHCARE CLOUD

Security requirements are essential for any model that deals with storing patient's medical records. The requirements fall in the following categories, data confidentiality, on-demand revocation, write access control, scalability, efficiency and usability, patient consent and authorization for medical record sharing [**42**].

3.2.1 Data Confidentiality

To ensure confidentiality unauthorized users who do not have the access keys must not be allowed to decrypt the patient health record (PHR). A very strong access control mechanism should be employed [**43**]. And to avoid violating privacy rights information about the health records should be kept confidential [**1**].

3.2.2 On-demand Revocation

The system should not be able to allow a user whose attribute is invalid access. Hence the user can't use that attribute in future to access the patient health record files. A user's access rights can also be cancelled which is called user revocation [**43**].

3.2.3 Write Access Control

The system policies should be flexible so that changes can be applied to them especially in case of emergencies access to personal health records should be allowed. Unauthorized users shall not be allowed to have a write-access to the PHR's [**43**].

3.2.4 Scalability, Efficiency and Usability

The system should be able to accommodate many users, both from the personal and private domains. Users should have minimum responsibilities in the system to enjoy using it, and it should respond accordingly in order to deliver the services to the users [**44**].

3.2.4 Detection and Prevention of Security Attacks in the System

The system should be able to identify all the security attacks threatening it. Its main goal should be to protect the patient's highly sensitive data from disclosure and malicious violations [**45**].

3.3 DATA SECURITY MODELS

This section discusses different data security models proposed by different authors in this area to address the security issues in healthcare cloud computing. The models are based on the security requirements presented in section 3.3.

3.3.1 Electronic Health Record Security Reference Model

Electronic Health Records are used to store the patient's health records. Sharing of such records is facilitated by using cloud computing technology; hence there is a need to preserve their privacy and confidentiality. The electronic record security reference model provides a framework for managing such security issues in the healthcare cloud. This model is made up of three components and secure interactions which are integrated together to meet the security and privacy needs. The Electronic Health Record secure collection and the integration component which is responsible for collecting all the EHR data from various care delivery organizations (CDO's) and integrating it into a single storage area. Integrity, authenticity and confidentiality of the collected data are verified by the integrator. EHR secure storage and access management is responsible for ensuring secure storage and access control. It is divided into two parts, the secure storage server whose responsibility lies in storing the encrypted data and allowing only authorized users to access it, and the access control engine which oversees role-based or attribute-based access control policies, Health Insurance Portability and Accountability Act (HIPPA) compliance policies and also ensures that the access policies are adhered to. Lastly, there is EHR secure usage model whose role is to give access to the verified source content to consumers. For the signature and verification, the doctors who took part in the consultation will have to sign the medical certificate using a signature algorithm which is then sent to the patient, who must verify the authenticity of the consultation by looking at the medical certificate and the doctor's digital signatures [**46**]. Fig 3.1 shows the integration of components in the Electronic Health Record Security Reference model.

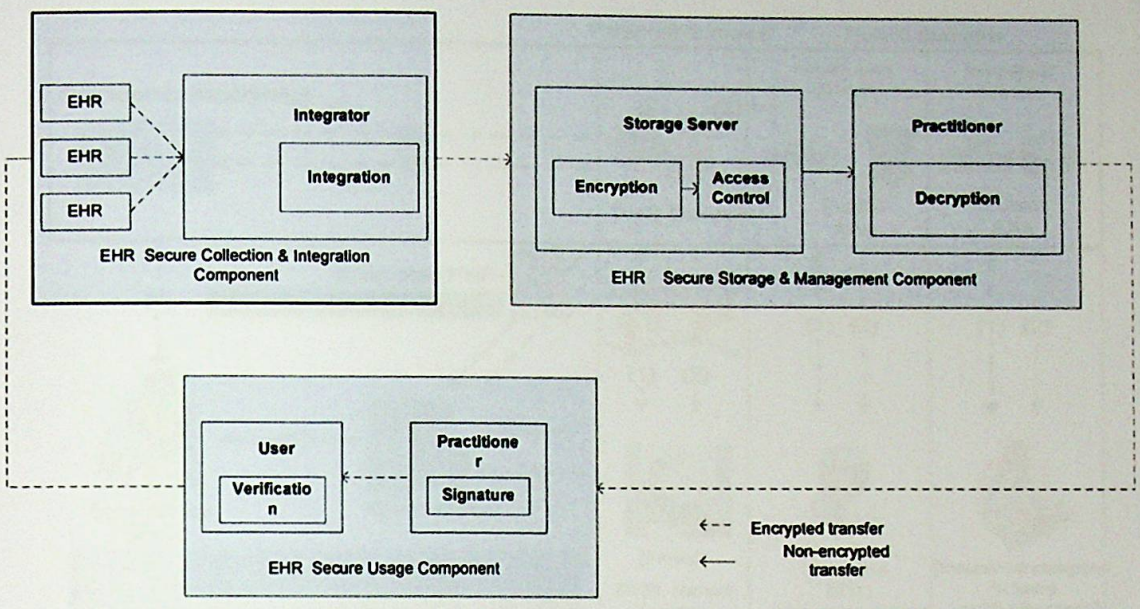


Fig 3.1: Electronic Health Record Security Reference Model system components [46]

3.3.2 Patient-centric, secure and scalable PHR sharing model

The advent of personal health records came with the benefits of sharing medical information across different servers. However, it came with privacy risks as information can be exposed to third parties and even unauthorized access to such information can occur. Encryption methods were introduced to protect the information from such risks before sending them, but still issues of scalability in key management, flexible access and efficient user revocation are a problem in achieving a fine-grained PHR model.

Li et al., proposed a patient-centric secure and scalable PHR sharing model. This model gives the patient control over who can access their personal health records. It focuses on multiple ownership of the PHR, whereby users are divided into many security domains according to their different access requirements. The security domains can be categorized as public where people can have access rights to data based on their professions e.g. doctors, nurses and medical researchers. And personal domains where access can be granted to people who are directly linked to the data owner like family and friends. This model is based on Attribute Based Encryption (ABE) encryption to provide security to the personal health information [43]. Fig. 3.2 shows the architecture of the proposed model.

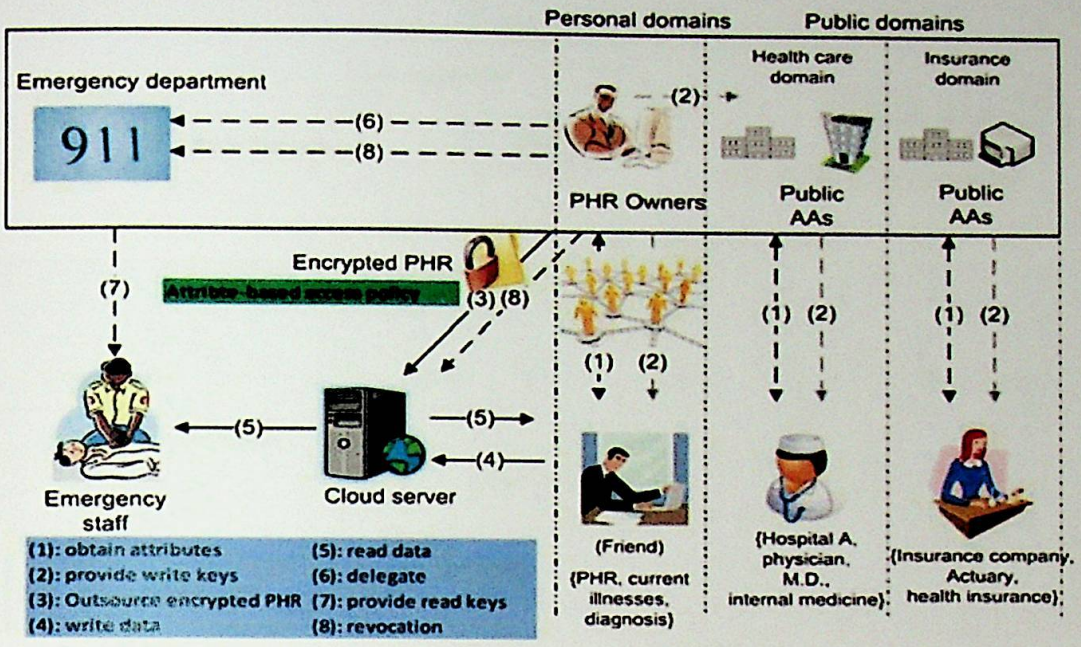


Fig. 3.2: Patient-centric, secure and scalable PHR sharing model [43]

3.3.3 A Novel Framework with Cloud-Based Privacy-Aware Role Based Access Control (CPRBAC) Model

Moving healthcare records to the cloud environment came with some risks such as loss of ownership of the user, privacy and confidentiality issues hence there is the need for a state-of-the-art-model to address such risks. Authors in [45] proposed a novel framework with CPRBAC model which uses less memory, resources, bandwidth and time but providing a firm security to the healthcare cloud. This model improves the existing CPRBAC model by adding some new components which are vital for improved data protection. This improvement is on authorisation delegation, resource sharing and context-based access control. The new components introduced by this model are; organisation (Or), conditions (Co), obligations (Ob) and purposes (Pu). The "Or" addresses the distributed role allocation, it also explains the R which defines a certain responsibility. Co defines a requirement which should be satisfied before an operation can be executed. Pu describes the purpose for performing the operation. Fig. 3.3 shows the components of the CPRBAC model.

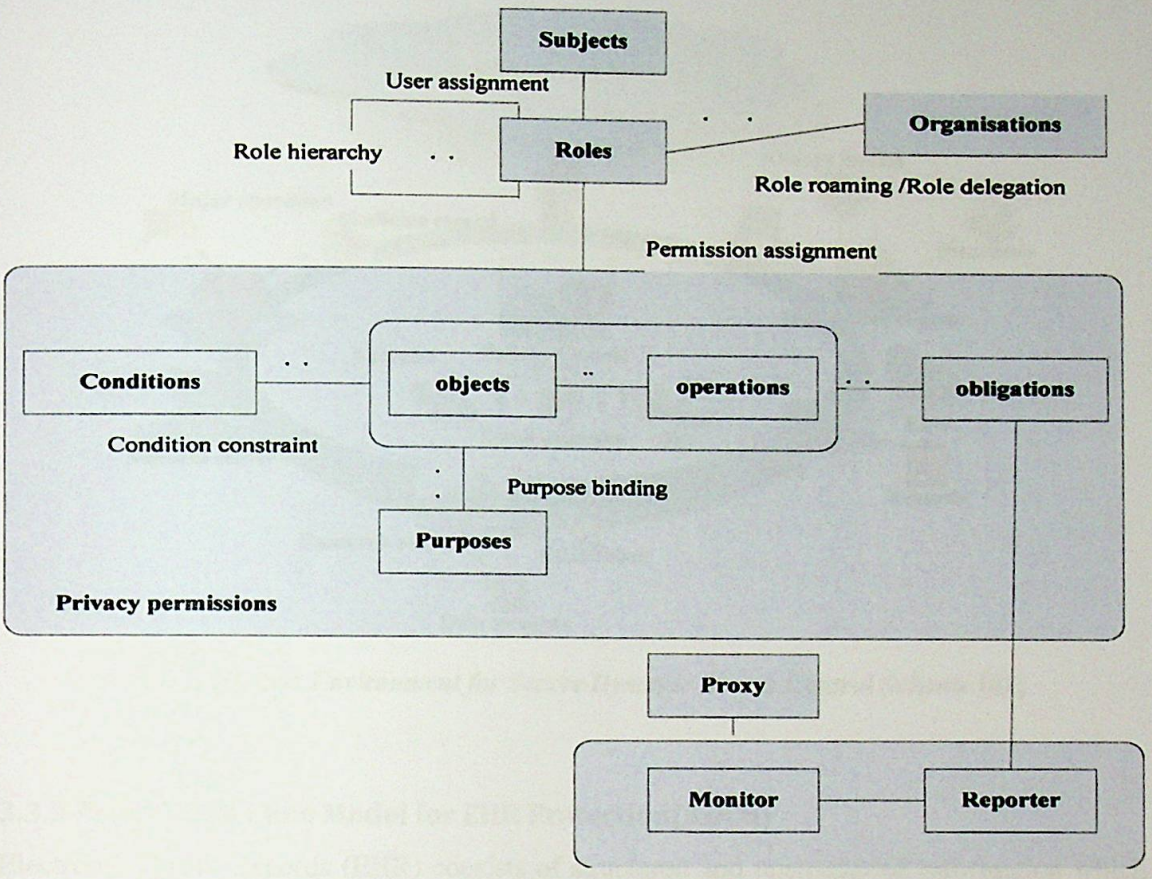


Fig. 3.3: Components of the CPRBAC model [45]

3.3.4 Secure Dynamic Access Control Scheme of PHR in Cloud Computing

The introduction of personal health records has brought many benefits to the medical sector involving convenient storage and flexible access in the cloud. Nevertheless, issues of privacy and confidentiality are inevitable as with every cloud platform, and data must be encrypted. Hence there is a need for an effective PHR scheme which will address such problems. Chen et al., [47] proposed a dynamic Access Control Scheme of PHR in Cloud Computing. This model offers users the ability to manage access and share personal health records in cloud computing environments. This scheme dynamically caters for multiple user access by providing legal authorities to access the PHR. The scheme provides users with a fair certification authority (CA) which assigns a user a super key for accessing the file. A super key verifies the identity of the user and ascertains that the user carries a legal key to ensure security, authenticity and reliability of the information transmitted. The scheme comes up with three phases which are initialization, key generation and derivation. Fig. 3.4 shows the access environment for the Secure Dynamic Access Control Scheme of PHR in Cloud Computing.

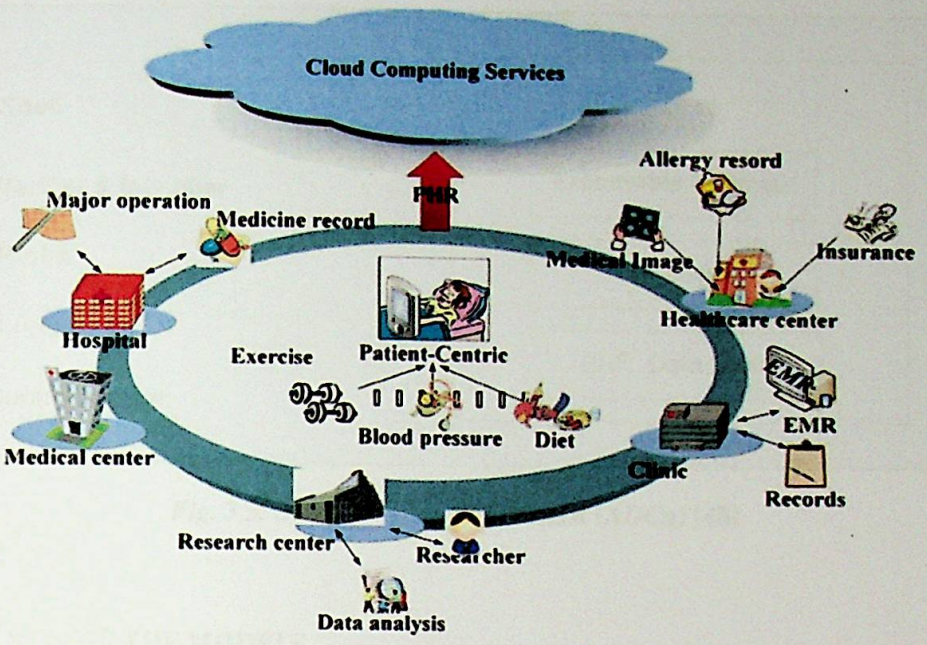


Fig. 3.4 Access Environment for Secure Dynamic Access Control Scheme [47]

3.3.5 Active Data Cube Model for EHR Protection(ADCu)

Electronic Health Records (EHR) consists of structured and unstructured information which follows no particular standard, thus resulting in interoperability and security issues when integrating them into the cloud environment. The EHR are stored in a multi-user-environment which makes it susceptible to security, privacy and data breaches often from malicious attackers on the cloud platform. The Active Data Cube Model (ADCu) is a solution which offers protection to the Electronic Health Records. The ADCu provides functions such as data encapsulation, data manipulation, security verification and networking. It encompasses a mechanism which monitors the dynamic flow of information hence any violations to data is reported to the users and recorded using the ADCu mechanisms. It also contains a log which records access activities to the data and distribute it to the various data owners. The model consists of two components; the shell and core. The shell is further made up of the scripts invoke verifier and identifier whose responsibility lies in verifying if the requests sent are legit and is able to verify the authenticity of requests because of the active taper-proofing codes embedded in it. The shell is executed before the core therefore after the whole verification and identification processes is complete the core can now start being active. The core is made up of the Executable Segment (ES), header and data blocks. The ES is responsible for making calls to the shell for activating the core, so that it can start the execution on data operations. The header specifies the basic information about data which is housed in the data blocks [48]. Fig. 3.5 shows the structure of an active EHR (ADCu).

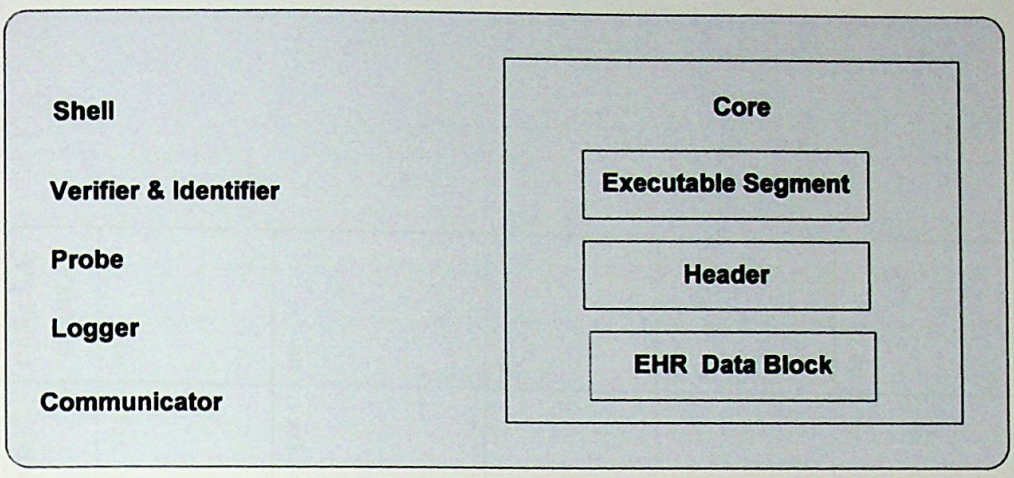


Fig. 3.5: Structure of an Active EHR (ADCu) [48]

3.4 ANALYSIS OF THE MODELS

This section evaluates the models proposed by different authors based on the requirements mentioned in this chapter which are security, integrity, privacy and availability. Lastly it provides a summary of the analysis in Table 3.1. From Table 3.1 we can conclude that all the models satisfy some of the requirements mentioned in section 3.3 except that they do not provide security at the client's side.

TABLE 3.1: ANALYSIS OF SECURITY MODELS

Model	Reference	Description	Cloud security	integrity	privacy	availability	Client-side security
Electronic Health Record Security Reference model	**[46] [49]**	An electronic Health Record Security Reference model which offers a secure storage, integration and access management.	yes	yes	yes	yes	no
Patient-centric, secure and scalable PHR sharing model	**[43] [50]**	A model that uses ABE techniques to encrypt PHR files and separate users into numerous security domains to reduce problems which comes with key management.	yes	yes	yes	yes	no
Cloud-based Privacy aware Role Based Access Control (CPRBAC) model	**[45] [51]**	An improvement of the already existing CPRBAC model with added components vital for improved data protection.	yes	yes	yes	yes	no
Secure Dynamic Access Control Scheme of PHR in Cloud Computing	**[47] [52]**	A PHR access control scheme with Lagrange interpolation polynomial to establish Patient Health Record access which is secure.	yes	yes	yes	yes	no
Active Data Cube Model for EHR Protection(ADCu)	**[48] [53]**	An active model for storing and securing EHR's in a cloud.	yes	yes	yes	yes	no

3.5 PROPOSED SECURITY ENHANCEMENT MODEL FOR HEALTHCARE CLOUD (SELF-ENCRYPTING SYSTEM)

From the models discussed in section 3.4, it was established that they focus mostly on the security of the data on the cloud rather than on the client side, which is a disadvantage since data can be accessed by unauthorized users on its way to the cloud. This research proposed a self–encrypting system which will improve on the security measures at the client's side proposed by other researchers, by encrypting data before it is sent to the cloud. The system will be linked to the cloud in such a way that before the client submits the data to the cloud, the data will go through that system for encryption. After the data has been encrypted that is when it will go to the cloud. This type of system will reduce the pressure of encrypting data on the clients and every client can be able to send their information on the cloud without worrying about the security. Fig. 3.6 shows a model of the proposed system (self-encrypting system).

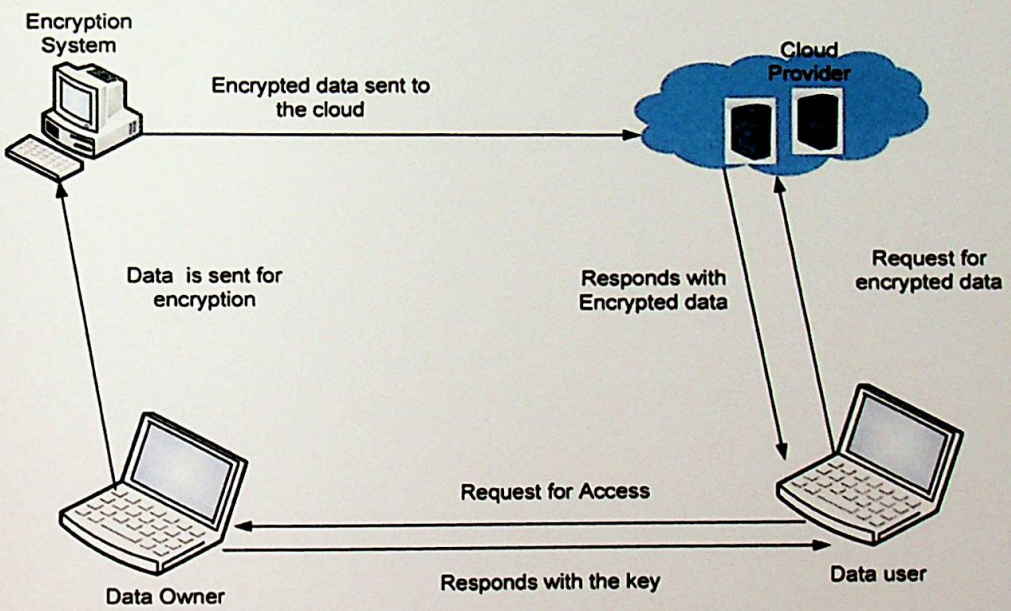


Fig. 3.6: System architecture of the proposed system

3.5.1 An Algorithm of the Proposed Model

The following steps are taken for successful operation of the proposed system:

STEP 1: User request a file from the healthcare

STEP 2: Data owner loads data into the encryption system.

STEP 3: The encryption system encrypts the data and saves it into the database.

STEP 4: The data owner sends encrypted data to the cloud servers. The cloud provider does not have any access to the data stored. Instead the data users are the ones who are granted access.

STEP 5: When the data user wants to access data from the cloud, they send a message to the data owner

STEP 6: The data owner will then check the user's profile to confirm if they are authorized to access that data.

STEP 7: If the user is authorized to access the data, the data owner will grant the user a random key with a time limit. Once the user accesses the data, the data owner will store the key in the key database and updates the cloud servers. The flowchart for this proposed model is shown in Fig. 3.7.

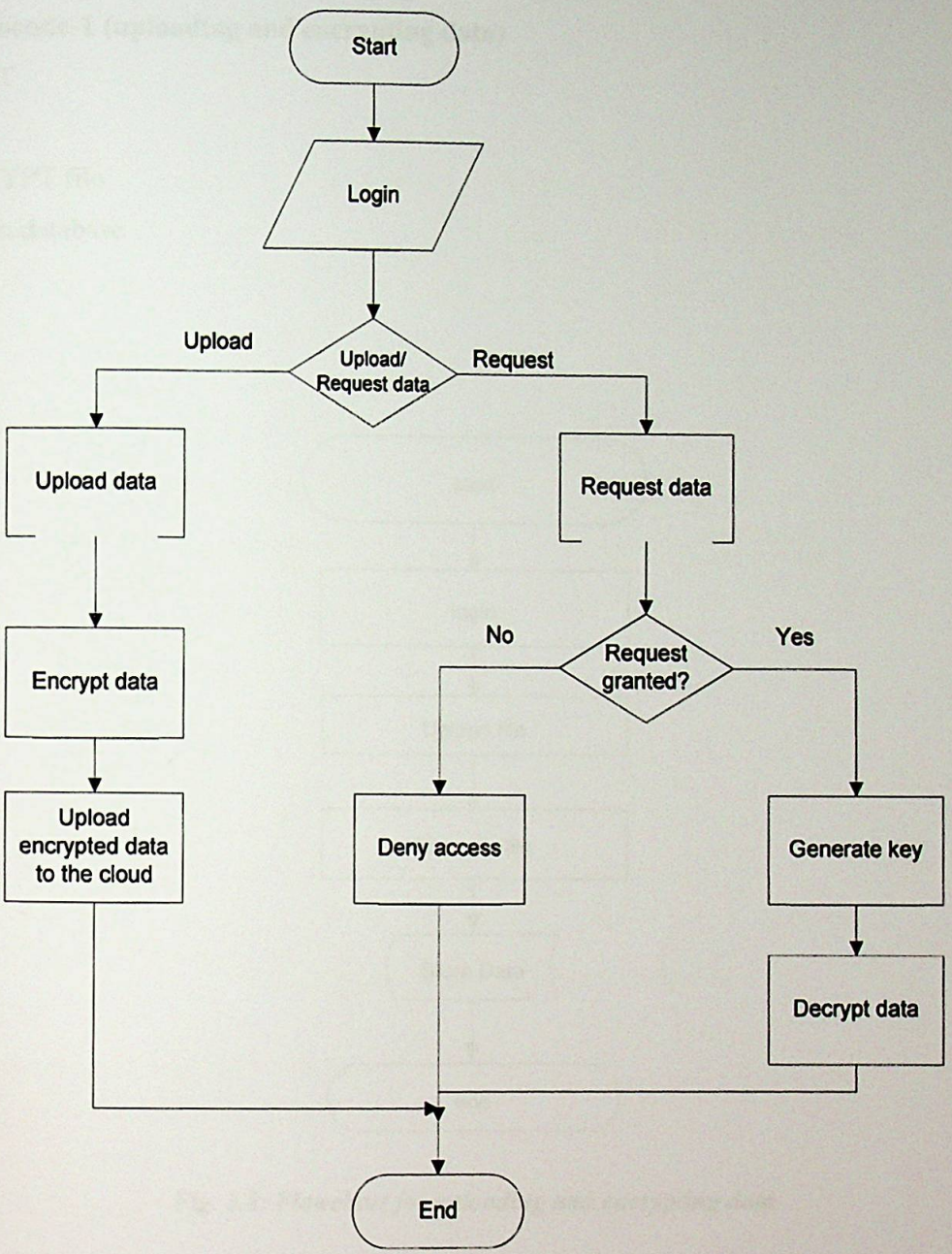


Fig. 3.7: Flowchart of the proposed security model

3.5.2 Algorithm for Uploading and Encrypting

The flowchart in Fig.3.8 shows the process of encrypting information to be secured in the cloud. The algorithmic descriptions as well as the pseudo-code are also shown for better understanding of the encryption part of the proposed system as:

1. Data owner logs into the encryption system
2. Data owner loads data into the system
3. The system encrypts data
4. Stores data into the database

Pseudocode-1 (uploading and encrypting data)

START

Login

ENCRYPT file

Store in database

EXIT

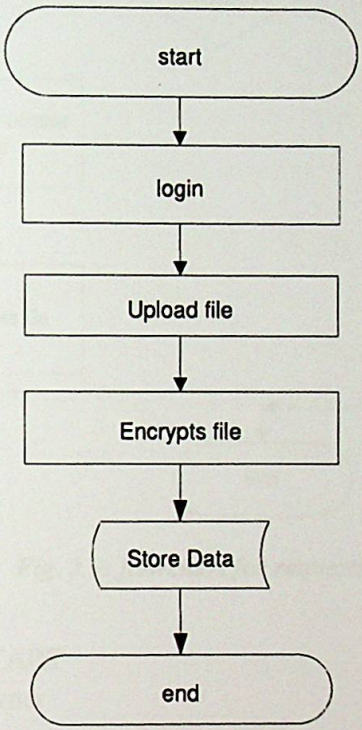


Fig. 3.8: Flowchart for uploading and encrypting data

3.5.3 Algorithm for Requesting Data

The flowchart in Fig. 3.9 shows the process of the data user getting access to the information. This involves access granting, random generation of keys and the decryption of the needed information. The algorithmic descriptions as well as the pseudo-code are also shown for better understanding of the data user part of the proposed system as:

1. Registers with the cloud
2. Requests data access from the data owner
3. Get the random key
4. Decrypts data

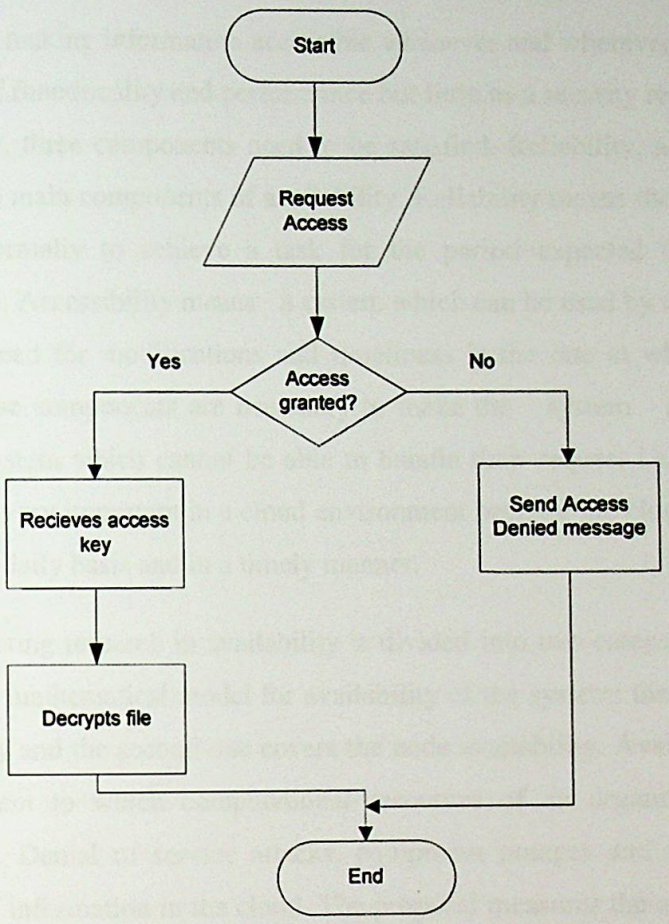


Fig. 3.9: flowchart for requesting data

Pseudocode-2 (Data User) START
SENDS a request to the data owner

IF request accepted

RECIEVES access key

DECRYPT data

ELSE access denied

EXIT

3.6 AVAILABILITY IN CLOUD COMPUTING

Availability deals with making information accessible whenever and wherever it is needed. It is explained in terms of functionality and performance but little as a security requirement [54]. To achieve availability, three components need to be satisfied. Reliability, accessibility and timeliness are the three main components of availability. Reliability means the likelihood of a system functioning normally to achieve a task for the period expected under operating conditions encountered. Accessibility means a system which can be used by many users with no complications or need for modifications and timeliness is the rate at which the system responds [55]. All these components are necessary to make the system available. Users cannot depend on a system which cannot be able to handle their request i.e. an unavailable system. Availability is very important in a cloud environment because the cloud stores tons of information needed on daily basis and in a timely manner.

In this section, the existing research in availability is divided into two categories, which also serves as the proposed mathematical model for availability of the system; the first one covers information availability and the second one covers the node availability. Availability in cloud computing is the extent to which computational resources of an organization are fully accessible and usable. Denial of service attacks, equipment outages and natural disasters threaten availability of information in the cloud. The proposal measures the availability based on all factors that interfere with the system availability. This measurement can lead to an optimum decision taking in terms of the system's availability. Availability depends on the security or performance of a system and also on the applications that runs on the system. Some applications focus on the performance more than security and their policy for redundancy and load balancing will define how this goal will be achieved. While in other applications, security is more important, they use different techniques for keeping data confidential. These decisions and techniques impact directly to the availability, and application providers should map proposed a mathematical method to their applications.

3.6.1 Information Availability in Cloud Computing

For having available information, redundant instances are expected to be stored in other nodes. The redundancy is due to replication of data. Cloud scenarios face a huge amount of data which is known as big data, as such, the replication method affect the storage and communication cost directly. Therefore, a method in which the size of the replica is reduced has high priority. In a replication method, there is great concern about resource consumption as well as availability. In fact, a trade-off between cloud resource consumption and availability seems necessary.

Data could replicate among different data centres by two general methods. It can replicate using full data redundancy algorithms or a partial one. In full data redundancy algorithms, all data are expected to be replicated to one or more data centres based on the selected algorithm. That means it consumes more resources in terms of communication and storage, but it provides high availability. The partial data redundancy algorithms split data into multiple parts and store each part in different data centres. Therefore, the communication and storage cost would be decreased regarding the size of chunked data. In the case of failure, the portions of data are gathered from all or number of data centres. Further information about the redundancy methods can found in [**34**].

Authors in [**56**] proposed a model in which servers are divided into two levels, which are master and slave servers. Users who want to upload a file and modify it perform these operations on master servers and they do not have any connection with slaves which store chunks of data. Therefore, in case of failure or misbehaviour appearance of master servers, chunks of data will be recovered from slaves. Dispersing and recovery of data is performed based on the token keys which are generated through a token generation algorithm. This method supports both security and availability with the assumption that the administrators authenticate users as well as chunking data. In fact, with chunking data and storing them in the separate slave servers, the errors are localized. Authors in [**57**] discussed the Secured Virtual Diffused File System (SVDFS) which provide data security, integrity and availability suited for deployment in a public cloud environment. The main idea of this model is the separation between storage and information of owners, it allows data storage outsourcing, but the control of data is not outsourced. All these is achieved by using a file system interface and registry server that the interface could locate in the cloud or outside the cloud while the registry server is in the data owner site and keeps the metadata that are used for gathering or distributing the data. More data redundancy methods are found in [**55**] [**58**] [**59**].

3.7 Node Availability

Node availability is computed by a fraction of the time that the node is under operation. A node in this study refers to the data centres which include both physical servers and virtual machines. Parameters used in measuring the node availability are Mean Time to Failure (MTTF) or mean up time, Mean Time to Recover (MTTR) or mean downtime and Mean Time between the Failure (MTBF). MTTF refers to the expected time of operation before first failure occurrences. MTTR is the time taken by each failed machine to be recovered while the MTBF indicates the time between two consecutive failures. There are some factors that are affected by these mentioned parameters. In [**59**] Mean Failure of Cost (MFC) base on the

mentioned terms of time was calculated. The authors worked on availability and security and then calculated the effect of failure with the financial cost that this faultier impose to the organizations or cloud providers. Daniel investigated the availability of Internet Data Centres (IDC) in [**58**] and made it bearable by comparison with performance. IDC has *M* equivalent machines that respond to user requests and applications which are running in IDC are expected available 99.999 percent of the time. In this research, the availability is computed using the Equation 3.1 as:

$$A_j = \Sigma_{k=j}^{M} P_j \tag{3.1}$$

Where, *j* indicates, at least, *j* machines in operation and the probability P_j is computed using Markov chain. P_j is probability that at least *j* machines are in operation. To explain the performance and availability of machines in the system, a model as given in [**58**] is adopted and shown in Fig. 3.10.

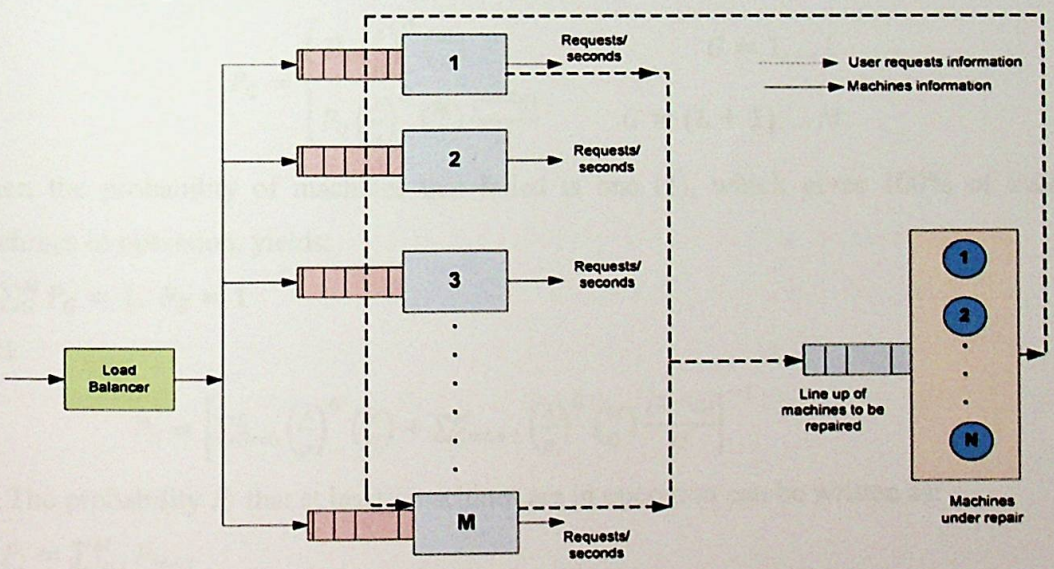


Fig. 3.10: Performance and availability of machines in the system

For better understanding of availability determination of the system, the following are of interest for the data owner:

1. Rate of failure of the machines, the number of the system machines, the number of personnel available to take up the repair of the failed machine, the average time it takes the personnel to revive a failed machine and the probability that exactly *j* machines are in operation and active at any given time;

2. Rate of failure of the machines, the number of the system machines, the number of personnel available to take up the repair of the failed machine, the average time it takes the personnel to revive a failed machine and the number of personnel that will guarantee at least j machines are operational for a given probability;
3. The effect of the average time taken by the repair personnel to fix a machine on the total MTTR;
4. Effect of the size of personnel involved in the repair of the failed machine, considering Mean Time to Repair (MTTR);

It is then worthy of noting:

1. The probability that exactly j machines are operational for a given number of machines in operation for different number of available personnel to repair the failed machines;
2. The probability that at least j machines are in operation for a given number of machines in operation for different number of available personnel to repair the failed machines given the probability P_G that G number of machines has failed is represented

$$P_G = \begin{cases} P_0 \left(\frac{\lambda}{\mu}\right)^G \binom{M}{G} & G = 1, \dots L \\ P_0 \left(\frac{\lambda}{\mu}\right)^G \binom{M}{G} \frac{L^{L-G} G!}{L!} & G = (L+1), \dots M \end{cases} \tag{3.2}$$

When the probability of machines that failed is one (1), which gives 100% of available machines in operation, yields:

$$\sum_G^M P_G = 1, \; P_G = 1 \tag{3.3}$$

And:

$$P_0 = \left[\sum_{G=0}^{L} \left(\frac{\lambda}{\mu}\right)^G \binom{M}{G} + \sum_{G=L+1}^{M} \left(\frac{\lambda}{\mu}\right)^G \binom{M}{G} \frac{L^{L-G} G!}{L!}\right]^{-1} \tag{3.4}$$

The probability P_j that at least j machines are in operation can be written as:

$$P_j = \sum_{i=j}^{M} P_{M-i} \tag{3.5}$$

Where, P_{M-i} in (3.5) is the relationship gotten from Equations (3.2) to (3.4).

The sets of physical and virtual machines are defined as:

$$P = \{P_1, P_2, P_3, \dots., P_M\} \tag{3.6}$$

$$VM = \{VM_1, VM_2, VM_3, \dots. VM_N\}$$

P is a set of the physical machines and VM is a set of virtual machines for each physical machine.

Assuming that every VM can play a backup role for the failed one, then the node availability is achieved by the following equation:

$$\text{Availability} = \sum_{j=K}^{M} \sum_{i=L}^{N} p(P_j) p(VM_i) \quad (3.7)$$

The first summation calculates the probability of availability of M physical machines which are under operation and the second one does the same for N virtual machines which are under operation on each physical machine.

The probability of availability of physical or virtual machine is calculated by:

$$p^{(pj)} = 1 - p^{(fpj)} \quad (3.8)$$

Where $p\left(FP_j\right)$ depends on the following:

1. Rate of failure of j (α)
2. Time that j needs to be recovered (MTTR)

The second factor, Mean Time to Recover (MTTR) is also influenced by the following parameters: R, shows the number of machines which are waiting in the repair queue, S, represents the number of staff who are repairing the machines and Tr indicates the average of time that each machine needs to be recovered and back to the system. Then MTTR is computed as follows;

$$MTTR = \begin{cases} Tr; & IF\ S > R \\ (R - S)Tr; & IF\ S < R \end{cases} \quad (3.9)$$

To make the calculation more accurate, the assumption made need to be modified based on the reality. Therefore, the exact number of VMs that host the redundant data is expected to be determined. To this end, graph (G_{b_node}) is necessary to be defined. The $G_{b\text{-}node}$ represents which node could be a backup node, for a particular one, the directed and weighted graph is expressed as:

$$G_{b\_node} = (P_x VM_y, E, W) \quad (3.10)$$

Where, $P_x VM_y$ represents Y^{th} VM that belongs to the X^{th} physical machine as the node of the graph. While E shows the edge of the graph that gives to the system an input parameter and it is a function of redundancy algorithm which is used for the data availability. Finally, W determines the weight of each edge. Fig. 3.11 illustrates the G_{b_node} graph which has 3 nodes.

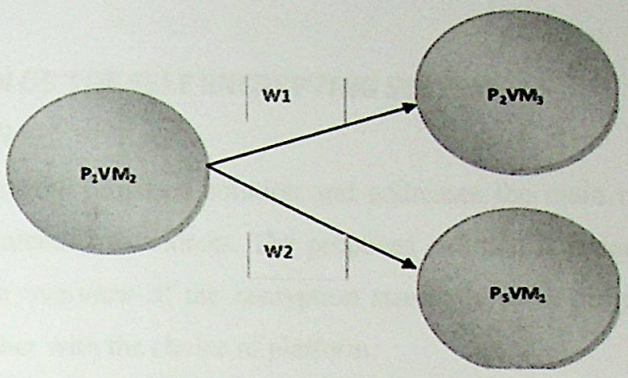


Fig. 3.11: The G_{b_node} graph

Fig. 3.11 shows third virtual machine by second physical machine and first virtual machine by fifth physical machine have redundant data of the second virtual machine of the first physical one. If one of them is available at the failure time that one will be replaced. But if both are available, a decision would be made based on the W_1 and W_2.

W is a dynamic parameter which is calculated based on three factors; which are the capacity of CPU, available memory of the second node and the available bandwidth. Therefore, in the case of failure, the requests transfer to the node which has a higher W.

Data redundancy algorithm has a direct impact on the availability of system because it defines the number of edges of the graph and it adds another weight to the graph that shows the number of virtual machines needed for recovery the data of the system. Then by defining it, the graph would change to the 2-weighted and directed graphs in which decision making will be done based on the combination of both weights.

3.8 CONCLUSION

This chapter described the overall research methodology used to answer the research questions presented in Chapter 1. It started by describing the security requirements which every security model should address. An evaluation of the security models in healthcare cloud was done. The models were evaluated looking at security, privacy availability, integrity and security on the client's side. The models for availability were also discussed. Lastly, the chapter presents architecture for the proposed self-encrypt system together with the flowcharts diagrams which explain the steps in developing the system. The mathematical model for availability is also explained. The implementation of the system is presented in Chapter 4.

CHAPTER 4

IMPLEMENTATION OF THE SELF ENCRYPTING SYSTEM

4.1 INTRODUCTION

This chapter describes the proposed solution and addresses the main research question for which this study is intended to address. The proposed solution is based on data encryption standards. It gives an overview of the encryption standards used, design, development and implementation together with the choice of platform.

4.2 DEVELOPMENT TOOLS AND ENVIRONMENT

The implementation tools and environment were prepared for the client side. Java language was used for implementation and was based on NetBeans IDE version 8.2. The specifications for the client machines are as follows;

1. Hp laptop
2. Ram-8.00 GB
3. Processor-intel® Core™ 17-7500U CPU
4. Operating system 64-bit operating system

4.3 THE SELF-ENCRYPTING SYSTEM

The system works by encrypting a plaintext file which is the patient's health record at the client's side. The encryption standard used is a symmetric encryption standard whereby one key is used for both encrypting and decrypting. In this case the symmetric method which was chosen was the Triple Data Encryption Standard (3DES). The client who is either a patient or a doctor uploads a file into the system which encrypts the file and sends it to the cloud. The second part of the system allows a user to request for a key to decrypt the file. If the user is authorised to access the file, a secret key is sent to the user through RSA algorithm. This is done to protect unauthorised users from getting hold of the secret key.

4.3.1 Encryption of the Original File

Encryption is a very important aspect of this system as data needs to be protected before it can be transmitted though the network to the cloud. The original file is a patient health record which contains plain text and needs to be encrypted before being sent to the cloud because it contains sensitive data. In this implementation 3DES is used to encrypt the file. A random key is generated to prevent hackers from guessing the keys. Each time a file is encrypted, a different key is produced. During encryption process, the original patient health record is converted to cipher text which can now be sent to the cloud and be accessed using the

corresponding key. Code 4.1 shows the code for encrypting a file. A graphical user interface (GUI) was created to allow the user access to the system.

CODE 4.1: CODE FOR ENCRYPTING A FILE

```
public static void encryptOrDecrypt(String key, int mode, InputStream is, OutputStream os) throws
Exception {

 DESEDEKeySpec dks = new DESEDEKeySpec(key.getBytes());

 SecretKeyFactory skf = SecretKeyFactory.getInstance("DESEDE");

 SecretKey desKey = skf.generateSecret(dks);

 Cipher cipher = Cipher.getInstance("DESEDE");

 if (mode == Cipher.ENCRYPT_MODE) {
        cipher.init(Cipher.ENCRYPT_MODE,
desKey);

 CipherInputStream cis = new CipherInputStream(is, cipher);
makeFile(cis, os);
```

4.3.2 Decryption of the File

Access to the encrypted file will be through a secret key used to encrypt the file, and which is generated randomly. The user who can be a doctor, physician, insurance consultant will have to request for a secret key from the data owner. If the request is accepted, the user will be sent a secret key. The secret key is transported to the other party using Asymmetric cryptography (RSA). This is done to solve the key distribution issue which is common amongst symmetric algorithms. Otherwise if the user is not authorized to access the file a denial of access message will be sent to the user. This is to ensure that the privacy and confidentiality of data is enforced. Code 4.2 shows a code for decrypting data.

CODE 4.2: CODE FOR DECRYPTING DATA

```
} else if (mode == Cipher.DECRYPT_MODE) {

        cipher.init(Cipher.DECRYPT_MODE, desKey);

        CipherOutputStream cos = new CipherOutputStream(os, cipher);
        makeFile(is, cos);
```

4.3.3 How the System Addresses Privacy and Confidentiality Issues

The healthcare holds highly sensitive information, hence the need to protect it from malicious access which can jeopardize the privacy and confidentiality of information. The proposed system was built to solve some security issues in healthcare cloud more especially the privacy and confidentiality. This system ensures that information is not accessed by unauthorized users by random generated keys. These keys are generated so that hackers cannot be able to guess the keys and gain access to the files. Every time a user request for a key, a different key is generated. This method protects the privacy and confidentiality of information. Code 4.3 shows the code for generating random keys.

CODE 4.3: CODE SNIPPET FOR GENERATING RANDOM KEYS

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
String contain="1ab2cd3ef4gh5ij6klm7nop8qrst";
    Random rndm =new Random();
    StringBuilder sb=new StringBuilder(7);
for (int i=0; i<7; i++)
    {
       sb.append (contain.charAt(rndm.nextInt(contain.length())));
}
    String Random=sb.toString();
    jTextArea2.setText(Random);
```

Table 4.1 summarizes the main methods used for Triple DES encryption and decryption.

TABLE 4.1: CODE DESCRIPTION FOR ENCRYPTION AND DECRYPTION

Code	Description
final byte[] digestOfPassword = md.digest("HG58YZ3CR9")	A function that takes in the plaintext and key and then returns a cipherText
DESKeySpec dks = new DESKeySpec(key.getBytes());	Returns An encryption **key** value implementing the DES encryption algorithm
Final SecretKey key = new SecretKeySpec(keyBytes, "DESede");	Returns the algorithm name of the SecretKeySpec object
SecretKey desKey = skf.generateSecret(dks);	Generates a secret key
JFileChooser fc=new JFileChooser();	Chooses a file to display.
String path=fc.getSelectedFile().getAbsolutePath ();	Gets the path of the selected file
FileInputStream fis=new FileInputStream(f);	Reads bytes from a file.
FileOutputStream fos=new FileOutputStream("encrypted.txt");	Creates a file to display encrypted data.
Random rndm =new Random();	Generates random numbers
StringBuilder sb=new StringBuilder(4);	Creates a string of 4 characters
sb.append (contain.charAt(rndm.nextInt(contain.leng th())));	appends the value of the parameter to the value of the current String Builder object
jTextArea2.setText(Random);	Displays the generated string into a text area.
FileReader reader= new FileReader("C:\\Users\\GEEMOLAM S\\Documents\\NetBeansProjects\\logi n\\src\\login\\healthcard.txt");	Reads the encrypted file.

JOptionPane.showMessageDialog(null ,"Decryption key not Found");	Displays an error message when trying to access a file without a key.
final MessageDigest md = MessageDigest.getInstance("md5");	Performs a final update on the digest using the specified array of bytes

4.4 CONCLUSION

The chapter described the implementation of the self-encrypt system. This system encrypt data at the client's side before it can be sent to the cloud thus reducing the risk of sending data in plaintext to the cloud which can affect the confidentiality and privacy of data. The tools and the programming environment used to develop the system were explained, and the sample codes for encryption and decryption were given, and lastly a description of the methods used. The results of this implementation are found in Chapter 5.

CHAPTER 5

RESULTS AND ANALYSIS

5.1 INTRODUCTION

This chapter presents the results from the proposed availability model and the self-encrypt system which were discussed in the previous chapters (chapter 3 and 4). The chapter is organized as follows; section 5.2 presents the results for the availability model, 5.3 the results of the self-encrypt system and 5.4 concludes the chapter.

5.2 RESULTS FROM AVAILABILITY MODEL

For this research model, we consider number of nodes in operation (j) =60 in the data centre for the different numbers of personnel available for the repairs (L) of failed machines as: 10, 5, 3 and 2. This is also with the assumption that MTTF is 250 min, λ represents the number of failures per minute. (λ = 1/250 =0.004 failures per minute). The average time taken for personnel to diagnose and repair a machine (MTTR) is assumed to be equal to 20 min, μ represents the number of repairs per minute (μ = 1/20 = 0.05 repairs per minute).

Fig. 5.1 shows the probability that at least j machines are operational with respect to j machines in the data centre for the different number of personnel available to service the failed machines. In the Fig. 5.1, it should be observed that, for 2 personnel ready to service failed machines, L = 2, the probability that at least 20 machines are operational is 0.94 and the probability of at least 30 machines being operational is 0.18. From Fig. 5.1, it can be observed that, for the operation of 40 machines the probability of them being fully operational is zero. If we also look at having 3 personnel available to put the failed machines or nodes in action (L = 3), we can have up to 30 machines fully operational. This means that, if we need 40 machines to be operational, with 2 or 3 personnel, it will not be possible. To have 40 machines fully operational, we will need 5 or 10 personnel to be on ground for service. However, to have 98% of the machines fully operational, 10 personnel are required as shown in Fig. 5.1.

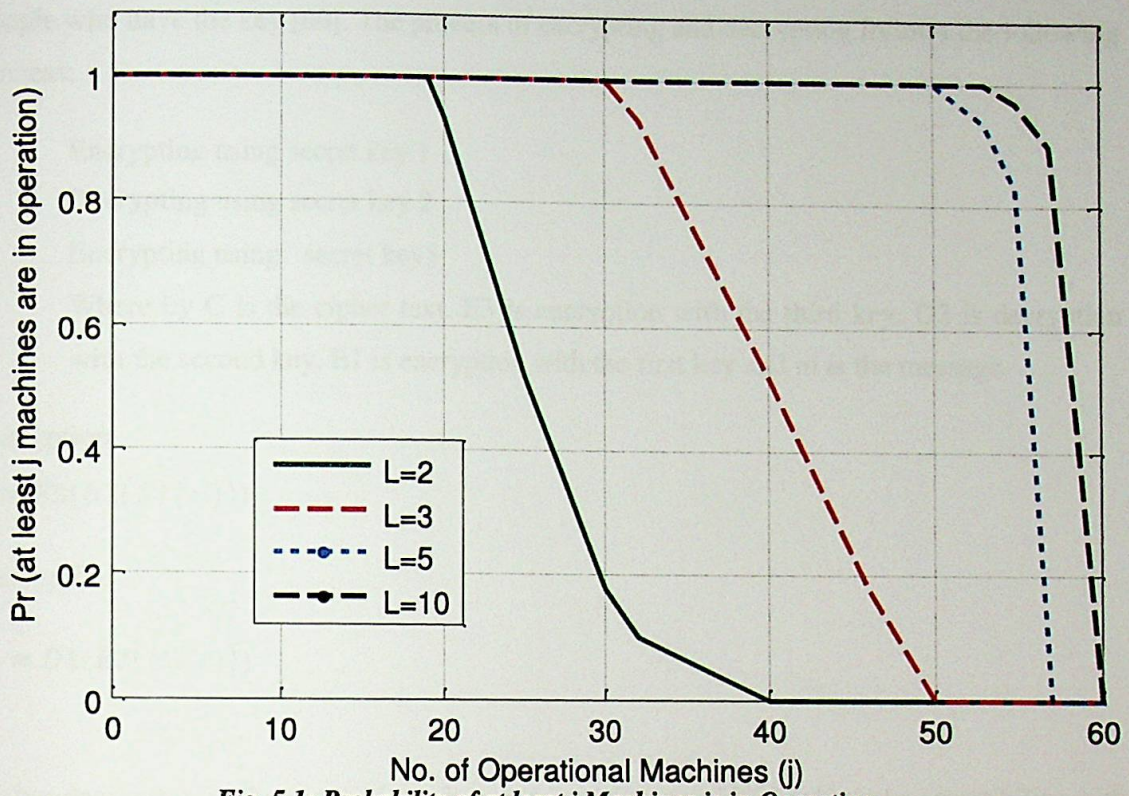


Fig. 5.1: Probability of at least j Machines is in Operation

This model can be applied in the self-encrypt system to make optimum decisions regarding system availability. From the model, it can be observed that the number of personnel available to repair machines impact on the number of machines to be repaired. The more the personnel, the more the number of machines which can be repaired in less time, thus making the system available. Furthermore, the time in which the system can be restored for service is very important, therefore this model allows for calculations on the time taken to repair the system.

5.3 RESULTS FROM THE PROPOSED SELF -ENCRYPTION SYSTEM

This section presents the results from the proposed system (self-encrypt system) which is meant to address the problems discovered in other security models. The section shows the original patient health care record being encrypted though the use of Triple data encryption standard (3DES) using 168-bits key length consisting of three sub-keys and each key is 56bits.

5.3.1 Operation of Triple DES (3DES)

Triple DES is based on DES standard. It improves the key-length from 56 bits to 168 bits as a countermeasure to overcome the weakness of DES which was the key-length which can be easily guessed. Triple DES converts data into cipher text in three stages and use a different key for one of the three stages so that converting data back into plaintext can only be done by

people who have the key [**60**]. The process of encrypting and decrypting follows the following process;

- Encrypting using secret key 1
- Decrypting using secret key 2
- Encrypting using secret key3

  Where by C is the cipher text, E3 is encryption with the third key, D2 is decryption with the second key, E1 is encryption with the first key and m is the message.

Encryption

$c = E3(D2(E1(m)))$

Decryption

$m = D1(E2(D3(c)))$

Using decryption in the second step during encryption provides backward compatibility with common DES algorithm. In this case, first and second secret keys or second and third secret keys are the same whichever key.

- $c = E3\left(D1(E1(m))\right) = E3(m)$
- $c = E3\left(D3(E1(m))\right) = E1(m$

  Fig. 5.2 shows the operation of 3DES.

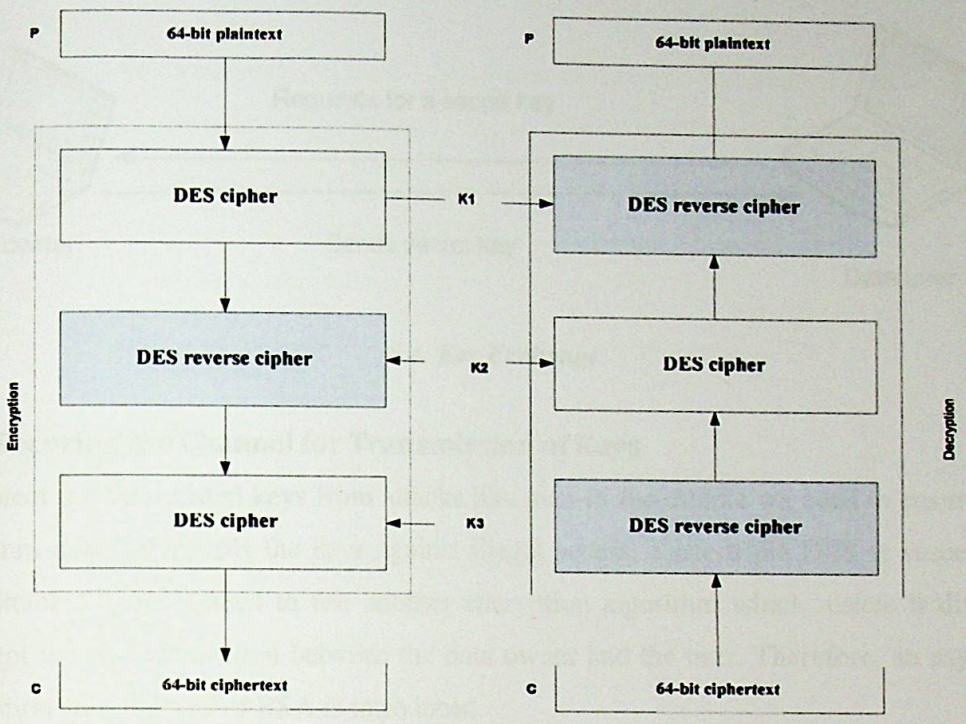


Fig. 5.2: Operation of Triple DES [60]

5.3.2 Random Key Generation

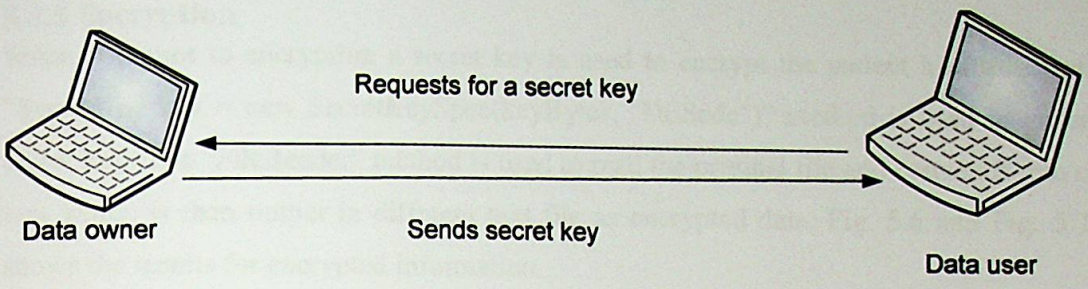
To encrypt and decrypt a patient health file, a secret key is generated randomly. The "Random rndm =new Random ();" method was used to generate a new key each time a user click on the generate key button. A 56-bit secret key consisting of eight characters was generated. The key was generated during the encryption stage. Fig. 5.3 shows possible keys generated.

5at881b jbl86e7 nqco11c rbrtprc sf5endr kt34mr6 fciar2b 3eq8dkj ln7k2gf 1b80jdf cofb2md sall5lo d6impbb egaon13 ghoasir 125rcio 4j875gt toprgrn rmfgoq2 6gffdbo jngppf1 mgkjeej 3mijc6t ej7l14 cohj6gm cjhnlm3 hm1jg6b 4t5mopo b421tpd jhosa3q bn5mb3a a51jzb3

Fig. 5.3: Random Generated Keys

5.3.3 Key Distribution

When the data user wants to access the data stored in the cloud, they need to request for a secret key from the data owner instead of the cloud which is susceptible to attacks. The data owner will check if the user is authorized and send the key though a secure channel. The user will in turn use the key to decrypt the file. Fig. 5.4 shows key exchange in a self-encrypt system.



5.4: Key Exchange

5.3.4 Securing the Channel for Transmission of Keys

To protect the transmitted keys from attacks like man-in-the-middle we need to ensure that an algorithm used safeguards the keys against illegal access. Since triple DES is susceptible to such attacks, there is need to use another encryption algorithm which makes it difficult to intercept the communication between the data owner and the user. Therefore, an asymmetric encryption standard called RSA is introduced.

In RSA a pair of keys are generated, which are the public key (k_{pub}) and the private key (K_{pr}). The public key is known by everyone while the private key is kept a secret. The public key is used for encryption and the private key is for decryption hence only the person with the correct key able to decipher the message. During encryption, a message(X) is encrypted with the user's private key(K_{pr}) and the user will decrypt the message using their private key ($d_{K_{pr}}$). By using this algorithm, the secret key for 3DES is encrypted by the data owner using the public key of the data user. The data user then recovers the secret key by decrypting the cipher text sent by the data owner. Fig. 5.5 shows the key distribution using RSA.

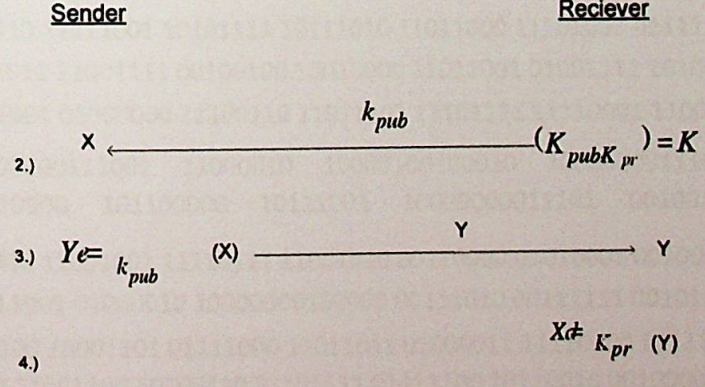


Fig. 5.5: Key Distribution in RSA [62]

5.3.5 Encryption

When it comes to encryption a secret key is used to encrypt the patient health record. The "SecretKey key = new SecretKeySpec(keyBytes, "DESede")" method is used to generate a secret key. The "FileReader" method is used to read the original file and convert it into cipher text which is then output in different text file as encrypted data. Fig. 5.6 and Fig. 5.7 [63] shows the results for encrypted information.

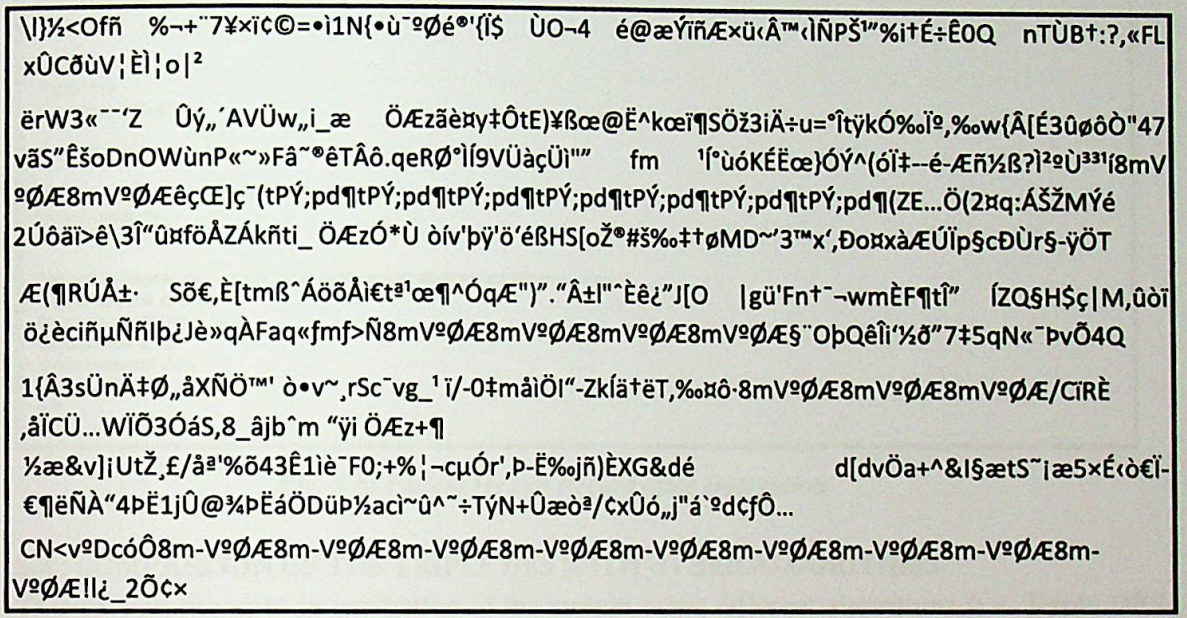


Fig. 5.6 Encryption Results

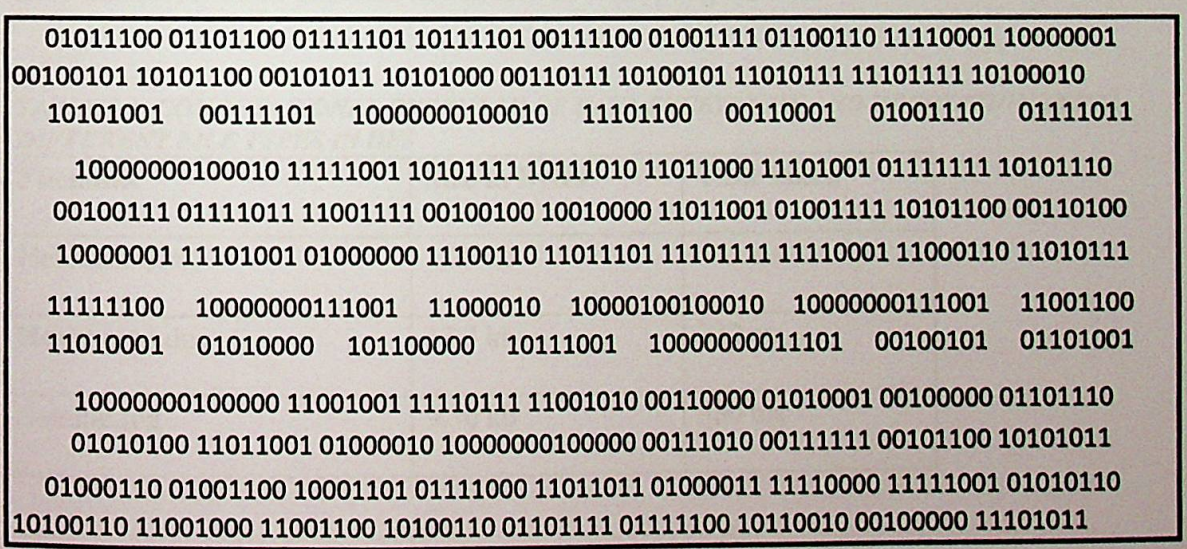


Fig. 5.7: Encryption results in Binary form

5.3.6 Decryption

In the decryption process, the secret key which was used to encrypt the file is used to decrypt the file and displays the information in a text area. "String plaintext Str = new String (plaintext)" method decrypts a file. Trying to access a file without a key will result in an error. For the user to decrypt a file they need to request for the secret key which was used to encrypt the file from the data owner. Fig. 5.8 shows the file after decryption.

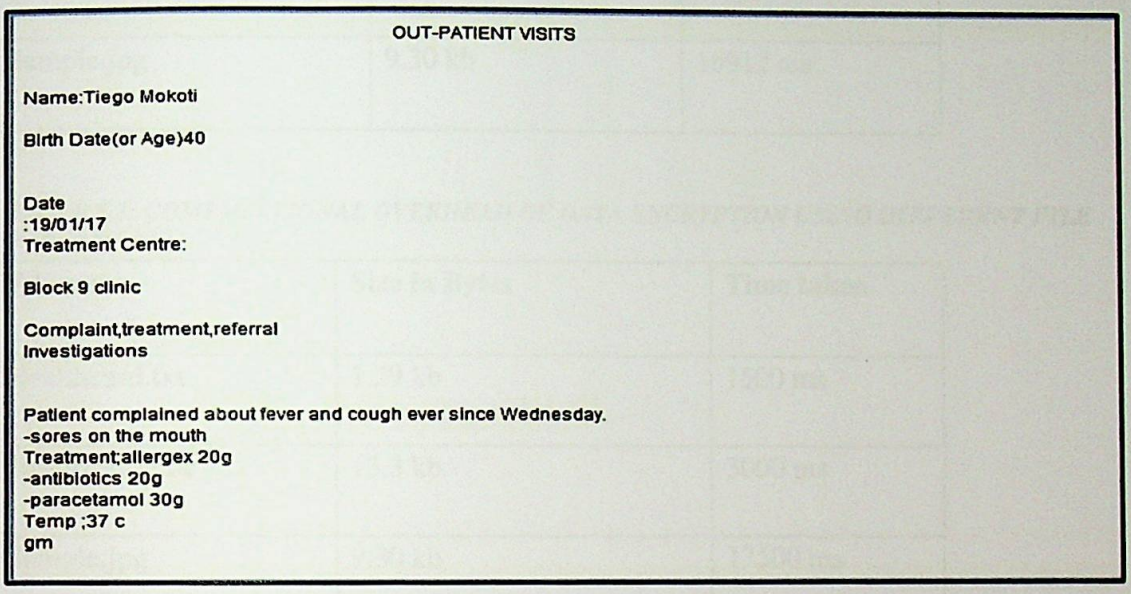


Fig. 5.8: Patient Health Record after Decryption

5.4: COMPARISON OF THE TRIPLE DES WITH OTHER ALGORITHMS

This section presents an evaluation of the system using different algorithms (i.e. Triple DES, DES and RSA). The parameters used in this evaluation are encryption time and decryption time. The comparison was done using different file types of different sizes. Table 5.1, 5.2 and 5.3 shows the computational overheads of three different encryption algorithms (Triple DES, DES and RSA) when implemented in the self-encrypting system.

TABLE 5.1: COMPUTATIONAL OVERHEAD OF DATA ENCRYPTION AND DECRYPTION USING DIFFERENT FILE TYPES IN DES

Filename	Size in Bytes	Time taken
Healthcard.txt	1.29 kb	413 ms
Healthcard.doc	13.3 kb	332 ms
Sample.jpg	9.30 kb	330 ms

TABLE 5.2: COMPUTATIONAL OVERHEAD OF DATA ENCRYPTION USING DIFFERENT FILE TYPES IN TRIPLE DES

Filename	Size in Bytes	Time taken
Healthcard.txt	1.29 kb	327 ms
Healthcard.doc	13.3 kb	22648 ms
Sample.jpg	9.30 kb	16912 ms

TABLE 5.3: COMPUTATIONAL OVERHEAD OF DATA ENCRYPTION USING DIFFERENT FILE TYPES IN RSA

Filename	Size in Bytes	Time taken
Healthcard.txt	1.29 kb	1500 ms
Healthcard.doc	13.3 kb	3000 ms
Sample.jpg	9.30 kb	17500 ms

5.4.1 Encryption Time

For encryption time we compare DES, Triple DES and RSA by looking at the time it takes for them to encrypt different types of files. Fig. 5.9 shows the encryption times for these algorithms. From the bar chart in Fig. 5.9 it can be concluded that RSA takes the highest time to encrypt and DES takes the least time. This is because RSA is very slow, especially in encrypting large files hence it is often used to encrypt small amounts of data like session keys. From the results we can conclude that there is a trade-off between security and encryption, the more secure the algorithm the more time it takes to encrypt, and the less secure algorithms takes less time like the DES algorithm.



Fig. 5.9: Encryption Time

5.4.2 Decryption Time

This is the time taken to convert the cipher text back into plaintext. The algorithms were compared using different file types. Fig. 5.10 shows decryption times. From the bar chart in Fig. 5.10 it shows that the decryption time for all the algorithms is less, RSA being the algorithm which takes more time to decrypt.

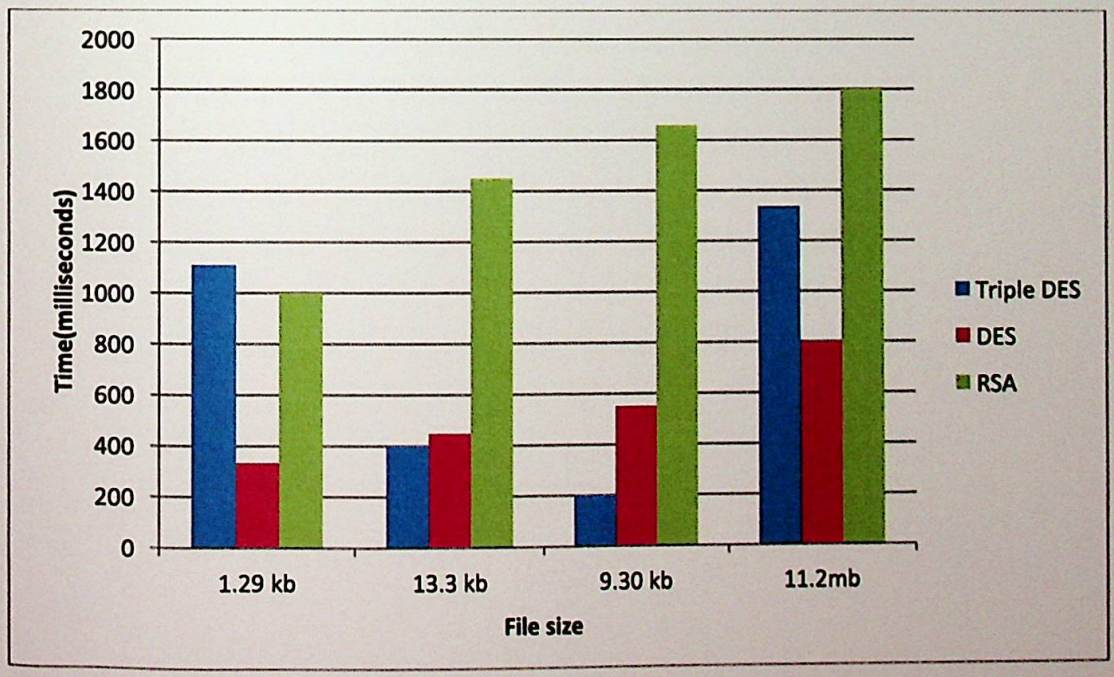


Fig. 5.10: Decryption Time

5.4.3 Comparative analysis of results from other systems with the result from the self-encrypting system

This section compares results of different models with the self-encryption model.Table 5.4 summarises the results of different models produced by different authors to address security issues in healthcare cloud computing. All the models uses different encryption schemes to provide security to the healthcare records. These models focus on securing data in the cloud only, as compared to the self-encrypting system which addresses issues at the client's side. The ESPAC scheme allows users to have different access privileges thus ensuring privacy of users and also protects data against eavesdropping and collusion attacks. SECHA provides an environment where multiple users can access the file with different access policies. The owner of the PHR is able to encrypt the file and stores it in the cloud. When access request is made, the policy is checked to see whether the requester has the required pseudo-role. SECHA has a similar concept with the self-encrypt system, the difference is in the area where security is applied. Based on this results it can be concluded that the self-encryption system is a better solution to providing security for data which is on transit to the cloud.

Model and reference	Description	Results
Secure Dynamic Access Control Scheme of PHR in Cloud Computing [47]	A PHR access control scheme with Lagrange interpolation polynomial to establish Patient Health Record access which is secure.	• The designed scheme protects against Equation Attacks by proposing three dynamic updates. • Protects against external Attacks • The proposed PHR in Cloud computing environments covers numerous external users, in addition to the legal multi-users. Illegally authorized external personnel need to obtain the decryption key with the public parameters for useful patients' records or medical information so that the encrypted medical files would become meaningful PHR after the decryption. • Safeguards PHR against collaborative attack
Enabling Security and Patient-centric Access Control for eHealth in cloud computing(ESPAC) [5]	A scheme which allows data requesters to have different access privileges based on their roles, and then assigns different attribute sets to them.	• The ESPAC scheme ensures user and eHealth agent's identity privacy • Data transmissions from user to health agent, as well as from health agent to health cloud service provider are done with proper encryption schemes • The scheme is resistant to the eavesdropping and collusion attacks • The scheme ensures message integrity, non-repudiation, and

		source authentication • The scheme prevents user to access the plaintext before providing the required attributes that satisfy the access policy.
Secured e-Health System(SECHA) [32]	A Secured Cloud based aided medical system is a program designed to aid standard and effective use and access of patient records anytime it is required.	• PHR is stored in the cloud, and can be accessed through a web portal by multiple owners and users. • The patient which is the owner of the PHR encrypts it and stores the ciphertext in the cloud server. • When an access request is made, the policy associated with the requested object is first checked to see whether the requester (user) has the required pseudorole or not. • If the requester holds the right pseudorole, rules within the policies are then checked for additional constraints to approve or deny the access request. With the use of this proposed system large number of patient will be attended to at a limited time and the patient records can be accessed anywhere in the world with security of the patient data ensured.
Self-encryption system	A system which provides security at the client's side.	• Ensures confidentiality and privacy by encrypting data before it is sent to the cloud • Protects data against man-in –the middle attacks • Secures data at the client's side

5.5 CONCLUSION

This chapter presented the results of node availability and results of the proposed encryption system. The results from the node availability model showed the number of personnel needed for the full operation of the machines. This can help in making better decisions regarding availability of the system. The results of self-encryption system were presented in the form of an encrypted file encrypted with Triple DES encryption algorithm and a plaintext file after successful encryption. For the exchange of keys, RSA algorithm was used to safeguard against attacks to the session keys. Lastly, the three algorithms were compared in terms of encryption time and decryption time using different file extensions. RSA proved to be the algorithm which takes more time for encryption and decryption as compared to others. DES was the least concerning the encryption and decryption time. There is a trade-off between security and time in relation to DES. DES has proven not to be secure but efficient hence Triple DES is a better choice hence it's secure and does not take a lot of time to encrypt as compared to RSA. This chapter also compared the results of systems similar to the self-encryption system. Among all these systems the self-encryption system has proven to be the only one which addresses security issues at the client's side.

CHAPTER 6

CONCLUSIONS AND FUTURE WORKS

This chapter summarizes the thesis, discusses its main findings and contributions, points out the limitations of the current work and gives directions for future work. The aim of the thesis was to design and implement a system that will address the security issues in healthcare cloud, by enhancing the current security implemented on the client's side.

6.1 CONCLUSIONS

An extensive literature review of the security and availability issues in the cloud was carried out, in which the finding was presented in Chapter 2. From this study, privacy and confidentiality have proven to be the main security issues in healthcare cloud. Existing security models were evaluated to find gaps. The evaluation proved that little security was done on the client side hence making data to be vulnerable to attacks before being sent to the cloud. The chapter further discussed the encryption algorithms. The methods and techniques used in this research work were outlined in Chapter 3 together with the proposed security and availability models. The main contributions of this study are the implementation of the security enhancement model, and also the availability model which shows the number of personnel needed for a certain number of machines to be operational was discussed in Chapter 5. This model helps in making better decisions concerning the number of personnel required to repair the machines. The self-encryption system results were presented which showed the file after applying the Triple des encryption and after successful decryption. RSA algorithm was used for key distribution in order to safeguard the file against man-in-the-middle attacks. Different encryption algorithms were evaluated using different parameters. The three encryption algorithms were also compared, and Triple DES proved to be the fairest among them all since it's more secure and efficient as compared to RSA which is slow and DES which is not secure.

6.2 FUTURE WORKS

As a future work, the input function for the graph and weight of graph need more attention for achieving the complete availability model. Due to the complexity of the cloud, it will be difficult to achieve end-to-end security. In addition, a database is needed in order to store the keys and the users who logs into the system. Currently there is no database, which makes the exchange and storing of keys more tedious.

6.2.1Homomorphic encryption

Fully homomorphic encryption can be adopted to enhance the security of the self-encrypting system. Homomorphic encryption solves the confidentiality problems in cases where data is

shared by multiple users, who perform different operations on it. It also ensures privacy of data by allowing the users to directly work on data without the need for decryption and also in the healthcare patients can be given treatment after analysis of the disease without disclosing their details [86].

6.2.2 Honey encryption

Another important direction in security is the introduction of honey encryption. Most encryption algorithms use n-bit keys where the security of the algorithm increases with the length of the keys. Although these algorithms are considered secure, they are also susceptible to brute force attacks. The decryption of a cipher text can be achieved through brute-force guessing of keys .The honey encryption can be used as a countermeasure against such attacks. A cipher text can be further honey encrypted in such a way that an attempted decryption with invalid keys yields valid-looking output messages. Thus attackers trying to guess keys through brute-force gains no information [87].

6.2.3 Implementing the algorithm in parallel on a multicore GPU using open CL

Open CL (Open Compute Language) is amongst the commonly used cross platform parallel programming frameworks. With an advancement in modern multicore CPUs (Central Processing Units) and heterogeneous architecture, significant performance improvements can be seen in parallel computing using OpenCL [88]. To improve the performance of the self-encryption system and also to ensure portability the system can be implemented using OpenCL. Since the system deals with a lot of information about patients, it is necessary for it to be able to meet up with the demands of the users, hence high performance is important.

REFERENCES

[1] Ahmad-Reza Sadeghi, Marcel Winandy lohr, "Securing the E-Health Cloud," in *1st ACM International Health Informatics Symposium (IHI 2010)*, 2010.

[2] Vladimir Trajkovik Goce Gavrilov, "Security and Privacy Issues and Requirements for Healthcare Cloud Computing," in *ICT Innovations 2012 Web* Proceedings, 2012.

[3] P. Herbert Raj, P. Jelciana P. Ravi Kumar, "Exploring Data Security Issues and Solutions in Cloud Computing," in *6th International Conference on Smart Computing and Communications*, india, 2017.

[4] Radziszowski, Raj Alsheri, "Designing a Secure Cloud-Based EHR System using CiphertextPolicy Attribute-Based Encryption," 2012.

[5] Liang Barua, "ESPAC: Enabling Security and Patient -centric Access Control for eHealth in cloud computing," *Int.J. Security and networks*, April 2011.

[6] Grance mell, "The NIST Definition of Cloud Computing," pp. 800-145, september 2011.

[7] Z. Al-Khanjari, A. Al-Ani, and S. Al-Hermizy, "A PROPOSED SECURITY ARCHITECTURE FOR ESTABLISHING PRIVACY DOMAINS IN E-HEALTH CLOUD," *European Scientific Journal*, vol. vol.2, pp. 1857 – 7881, June 2014.

[8] Mogasala Leelarani, Y Ramesh Kumar Ch Chakradhara Rao, "Cloud: Computing Services And Deployment Models," *International Journal of Engineering And Computer Science*, vol. Volume 2, no. Issue 12, pp. 3389-3392, December 2013.

[9] and Ling Liu Rui Zhang, "Security Models and Requirements for Healthcare Application Clouds," in *IEEE International Conference on Cloud Computing, CLOUD 2010*, Miami, 2010.

[10] Chen, Nepal, Calvo, Alem Thilakanathana, "A platform for secure monitoring and sharing of generic health data in the Cloud," *Future Generation Computer Systems*, pp. 102–113, 2014.

[11] Syed Sabir Salman-Al-Musawi, Khairul Anam, Nafiul Rashid Fakhrul, "A Secured Cloud based Health Care Data Management System ," *International Journal of Computer Applications* , vol. volume 49, no. 12, july 2012.

[12] Abdulaziz, Garraghan, Townend, Lau, Xu Hussain, "Multi-Tenancy in Cloud Computing," in *Proceedings of the 8th IEEE International Symposium on Service-Oriented System Engineering*, oxford, 2014, pp. 344 - 351.

[13] Jaydip Sen, "security and privacy issues in cloud computing," 2013.

[14] Allen Aldossary, "Data Security, Privacy, Availability and Integrity in Cloud Computing: Issues and Current Solutions," *(IJACSA) International Journal of Advanced Computer Science and Applications*, vol. vol 7, no. no 4, 2016.

[15] Patra, Satapathy Padhy, "Cloud Computing: Security Issues and Research Challenges,"

International Journal of Computer Science and Information Technology & Security (IJCSITS), vol. vol 1, no. no 2, pp. 136-146, December 2011.

[16] Sindhu, Jesus Sanjay, "A Survey of the State of Cloud Computing in Healthcare," *Network and Communication Technologies,* vol. vol 1, no. no 2, September 2012.

[17] Sasikumar Shaikh, "Security Issues in Cloud Computing: A survey," *International Journal of Computer Applications,* vol. volume 44, no. 19, April 2012.

[18] S. Juare, "Survey on Data Security and Integrity Issues in Cloud Computing," *IOSR Journal of Computer Engineering,* pp. 59-62, 2016.

[19] Xiao Y Xiao Z, "Security and Privacy in Cloud Computing," *IEEE COMMUNICATIONS SURVEYS & TUTORIALS*, vol. vol 15, no. no 2, 2013.

[20] A M. Khan, "A survey of security issues for cloud computing," *Journal of Network and Computer Applications*, pp. 11-29, 2016.

[21] Ding, Winer,Yao Wu, "Network Security for Virtual Machine in Cloud Computing," in *International Conferenceon Computer Sciences and Convergence Information Technology(ICCIT)*, 2010, pp. 18–21.

[22] Yang, Ma,Dai Luo, "Virtualization Security Risks and Solutions of Cloud Computing Via Divide-
Conquer Strategy," in *Third International Conference on Multimedia Information Networking and Security*, 2011.

[23] Masood, Shibli, Niazi Habiba1, "Cloud identity management security issues & solutions: a taxonomy," *Complex Adaptive Systems Modelling,* 2014.

[24] Jalali Masdari, "A survey and taxonomy of DoS attacks in cloud," *SECURITY AND COMMUNICATION NETWORKS*, July 2016.

[25] Amirgholipour, Alirezanejad,Aski, Ghiami Sharifi, "Availability challenge of cloud system under DDOS attack," *Indian Journal of Science and Technology*, vol. vol 5, no. 6, june 2012.

[26] Rosado, Fernández-Medina,Fernandez Hashizume, "An analysis of security issues for cloud computing," *Journal of Internet Services and Applications* , 2013.

[27] Grundy, Muller mohamed Al Morsy, "an analysis of the cloud computing security problem".

[28] Rosado, Fernández-Medina,Fernandez Hashizume, "An analysis of security issues for cloud computing," *Journal of Internet Services and Applications* , 2013.

[29] Salman-AL-Musawi, Anam,Rashid onik, "A secured Cloud based Health Care Data Management System," *International Journal of Computer Applications*, vol. 49, no. 12, July 2012.

[30] Chana Kaur PD, "Cloud Based Intelligent System for Delivering Healthcare as a Service," *ComputMethods Programs Biomed*, pp. 346-359, 2014.

[31] Chen, Nepal,Calvo,Alem Thilakanathan, "A platform for secure monitoring and sharing of generic health data in the cloud," *Future Generation Computer Systems*, pp. 102-113, 2014.

[32] Arowolo Ikuomola, "Securing Patient Privacy in E-Health Cloud Using Hormomophic Encryption," *International Journal of Computer Networks and Communications Security*, vol. vol 2, no. 1, pp. 15-21, January 2014.

[33] Ashraf Hussain, "Security Issues in Cloud Computing - A Review," *Int. J. Advanced Networking and Applications*, vol. volume 6, no. issue 2, pp. 2240-2243, 2014.

[34] E M Hernandez-Ramirez, V J Sosa-Sosa, and I Lopez-Arevalo, "A Comparison of Redundancy Techniques for Private and Hybrid Cloud Storage," *Journal of Applied Research and Technology*, pp. 893-901, 2012.

[35] sosa-sosa, Lopez-Arevalo Hernandez-Ramirez, "A Comparison of Redundancy Techniques for Private and Hybrid Cloud Storage," *J. Applied Research and Technology*, vol. 10, pp. 893□901.

[36] A S Gughane, P L Hasija,S P Katpale Deshmukh P.M, "Maintaining file storage Security in cloud computing," *international journal of Emerging Technology and Advanced Engineering*, vol. vol 2, no. 10, October 2012.

[37] Mar K K, "Secured virtual diffused file system for the cloud.," in *6th international conference on internet technology and secured transactions*, Abu Dhabi, december 2011, pp. 116-121.

[38] Gurpreet Singh and Supriya, "A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security," *International Journal of Computer Applications*, vol. 67, no. 19, April 2013.

[39] Akashdeep Bhardwaja, GVB Subrahmanyamb, Vinay Avasthi, and Hanumat Sastryd, "Security Algorithms for Cloud Computing," in *International Conference on Computational Modeling and Security (CMS 2016)*, 2016, pp. 535 – 542.

[40] Prerna Mahajan and Abhishek Sachdeva, "A Study of Encryption Algorithms AES, DES and RSA for Security," *Global Journal of Computer Science and Technology Network, Web & Security*, vol. 13, no. 15, 2013.

[41] Islam.K. V Nasarul and Riyas.K. V Mohamed, "Analysis of Various Encryption Algorithms in Cloud Computing," *International Journal of Computer Science and Mobile Computing*, vol. 6, no. 7, pp. 90-97, July 2017.

[42] Tatiana Ermakova and Benjamin Fabian, "Security and Privacy System Requirements for Adopting Cloud Computing in Healthcare Data Sharing Scenarios," in *Proceedings of the Nineteenth Americas Conference on Information Systems*, Chicago, 2013.

[43] Ming Li, Shucheng Yu, Yao Zheng, Kui Ren, and Wenjing Lou, "Scalable and Secure Sharing of Personal Health Records in Cloud Computing using Attribute-based Encryption," *IEEE Transactions on Parallel and Distributed Systems*, vol. xx, 2012.

[44] S Sujoy Basu et al., "Fusion: Managing Healthcare Records at Cloud Scale," *IEEE Computer Special Issue on Move Toward Electronic Health Records.*, 2012.

[45] Lingfeng Chen and Doan B Hoang, "Novel data protection model in healthcare cloud," in *IEEE International Conference on High Performance Computing and Communications*, 2011.

[46] Rui Zhang and Ling Liu, "Security Models and Requirements for Healthcare Application Clouds," in *IEEE 3rd International Conference on Cloud Computing*, 2010, pp. 268-275.

[47] Tzer-Shyong Chen et al., "Secure Dynamic Access Control Scheme of PHR in Cloud Computing," *J Med Syst,* August 2012.

[48] Doan Hoang and Dang Thanh Dat, "Health Data in Cloud Environments," in *PACIS 2015 Proceedings*, 2015.

[49] Eman AbuKhousa, Nader Mohamed, and Jameela Al-Jaroodi, "e-Health Cloud: Opportunities and Challenges," *Future Internet 2012*, pp. 621-645, 2012.

[50] Allamaprabhu G Rudraxi and Nayak S.K Parikshith, "A Novel Patient Centric Framework for Data Access Control in Semi-trusted Cloud Servers," *International Journal of Computer Science and Mobile Computing*, vol. III, no. 6, pp. 1-10, June 2014.

[51] José Luis Fernández-Alemán, Inmaculada Carrión Señor, Pedro Ángel Oliver Lozoya, and Ambrosio Toval, "Security and privacy in electronic health records: A systematic literature review," *Journal of Biomedical Informatics*, pp. 541–562, 2013.

[52] T-L Chen, Chen C, J-G Bau, and TC Lin, "Secure Dynamic Access Control of PHR in Cloud Computing," *Jounal of Medical Systems*, pp. 4005-4020, 2012.

[53] Hieu Vu Tran, Binh T.H. Nguyen, and Ngoc Dung Bui, "Cloud computing - a review on benefits, challenges and applications," *Science, Engineering & Education*, pp. 41-49, 2017.

[54] Suhail Qadir and S. M. K. Quadri, "Information Availability: An Insight into the Most Important Attribute of Information Security," *Journal of Information Security*, pp. 185-194, April 2016.

[55] K D Bowers, A Juels, and A Oprea, "HAIL:High Availability and Integrity Layer for Cloud Storage," in *Procedings of the 16th Conference on Computer and Communications Security*, chicago, 2009, pp. 187-198.

[56] P M Deshmukh, A S Ghughane, P L Hasija, and S P Katpale, "Maintaining File Storage Security in Cloud Computing," *International Jornal of Emerging Technologies and Advanced Engineering*, vol. 2, no. 10, october 2012.

[57] J Sen, "Security and Privacy Issues in Cloud Computing," *Architectures and Protocols for Secure Information Technology*, 2013.

[58] D A Menasce, "Perfomance and Availability of Internet Data Centres," *IEEE Internet Computing*, pp. 94-96, 2004.

[59] M Joini, A B Aissa, LBA Rabai, and A Mili, "Towards Quantitative Measures of Information

Security:A Cloud Computing Case Study," *International Journal of Cyber-Security and Digital Forensics*, pp. 248-262, 2012.

[60] Kumar B Praveen and N.S Rajaanadan, "Data Encryption and Decryption Using By Triple DES Performance Efficiency Analysis of Cryptosystem," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. iv, no. 3, March 2016.

[61] Punita Meelu and Rajni Meelu, "IMPLEMENTATION OF PUBLIC KEY CRYPTOGRAPHIC SYSTEM: RSA," *International Journal of Information Technology and Knowledge Management*, vol. 5, no. 2, pp. 239-242, July 2012.

[62] Shireen Nisha and Mohammed Farik, "RSA Public Key Cryptography Algorithm -A Review," *International Journal of Scientific and Technology Research*, vol. 6, no. 07, July 2017.

[63] Shireen Nisha and Mohammed Farik, "RSA Public Key Cryptography Algorithm – A Review ," *International Journal of Scientific & Technology Research*, vol. 6, no. 07, July 2017.

[64] K. Krippendorff, "Sampling Techniques," *Fundam*, vol. 1, p. 209, 2014.

[65] Radziszowski, and Rajendra K. Raj Alshehri, "Designing a Secure Cloud-Based EHR System using Ciphertext-Policy Attribute-Based Encryption".

[66] Ubale Khairnar, "Cloud Computing Security Issues And Challenges ," *International Refereed Journal of Engineering and Science (IRJES)*, vol. volume 2, no. issue 2, pp. 01-13, february February 2013.

[67] Mitrokotsa Douligeris, "DDoS attacks and defense mechanisms: classification and state-of-theart," *Computer Networks*, pp. 643-666, 2004.

[68] Tim Polk, Kerry McKay, and Santosh Chokhani, "Guidelines for the Selection,Configuration, and Use of Transport Layer Security (TLS) Implementations," *NIST Special Publication*, April 2014.

[69] Priyadarshini Patil, Prashant Narayankar, D G Narayan, and S M Meena, "Acomprehensive Evaluation of Cryptographic Algorithms: DES, 3DES, AES, RSA and Blowfish," in *International Conference on Information Security & Privacy(ICISP2015)*, INDIA, 2016, pp. 617-624.

[70] Kefa Rabah, "Theory and Implementation of Data Encryption Standard:A Review," *Information Technology Journal*, vol. v, pp. 307-325, 2005.

[71] Razieh Mokhtarnameh, Nithiapidary Muthuvelu, Sin Ban Ho, and Ian Chai, "A Comparison Study on Key Exchange-Authentication protocol," *International Journal of Computer Applications*, vol. 7, no. 5, September 2010.

[72] Somya Garg, Tarun Garg, and Bhawna Mallick, "Secure Message Transfer using Triple DES," *International Journal of Computer Applications* , vol. 165, no. 8, May 2017.

[73] Hamdan O Alanazi et al., "New Comparative Study Between DES, 3DES and AES within Nine Factors," *JOURNAL OF COMPUTING*, vol. ii, no. 3, MARCH 2010.

[74] Ritu Tripathi and Sanjay Agrawal, "Comparative Study of Symmetric and Asymmetric Cryptography Techniques," *International Journal of Advance Foundation and Research in Computer (IJAFRC)*, vol. I, no. 6, June 2014.

[75] Suriya kala L and Dr R Thangaraj, "A Study on different Image Encryption Algorithms ," *International Journal of Computer Science and Information Technologies, Vol. 5 (2), 2014, 1204-1206*, vol. 5, no. 2, pp. 1204-1206, 2014.

[76] Arun Hrishikesh Deshpande, "HoneyMesh: Preventing Distributed Denial of Service Attacks using Virtualized Honeypots," *International Journal of Engineering Research & Technology (IJERT)*, vol. 4, no. 8, August 2015.

[77] Khaled M Elleithy, Drazen Blagovic, Wang Cheng, and Paul Sideleau, "Denial of Service Attack Techniques: Analysis, Implementation and Comparison," *SYSTEMICS, CYBERNETICS AND INFORMATICS*, vol. III, no. 1.

[78] Akhil Arya, "SECURITY ENHANCEMENT USING TRIPLE DES ALGORITHM ," *International Journal of Computer Science and Mobile Computing*, vol. 6, no. 4, pp. 353-355, April 2017.

[79] Kehe Wu, Yakun Zhang, Wenchao Cui, and Ting Jiang, "Design and Implementation of Encrypted and Decrypted File System Based on USBKey and Hardware Code," in *AIP Conference Proceedings*, 2017.

[80] Shaik Musthafa and Dora Babu Sudarsa, "Patient–Centric Secure Data Sharing Frame Work for Cloud-Based PHR Systems ," *International Journal of Engineering Science Invention* , vol. II, no. 5, pp. 17-26, May 2013.

[81] S Karthik and A Muruganandam, "Data Encryption and Decryption by Using Triple DES and Performance Analysis of Crypto System," *International Journal of Scientific Engineering and Research (IJSER)*, vol. 2, no. 11, November 2014.

[82] "Data Encryption and Decryption Using By Triple DES Performance Efficiency Analysis of Cryptosystem," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 4, no. 3, March Vol. 4, Issue 3, 2016.

[83] Muhammad Faheem Mushtaq et al., "A Survey on the Cryptographic Encryption Algorithms," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 11, 2017.

[84] M Parekh and B Saleena, "Designing Cloud Based Framework for Healthcare System and Applying Clustering Techniques for Region Wise Diagnosis," in *Procedia-Computer Science*, 2015, pp. 537-542.

[85] Singh Narjeet.Raj Gaurav."Security on Bccp Through Aes Encryption Technique".International Journal Of Engineering Science & Advanced Technology Volume 2, Issue-4, 813-819.pp. 817

[86] Rahul Mohd, Alhumyani Hesham A. , Muntjir Mohd , Kambojl Minakshi."An improved Hormomophic Encryption for Secure Cloud Data Storage". *International Journal of Advanced Computer Science and Applications*,vol 8,no.12, 2017

[87] Tyagi Nirvan,Wang Jessica,Wen Kevin,Zuo Daniel."Honey Encryption Algorithms,Implementation of an Encryption Scheme Resielent to Brute-force attacks". *Computer and Network Security*,2015

[88] Zaidi Salman."Performance of OpenCL in MultiCore Processors"

```
package login; import
java.io.BufferedReader; import
java.io.File;

import java.io.FileInputStream;

import java.io.FileOutputStream;
import java.io.FileReader;

import java.io.IOException;
 import java.util.logging.Level; import
java.util.logging.Logger; import
javax.swing.JFileChooser; import
javax.swing.JOptionPane; import
java.security.MessageDigest; import
java.util.Arrays; import java.util.Date;
import javax.crypto.Cipher; import
javax.crypto.SecretKey; import
javax.crypto.spec.IvParameterSpec; import
javax.crypto.spec.SecretKeySpec;
/**
 *
 * @author user
 */ public class encryption extends
javax.swing.JFrame {
   String key ="java**12ff";
   /**
 * Creates new form encryption
   */
```

```
    private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
      System.exit(0);
  }
  private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
      try{
      JFileChooser fc=new JFileChooser();
      fc.showOpenDialog(null);
      String path=fc.getSelectedFile().getAbsolutePath();
      //jLabel2.setText(path);
      File f=fc.getSelectedFile();
      FileInputStream fis=new FileInputStream(f);
      FileOutputStream fos=new FileOutputStream("encrypted.txt");
      Thread.sleep(2000);
        long elapsedTime = 0L;
      long startTime = System.currentTimeMillis();
      DESEncrypt.encrypt(key, fis, fos);
      elapsedTime = (new Date()).getTime() - startTime;
       System.out.println(elapsedTime);
      jTextArea2.setText("C:\\Users\\Documents\\NetBeansProjects\\login\\encrypted.txt");
      }catch(Exception e)
{
        System.out.print("Exception in EncryptTXT");
      }

  }
```

```
 private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
JFileChooser chooser= new JFileChooser();
chooser.showOpenDialog(null);
    File f =chooser.getSelectedFile();
    String filename = f.getAbsolutePath();
       try
        {
           FileReader reader= new FileReader(filename);
           BufferedReader br = new BufferedReader (reader);
jTextArea1.read(br,null);            br.close();
jTextArea1.requestFocus();
          }
       catch(IOException e){
         JOptionPane.showMessageDialog(null,e);
       }
  }
   private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
decrypt ds =new decrypt();      ds.setVisible(true);      this.dispose();
     System.out.println("this is the ");
```

```
   System.out.print("Exception in DecryptTXT");

      }

// TODO add your handling code here:

   }
     private void jButton3ActionPerformed(java.awt.event.ActionEvent
evt) {

      Runtime runtime=Runtime.getRuntime();

try {

 runtime.exec("explorer.exe C:\\Users\\raks\\Documents\\NetBeansProjects\\DESEncryption");


      } catch (IOException ex) {

         Logger.getLogger(encryption.class.getName()).log(Level.SEVERE, null, ex);

     }

   }

   public static void main(String args[]) {
java.awt.EventQueue.invokeLater(() -> {
new encryption().setVisible(true);
      });


   }


   // Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JButton jButton4;
private javax.swing.JButton jButton5;
private javax.swing.JButton jButton6;
   private javax.swing.JCheckBoxMenuItem jCheckBoxMenuItem1;
     private javax.swing.JFileChooser
jFileChooser1;
     private javax.swing.JLabel
jLabel1;
```

DESEncrypt.java

```
package login;

import java.io.BufferedReader;

import java.io.File;

import java.io.FileReader;
 import java.io.IOException;

import java.io.InputStream;
import java.io.OutputStream;
 import javax.crypto.Cipher;
import javax.crypto.CipherInputStream;

import javax.crypto.CipherOutputStream;

import javax.crypto.SecretKey;

import javax.crypto.SecretKeyFactory;

 import javax.swing.JFileChooser;

 import javax.swing.JOptionPane;

 import java.security.MessageDigest;

import java.util.Arrays;

 import java.util.Date;

import javax.crypto.spec.DESKeySpec;

 import javax.crypto.spec.IvParameterSpec;

 import javax.crypto.spec.SecretKeySpec;

public class DESEncrypt {

    public static void encrypt(String key, InputStream is, OutputStream os) throws Exception {

encryptOrDecrypt(key, Cipher.ENCRYPT_MODE, is, os);

    }
```

```
public static void decrypt(String key, InputStream is, OutputStream os) throws Exception {

encryptOrDecrypt(key, Cipher.DECRYPT_MODE, is, os);

  }

  public static void encryptOrDecrypt(String key, int mode, InputStream is, OutputStream os) throws
Exception {

 DESKeySpec dks = new DESKeySpec(key.getBytes());

 SecretKeyFactory skf = SecretKeyFactory.getInstance("DES");

 SecretKey desKey = skf.generateSecret(dks);

Cipher cipher = Cipher.getInstance("DES");

     if (mode == Cipher.ENCRYPT_MODE) {

cipher.init(Cipher.ENCRYPT_MODE, desKey);

CipherInputStream cis = new CipherInputStream(is, cipher);

 makeFile(cis, os);

 } else if (mode == Cipher.DECRYPT_MODE) {

 cipher.init(Cipher.DECRYPT_MODE, desKey);

CipherOutputStream cos = new CipherOutputStream(os, cipher);

 makeFile(is, cos);

}

}

  public static void makeFile(InputStream is, OutputStream os) throws IOException {

  byte[] bytes = new byte[64];

int numBytes;

 while ((numBytes = is.read(bytes)) != -1) {

  os.write(bytes, 0, numBytes);

 }
```

```
os.flush();

os.close();
```