

DECLARATION

I hereby declare that the thesis entitled “**Secure Healthcare Services using Replicated Kerberos**” which is being submitted to the Delhi Technological University, in partial fulfillment of the requirement for the award of degree of Master of Technology in Software Engineering is an authentic work carried out by me.

Chauthmal Vaibhav Manikrao

(2K12/SWE/11)

Department of Computer Engineering

Delhi Technological University

Delhi – 110042



**Delhi Technological University
(Government of Delhi NCR)
Bawana Road, New Delhi-42**

CERTIFICATE

This is to certify that the work being presented in Major Project - I entitled “**Secure Healthcare Services using Replicated Kerberos**” by *Chauthmal Vaibhav Manikrao* (2K12/SWE/11), is authentic record of work carried out under my guidance and supervision and refers other researcher’s work which are duly listed in the reference section.

Ms. Divyashikha Sethia
Assistant Professor
Department of Computer Engineering
Delhi Technological University
Delhi - 110042

ACKNOWLEDGEMENT

With due regards, I hereby take this opportunity to acknowledge a lot of people who have supported me with their words and deeds in completion of my research work as part of this course of Master of Technology in Software Engineering.

A special thanks to my mentor and project guide, **Ms. Divyashikha Sethia**, Asst. Professor, Department of Software Engineering, Delhi Technological University, for her valuable guidance, stimulating suggestions and encouragement, helped me to make our ideas come into reality.

I wish to convey my sincere gratitude to our Head of Department, and all the faculties and PhD. Scholars of Computer Engineering Department, Delhi Technological University who have enlightened me during my project.

Also, I would like to appreciate the contribution and help provided to me by the juniors and staff working in LANS Lab.

Last but not least, I would like to thank my family and friends for their encouragement and persistent support.

Chauthmal Vaibhav Manikrao
M.Tech (Software Engineering)
(2K12/SWE/11)

ABSTRACT

It's been a very long years the healthcare industry has considered as a slow adaptor of technology. But in recent years, the picture is changing with adoption of new information technologies solutions. This got various benefits like improved hardware infrastructure, new advanced medical applications, improved data processing speed, etc. The new healthcare technologies are crucial part of medical treatment and follow up procedures.

To secure these healthcare services and to stop replay attacks is a big task. Here in this work we have deployed replicated Kerberos cryptographic servers. Kerberos will authenticate the service and the clients, while replication of Kerberos will provide reliability, by providing service even if one Kerberos server goes down due to some cause. Also Kerberos will mitigate the replay attacks. We have introduced the Intermediate Server as Load Balancer to share the authentication requests over the replicated Kerberos servers to improve the performance of authentication service in terms of resilience, better response time and availability of authentication service.

The Intermediate Server is implemented using Java RMI Server and all the Kerberos Server which gives authentication service as well as the clients requesting Kerberos authentication service are Java RMI Clients. Hence, the Intermediate Server comprises of implementation of various java interfaces which are requested by its clients i.e. Kerberos Server (KDC's) and Clients requesting its authentication service.

Keywords: Healthcare, Kerberos, Replication, Java Authentication and Authorization Service (JAAS), Generic Services Application Program Interface (GSS-API), Java Remote Method Invocation (RMI)

TABLE OF CONTENTS

DECLARATION	i
CERTIFICATE	ii
ACKNOWLEDGEMENT	iii
ABSTRACT	iv
TABLE OF CONTENTS	v
LIST OF FIGURES	ix
LIST OF TABLES	xi

CHAPTER 1

INTRODUCTION	1
1.1. BACKGROUND	1
1.2. MOTIVATION	1
1.3. THESIS OUTLINE	2

CHAPTER 2

LITERATURE SURVEY	3
--------------------------	----------

CHAPTER 3

RESEARCH BACKGROUND

3.1. KERBEROS	6
3.1.1. REQUIREMENTS	6
3.1.2. KERBEROS V4	6

3.1.3. KERBEROS V5	7
3.2. DESCRIPTION OF HOW KERBEROS WORKS	7
3.3. KERBEROS TERMINOLOGY AND CONCEPTS	7
3.3.1. REALMS, PRINCIPALS AND INSTANCES	7
3.3.2. SERVICE AND HOST PRINCIPALS	8
3.3.3. KERBEROS 4 PRINCIPALS	9
3.3.4. KERBEROS 5 PRINCIPALS	9
3.3.5. THE KEY DISTRIBUTION CENTER (KDC)	10
3.3.6. THE AUTHENTICATION SERVER (AS)	10
3.3.7. THE TICKET GRANTING SERVER (TGS)	10
3.3.8. TICKETS	10
3.3.9. KEYTAB FILES	10
3.4. KERBEROS AUTHENTICATION DIALOGUE	11
3.5. WHY KERBEROS?	12
3.6. REPLICATION	12
3.7. KERBEROS MASTER SLAVE REPLICATION	12
3.8. SOLUTION TO DATA INCONSISTENCY	13

CHAPTER 4

JAAS

4.1. INTRODUCTION TO JAAS	14
4.2. JAAS HIGH-LEVEL ARCHITECTURE	14
4.3. JASS CLASSES AND INTERFACES	15

CHAPTER 5

JAVA RMI

5.1. INTRODUCTION TO JAVA RMI 16

5.2. RMI CONCEPTS 16

CHAPTER 6

DESIGN ARCHITECTURE

6.1. PROPOSED ARCHITECTURE 18

6.2. KERBERIZED CLIENT AND APPLICATION SERVER 20

6.3. KERBERIZED CLIENT AND INTERMEDIATE SERVER 21

6.4. Kerberos Master-Slave Replication 22

CHAPTER 7

IMPLEMENTATION

7.1. KERBEROS SETUP 23

7.2. MASTER SLAVE REPLICATION 23

**7.2.1. METHODOLOGY USED TO INSTALL MASTER
KERBEROS 23**

7.2.2. INSTALL THE SLAVE KDC'S 26

7.3. KERBERIZED CLIENT AND APPLICATION SERVER 29

**7.4. IMPLEMENTATION OF SCALABLE MASTER SLAVE
REPLICATION 32**

7.5. INTERMEDIATE SERVER AND CLIENT IMPLEMENTATION 33

CHAPTER 8

TESTING AND RESULTS

8.1. TEST ENVIRONMENT	35
8.2. KERBEROS REPLICATED MASTER-SLAVE SERVERS	36
8.3. RMI INTERMEDIATE SERVER	37
8.4. RMI CLIENT ADD KDC	38
8.5. SERVICE SERVER	38
8.6. RMI CLIENTS	39
8.7. LOAD TESTING	40
8.8. COMPARISON OF AVERAGE KDC AUTHENTICATION TIME	49

CHAPTER 9

CONCLUSION AND FUTURE WORK

9.1. CONCLUSION	50
9.2. FUTURE WORK	51

REFERENCES	52
-------------------	-----------

APPENDIX	54
-----------------	-----------

LIST OF FIGURES

Fig. 1 : Operations of Kaman	3
Fig. 2 : Components of CTES Model	4
Fig. 3 : Messages used in CTES Model	4
Fig. 4 : Performance evaluation of Calculating Pi application on two different JMPI implementations	5
Fig. 5 : Relation between Primary, Instance and REALM	8
Fig.6 : Typical Form of Kerberos V5 Principal	8
Fig. 7 : Kerberos Authentication Process	11
Fig. 8 : JAAS High-Level Architecture	14
Fig. 9 : Basics of RMI	17
Fig. 10 : Proposed Architecture	19
Fig. 11 : Basic Flow between Kerberos KDC, Doctor and Service Server	20
Fig. 12 : Basic Flow of Intermediate Server	21
Fig. 13 : Kerberos Master-Slave Replication	22
Fig. 14: GSS-API Overview	31
Fig. 15 : Main Tasks of Intermediate Server	33
Fig. 16: Adding Principal into Kerberos Master Database	36
Fig. 17 : Kerberos Master Database Propagation to Slave	37
Fig. 18 : RMI Intermediate Server	37
Fig. 19 : RMI Client Add KDC	38

Fig. 20 : AppServer waiting for incoming connection	38
Fig. 21 : Testing of n-Clients at a once for Authentication	39
Fig. 22 : Generated Log containing Each Authentication Time of Clients	39
Fig. 23 : Principal Number VS Authentication Times	48
Fig. 24 : No. of Clients VS Avg. KDC Authentication Time	49

LIST OF TABLES

Table 1 : The Processing Speed according to No. of Computers used for executing SOR application	5
Table 2 : JAASS Classes and Interfaces	15
Table 3 : Configuration Details of All KDCs	35
Table 4.1 : Load Testing Result	40
Table 4.2 : Load Testing Result	41
Table 4.3 : Load Testing Result	42
Table 4.4 : Load Testing Result	43
Table 4.5 : Load Testing Result	44
Table 4.6 : Load Testing Result	45
Table 4.7 : Load Testing Result	46
Table 4.8 : Load Testing Result	47
Table 5 : Avg. KDC Authentication Time Comparison	49