

# KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY



KNUST

## Two-factor authentication in cloud computing: using kerberos with one-time password via sms

By:

**Robert Amo Otoo**

A thesis submitted to the Department of Computer Science, Kwame Nkrumah  
University of Science and Technology in partial fulfilment of  
the requirements for degree of

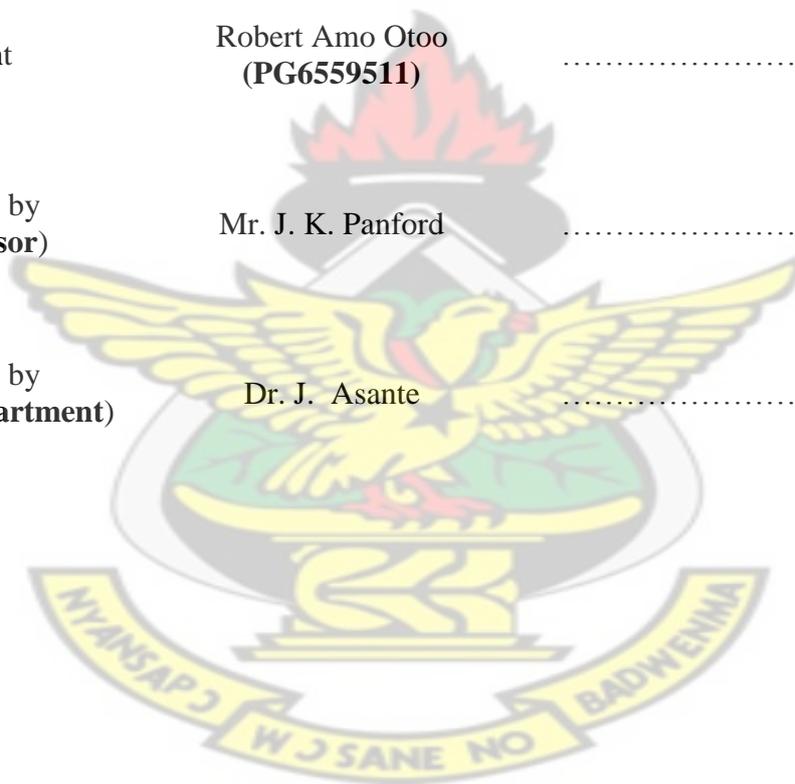
**Master of Philosophy**  
Information Technology

November 2014

# DECLARATION

I hereby declare that this thesis is my own work towards the Master of Philosophy and that to the best of my knowledge, it contains no material previously published by another person nor material which has been accepted for the award of any other degree of the University or elsewhere, except where due acknowledgement has been made in the text.

STATUS	NAME	SIGNATURE	DATE
Student	Robert Amo Otoo (PG6559511)	.....	.....
Certified by (Supervisor)	Mr. J. K. Panford	.....	.....
Certified by (Head of Department)	Dr. J. Asante	.....	.....



## ABSTRACT

Cloud computing is an emerging style of IT delivery that intends to make the Internet the ultimate home of all computing resources- storage, computations, and accessibility. It holds the promise of helping organizations because of its performance, high availability, least cost and many others. But the promise of the cloud cannot be fulfilled until IT professionals have more confidence in the security and safety of the cloud.

However cloud computing has the security issues such as service availability, massive traffic handling, application security and authentication. In this, user authentication requires high guaranteed security. To ensure secure authentication of client in the cloud environment, an effective method is being proposed.

This paper mainly focused on authentication issues in the cloud computing environment, where an enhancement is made to the intrusion login by Kerberos authentication service having One-Time Password via SMS as an added security base.

First, the model of the proposed services is described. Through this model all system entities that are necessary for managing and providing those authentication services are defined. Then, the design and specification of each service is described and explained. These services are based on existing and standardized security mechanisms and frameworks.

As a demonstration, a prototype java codes of an authentication service is developed and tested based on the designed authentication solution. It is shown that the model is feasible to secure against potential security risks associated with replay attacks, message information disclosure, message tampering, repudiation and impersonation.

**Keywords:** cloud computing, Kerberos, authentication

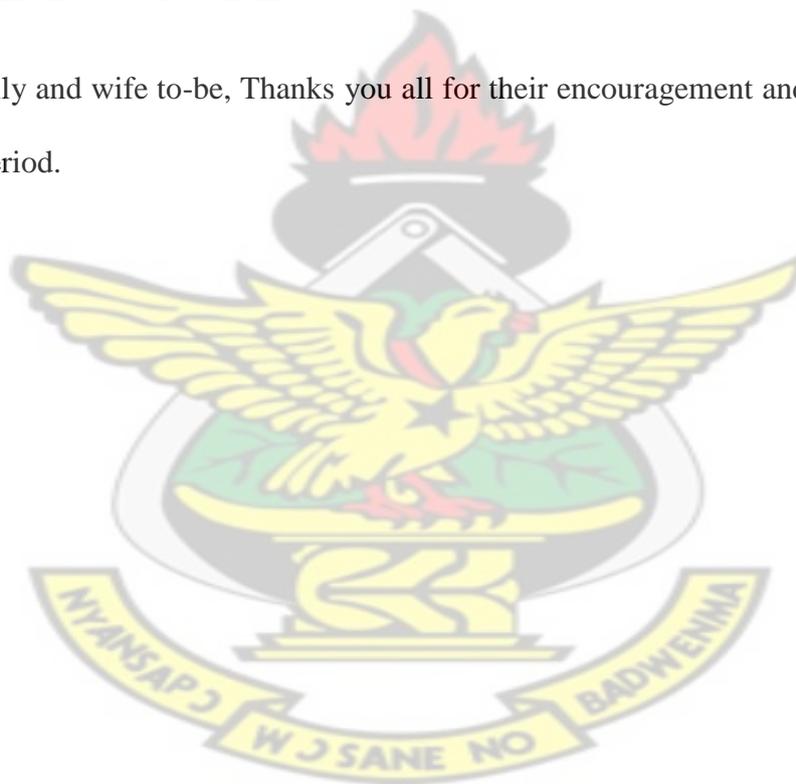
## ACKNOWLEDGEMENT

To God almighty for the His Care, Love, Strength and Wisdom He gave me throughout the period of my work.

I am honoured to work with my supervisor, Mr. J. K. Panford, Dr. J. B. Hayfron-Aquah and a colleague friend Ernest; it is all a sincere appreciation to them for their support, patient and guidance during my master thesis work.

Again to all my friends and loved once who gave me encouraged in the course of my work, I say a big shout out to them all. God bless them.

Finally, to my family and wife to-be, Thanks you all for their encouragement and invaluable support during this work period.



## DEDICATION

This work is dedicated to the all Mighty God and my Son (**Kofi Annan Otoo**)

# KNUST

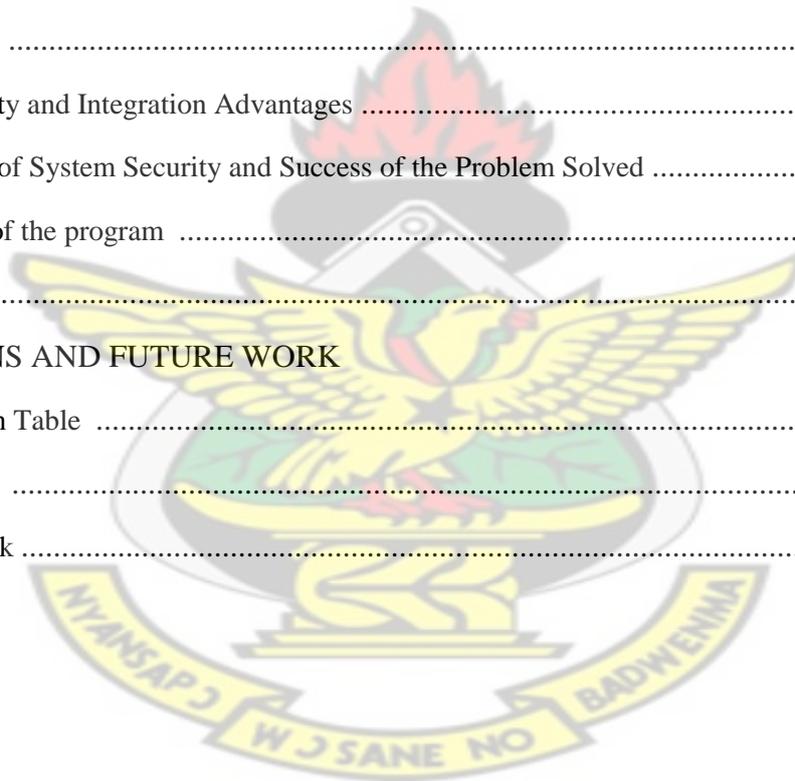


# Contents

Abstract	
Acknowledgements	
Dedication	
Table of Content.....	a
List of Figures.....	d
List of Tables .....	e
<b>1.0 BACKGROUD OF CLOUD COMPUTING</b>	
1.1. Introduction .....	1
1.2 Background .....	2
1.3. Problem Statement .....	8
1.4. Purpose of Study .....	11
1.5. Research Methodology .....	11
1.6. Scope and Limitations of Research .....	11
1.7. Thesis Organization .....	12
<b>2.0. LITERATURE REVIEW</b>	
2.1. Introduction .....	13
2.2. Related Work .....	14
2.2.1. Access control Method (Two-Factor Authentication) .....	14
2.2.2. A Proposed Model for Data Storage Security in Cloud computing Using Kerberos Authentication Service .....	15
2.2.3. T-CLOUD: A Multi – Factor Access Control Framework for Cloud Computing.....	23
2.2.4. Secure Substantiation in Cloud Computing Environment .....	28
2.2.5. Authentication in the Clouds: A Framework and its Application to Mobile Users.....	32
2.3. Summary .....	39
<b>3.0. RESEARCH DESIGN AND METHODOLOGY</b>	
3.1. Introduction .....	40
3.2. Kerberos Authentication Service .....	40
3.2.1 What is Kerberos .....	40

3.2.2 Kerberos Authentication Overview .....	40
3.2.3 Limitation of Kerberos .....	42
3.2.4 Kerberos Process .....	43
3.3. One Time Passwords .....	43
3.4. Two-Factor Authentication with OTP .....	45
3.5. Problem to be Addressed .....	45
3.6. Design of System .....	46
3.6.1. Authentication Process .....	46
3.6.2. Problem and Purposes Overview .....	46
3.7. Research Questions .....	47
3.8. Research Hypotheses .....	47
3.9. Existing Model .....	48
3.10. Proposed Model .....	48
3.11. Tools Needed To Be Used .....	49
3.12. Testing Of the Designed Algorithm .....	49
4.0. DESIGN OF THE MODEL	
4.1. Introduction .....	50
4.2. Pre- Authentication .....	50
4.3. Computation of OTP Hash Value and Storing .....	51
4.4. Time Synchronization .....	51
4.5. OTP via SMS Authentication .....	52
4.6. Model of Existing Kerberos Protocol .....	53
4.7. Proposed Kerberos Two-Factor Authentication System with OTP.....	54
4.8. Algorithms for implementation .....	56
4.8.1 SHA-1 Algorithm .....	56
4.8.2 OTP Algorithm .....	57
4.8.3 Mobile SMS algorithm .....	58
4.8.4 Kerberos Algorithm .....	59
4.9. Testing .....	62

4.9.1 Test Priorities .....	62
4.9.2 Test Environment/ System Specification.....	62
4.10. Simulation Platform .....	63
4.11. Tools for implementation .....	63
4.11.1 Integrated Development Tool (IDE) – Netbeans .....	63
4.11.2 GSM Modem (Mtn 3G Wireless modem) .....	63
4.12. Language .....	63
4.12.1 Java (Web Technology) .....	63
4.12 Summary .....	64
<b>5.0 TEST AND EVALUATION OF SYSTEM</b>	
5.1 Introduction .....	65
5.2. Java Security and Integration Advantages .....	65
5.3. Evaluation of System Security and Success of the Problem Solved .....	66
5.4. Weakness of the program .....	67
5.5. Summary.....	67
<b>6.0 CONCLUSIONS AND FUTURE WORK</b>	
6.1. Comparison Table .....	68
6.2. Conclusion .....	69
6.3. Future Work .....	69



## LIST OF FIGURES

- Figure 1.1 Cloud Service Models
- Figure 2.1 Cloud Data Storage Architecture
- Figure 2.2 High Level Diagram of the Proposed Scheme
- Figure 2.3 Multi – Step Authentication of Cloud User
- Figure 2.4 User Registration
- Figure 2.5 Ticket per session
- Figure 2.6 Hash value computations
- Figure 2.7 Requisition Process
- Figure 2.8 Decryption Process
- Figure 2.9 Authentication Flows
- Figure 2.10 Learning a User Model.
- Figure 2.11 The authentication between the smart phone and integrated authentication server
- Figure 2.12 False positive and false negative
- Figure 2.13 Tradeoffs between false positives and false
- Figure 3.1 Existing Model
- Figure 3.2 Proposed Model
- Figure 4.1 1<sup>ST</sup> Factor Authentication
- Figure 4.2 Hash Value Computations
- Figure 4.3 2<sup>nd</sup> Factor OTP via SMS Authentication
- Figure 4.4 Existing Kerberos Protocol
- Figure 4.5 Proposed Kerberos Protocol
- Figure 4.6 SMS messages from JAVA through HTTP.

## LIST OF TABLES

Table 2.1 Summary of Kerberos Message Exchange in Cloud Service

Table 6.1 Comparison Table

# KNUST



# CHAPTER 1

## 1.0 Background of Cloud Computing

### 1.1 Introduction

Today the use of cloud computing applications is mushrooming at an ever increasing rate. But what exactly is cloud computing? “Cloud computing is a technology that allows the user to access software applications, hardware, storage and computing processes directly from the web.”[1]. The study of 1,300 U.S. and U.K. executives, conducted by Rackspace Hosting finds cloud engagements are delivering positive impacts, from cost savings to more innovation. Interestingly, it also reveals that most of these executives see cloud as laying the groundwork for the next entrepreneurial boom. Sixty-two percent (62%) of respondents either agreed totally or somewhat with the statement that — cloud computing is a key factor in the recent boom of entrepreneurs and start-ups, the survey finds. Twenty-five percent (25%) agreed strongly with this idea. Cloud computing may be a shot in the arm our economy needs. Because it enables entrepreneurs and innovators to start up new ventures with minimal capital requirements — most of what they need is now available as online services, sometimes at no cost [2].

Pondering over unemployment and underemployment in our economy, the availability of cheap cloud computing may be laying the groundwork for a startup boom, the likes that has never seen before. Cloud provides a way to increase capacity or add capabilities on the fly without investing in new infrastructure, training new personnel, or licensing new software. This applies to departments of larger organizations as well — designing new products, without the need to go through corporate finance and IT approvals definitely is a great way to instill entrepreneurial spirit. Cloud computing is the product of the fusion of traditional computing technology and network technology like grid computing, distributed computing parallel computing and so on. Clouds are of particular commercial

interest not only with the growing tendency to outsource IT so as to reduce management overhead and to extend existing, limited IT infrastructures, but even more importantly, they reduce the entrance barrier for new service providers to offer their respective capabilities to a wide market with a minimum of entry costs and infrastructure requirements – in fact, the special capabilities of cloud infrastructures allow providers to experiment with novel service types at the same time reducing the risk of wasting resources. Cloud is not only simple collecting the computer resource, but also provides a management mechanism and can provide services for millions of users simultaneously.

## 1.2 Background of Cloud Computing

Cloud computing involves [distributed computing](#) over a network, where a program or application may run on many connected computers at the same time. It specifically refers to a computing hardware machine or group of computing hardware machines commonly referred as a [server](#) connected through a [communication network](#) such as the [Internet](#), an [intranet](#), a [local area network \(LAN\)](#) or [wide area network \(WAN\)](#). Any individual user who has permission to access the server can use the server's processing power to run an application, store data, or perform any other computing task. Therefore, instead of using a personal computer every-time to run the application, the individual can now run the application from anywhere in the world, as the server provides the processing power to the application and the server is also connected to a network via internet or other connection platforms to be accessed from anywhere [\[1\]](#). All this has become possible due to increasing computer processing power available to humankind with decrease in cost as stated in [Moore's law](#). In common usage, the term "the cloud" is essentially a metaphor for the Internet [\[3\]](#).

Cloud computing relies on sharing of resources to achieve coherence and [economies of scale](#), similar to a utility (like the [electricity grid](#)) over a network [\[4\]](#). At the foundation of cloud computing is the broader concept of [converged infrastructure](#) and [shared services](#). The cloud also focuses on maximizing the effectiveness of the shared resources. Cloud resources are usually not only shared by

multiple users but are also dynamically reallocated per demand. This can work for allocating resources to users. Cloud computing allows companies to avoid upfront infrastructure costs, and focus on projects that differentiate their businesses instead of infrastructure. Also cloud computing allows enterprises to get their applications up and running faster, with improved manageability and less maintenance, and enables IT to more rapidly adjust resources to meet fluctuating and unpredictable business demand [5][6][7].

**Growth and Popularity:** The development of the Internet from being document centric via semantic data towards more and more services was described as "dynamic web". This contribution focused in particular in the need for better meta-data able to describe not only implementation details but also conceptual details of model-based applications.

The present availability of high-capacity networks, low-cost computers and storage devices as well as the widespread adoption of [hardware virtualization](#), [service-oriented architecture](#), and [autonomic](#) and utility computing have led to a growth in cloud computing [8][9][10].

**Characteristics:** Cloud computing exhibits the following key characteristics:

The [National Institute of Standards and Technology](#)'s definition of cloud computing identifies "five essential characteristics":

*On-Demand Self-Service.* A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.

*Broad Network Access.* Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations).

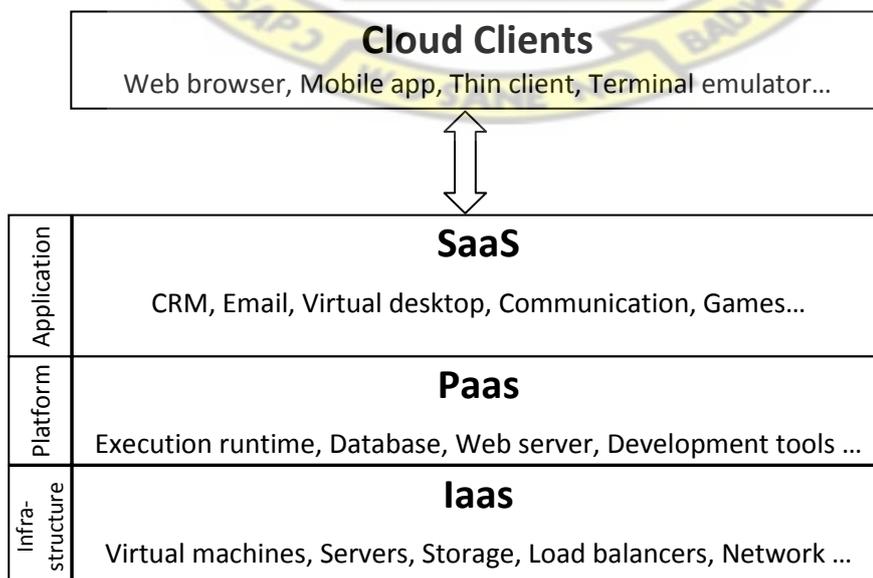
*Resource Pooling.* The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand.

*Rapid Elasticity.* Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear unlimited and can be appropriated in any quantity at any time.

*Measured Service.* Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

—National Institute of Standards and Technology [5].

**Service models:** Cloud computing providers offer their services according to several fundamental models:[5] infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS) where IaaS is the most basic and each higher model abstracts from the details of the lower models.



## Figure 1.1 Cloud Service Models

**Infrastructure as a Service (IaaS):** In the most basic cloud-service model, providers of IaaS offer computers – physical or (more often) virtual machines – and other resources. IaaS clouds often offer additional resources such as a virtual-machine [disk image](#) library, raw (block) and file-based storage, firewalls, load balancers, IP addresses, [virtual local area networks](#) (VLANs), and software bundles [11]. IaaS-cloud providers supply these resources on-demand from their large pools installed in [data centers](#). For [wide-area](#) connectivity, customers can use either the Internet or [carrier clouds](#) (dedicated virtual private networks). Recently, this paradigm has been extended towards *sensing and actuation resources*, [12] aiming at providing virtual sensors and actuators as a services [SAaaS](#).

[Cloud communications](#) and [cloud telephony](#), rather than replacing local computing infrastructure, replace local telecommunications infrastructure with [Voice over IP](#) and other off-site Internet services.

**Platform as a Service (PaaS):** In the PaaS models, cloud providers deliver a [computing platform](#), typically including operating system, programming language execution environment, database, and web server. Application developers can develop and run their software solutions on a cloud platform without the cost and complexity of buying and managing the underlying hardware and software layers. With some PaaS offers like [Microsoft Azure](#), the underlying computer and storage resources scale automatically to match application demand so that the cloud user does not have to allocate resources manually. The latter has also been proposed by an architecture aiming to facilitate real-time in cloud environments [13].

**Software as a Service (SaaS):** In the business model using software as a service (SaaS), users are provided access to application software and databases. Cloud providers manage the infrastructure and platforms that run the applications. SaaS providers generally price applications using a subscription fee.

In the SaaS model, cloud providers install and operate application software in the cloud and cloud users access the software from cloud clients. Cloud users do not manage the cloud infrastructure and platform where the application runs. This process is transparent to the cloud user, who sees only a single access point. To accommodate a large number of cloud users, cloud applications can be multitenant, that is, any machine serves more than one cloud user organization. It is common to refer to special types of cloud-based application software with a similar naming convention: desktop as a service, business process as a service, test environment as a service, communication as a service.

Proponents claim SaaS allows a business the potential to reduce IT operational costs by outsourcing hardware and software maintenance and support to the cloud provider.

**Security as a Service (SECaaS):** Security as a service (SECaaS) is a business model in which a large service provider integrates their security services into a corporate infrastructure on a subscription basis more cost effectively than most individuals or corporations can provide on their own, when total cost of ownership is considered. These security services often include authentication, anti-virus, anti-malware/spyware, intrusion detection, and security event management, among others.

**Types of Cloud [5]** the types of Cloud computing can be classified according to deployment. This deployment can increase or decrease the major cloud computer problems, the security and privacy can be increase or decrease upon the choice of cloud. These classifications are also based on different parameters like, the customer requirement, location of cloud and by their architecture.

There are basically four types of clouds, which are described below-

- **Public Cloud:** This is the one of the cloud in which cloud services are being available to users via a service provider over the Internet. It provides a control mechanism for them. The services may be free or offered on a pay-per usage model.

- **Private Cloud:** This provides many of the benefits of public, but the main difference among two is that the data is managed properly within the organization only, without the limits of network bandwidth.
- **Community Cloud:** This type of cloud is basically managed by group of originations that have a common objective to achieve. The members share access to the data in the cloud.
- **Hybrid Cloud:** This is the combination of public as well as private cloud. It can also be defined as multiple cloud systems that are connected in a way that allows programs and data to be moved easily from one system to another.

KNUST

### 1.3 Problem Statement

Authentication is the verification of the identity of a party who generated some data, and of the integrity of the data. A principal is the party whose identity is verified. The verifier is the party who demands assurance of the principal's identity. Authorization is usually performed after the principal has been authenticated, and may be based on information local to the verifier, or based on authenticated statements by others. Confidentiality is the protection of information from disclosure to those not intended to receive it.

Even though Security, Privacy and Trust issues exists since the evolution of Internet, the reason why they are widely spoken these days is because of the Cloud Computing scenario. Any client/small organization/enterprise that processes data in the cloud is subjected to an inherent level of risk because outsourced services bypass the "physical, logical and personnel controls" of the user.

When accessing data on cloud, one might want to make sure if the data is correctly stored and devoid of another person accessing his or her data on the cloud.

Hence, it is very important for both the cloud provider and the user to have mutual trust such that the cloud provider can be assured that the user is not some malicious hacker and can be assured of data

consistency, data storage and the instance he/she is running is not malicious. Hence the necessity for developing trust models/protocols is demanding.

### **1.3.1 What Needs To Be Done**

Both the user and the cloud provider instance must make sure that whatever requests/response they get is from a trusted source by estimating the authenticity of the person accessing the data or a service on the cloud.

This can be done and will be done by implementing a trust based protocol (Kerberos Two-Factor Authentication via mobile SMS OTP) that runs between the user and the instance before any “real requests/responses” to the user.

The addition of this extra authentication protocol will add another factor to the authentication process. The Kerberos process is more secured as no clear password is transmitted over a network and can be identified by only one person with the second factor device which is very common to every individual.

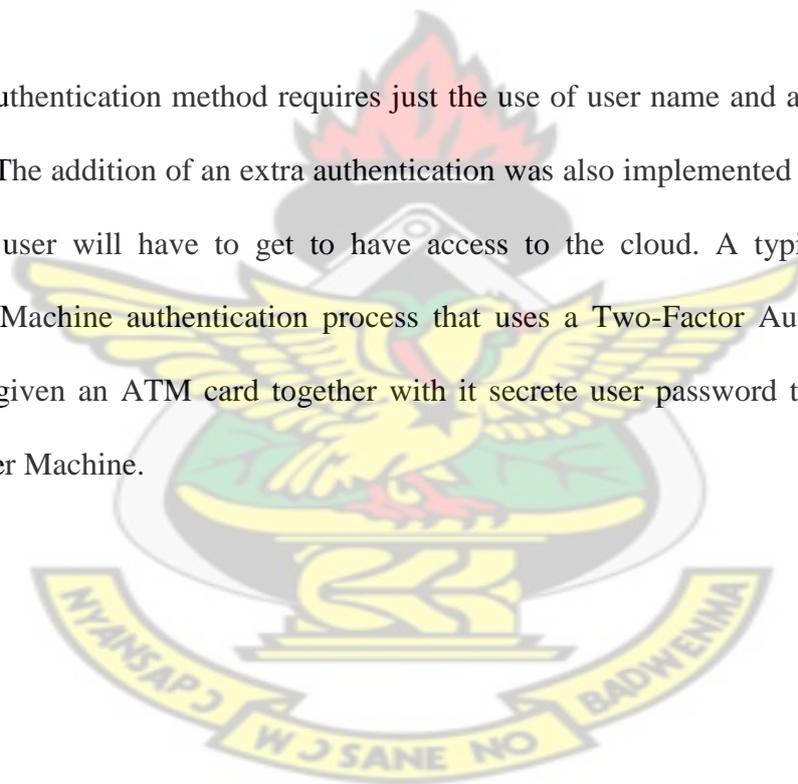
The protocol/model will determine the trust at both the ends by probing each other with challenges and then decide whether the other end is legitimate to handle requests/provide responses.

### **1.3.2 What has been done?**

A common approach to protect user data is that user data is encrypted before it is stored. In a cloud computing environment, a user’s data can also be stored following additional encryption, but if the storage and encryption of a given user’s data is performed by the same service provider, the service provider’s internal staff (e.g., system administrators and authorized staff) can use their decryption keys and internal access privileges to access user data. From the user’s perspective, this could put his stored data at risk of unauthorized disclosure. In which if a user (either employee or anonymous) want to access the data if it belongs to protection then user have to register itself. Now suppose the

user registered itself for accessing data, Organization will provide username and password for authentication. At the same time organization sends the username to cloud provider .Now when user sends request along with username to access the data to cloud provider, the cloud provider first check in which ring requested data belong. If authentication is required, it first checks the username in its own directory for existence, if the username does not exist it ask the user to register itself. If the username matches it redirect the request to company for authentication. Now the user sends password for authentication, and after authentication it redirect the request to cloud provider to access resource .If user-name and password doesn't match then user is not allow to access their account.

Another existing authentication method requires just the use of user name and a password to access data on the cloud. The addition of an extra authentication was also implemented by the use of Token device that every user will have to get to have access to the cloud. A typical example is the Automated Teller Machine authentication process that uses a Two-Factor Authentication method where the user is given an ATM card together with it secrete user password to authenticate for a service on the Teller Machine.



#### **1.4 Purpose of Study**

This paper mainly focused on authentication issues in the cloud computing environment, where an enhancement is made to the intrusion login by the use of OTP via mobile SMS of a person (A Two-

Factor Kerberos OTP Authentication) that cannot be copied and used by other person to identify, authenticate and authorize a user for access to the cloud network.

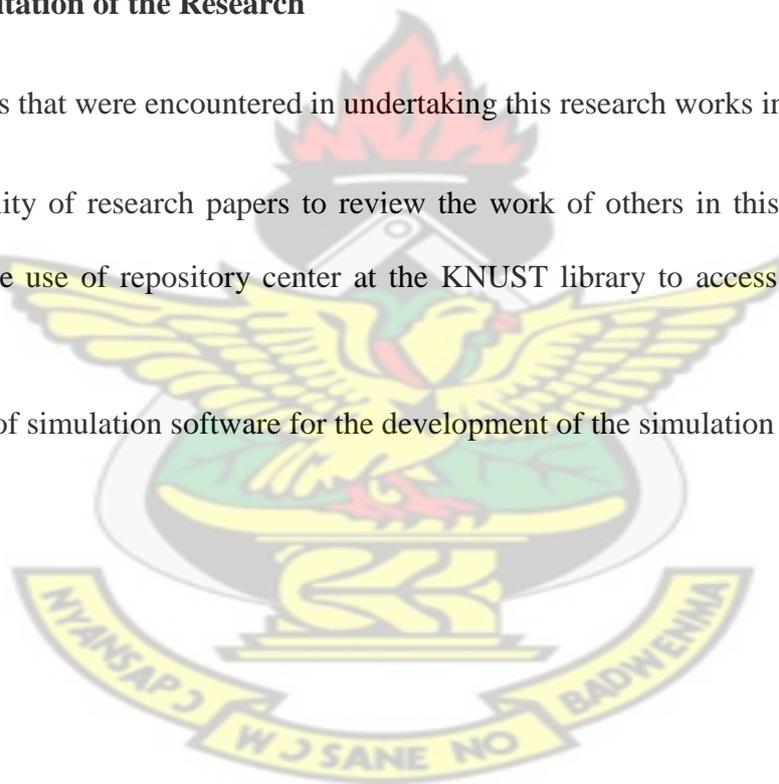
### **1.5 Research Methods (or Research Strategy)**

- Reading on other development areas of the research.
- Developing an algorithm.
- Simulation with simulation software.
- Flow chart and diagrams for demonstration simulation

### **1.6 Scope and Limitation of the Research**

The major challenges that were encountered in undertaking this research works include;

- [i] The availability of research papers to review the work of others in this area: this has been dealt with the use of repository center at the KNUST library to access any research paper online
- [ii] Availability of simulation software for the development of the simulation for this work:



### **1.7 Thesis Organization**

The composition of the paper is as follows:

- Chapter 1 introduces the conceptual framework from which the topic has evolved.
- Chapter 2 related works looks at the back ground of cloud computing authentication security

- Chapter 3 describes user authentication services in cloud computing and problems of them.
- Chapter 4 describes the proposed work for user authentication in cloud computing.
- Chapter 5 Test and Evaluation of System of the model proposed.
- Chapter 6 Conclusions and Future Work

# KNUST

## CHAPTER 2

### 2.0 Literature Review

#### 2.1 Introduction

This chapter reviews literature on relevant issues to provide a theoretical background for the research. The review presents and discusses issues on Cloud computing security purposely on user authentication and multi-factor authentication in the cloud. Other issues that have been captured in this chapter include information on the major works done on Cloud computing security issues.

A whole new range of techniques has been developed to identify people since the 1960s from the measurement and analysis of parts of their bodies to DNA profiles. Forms of identification are used to ensure that citizens are eligible for rights to benefits and to vote without fear of impersonation while private individuals have used seals and signatures for centuries to lay claim to real and personal estate. Generally, the amount of proof of identity that is required to gain access to something is proportionate to the value of what is being sought.

In essence, the growth and development of cloud computing security has been on the drawing board for a very long time and as such some work has already been done with some suggestions on the

possible solutions on security issues and frameworks for cloud computing have been presented and discussed.

## **2.2 Related Works**

### **2.2.1 Research on Access control Method by User Authority using Two-Factor Authentication**

(- Keunwang Lee Chungwoon University and Haeseok Oh, Gachon University, South Korea - 2013)

#### ➤ **Objective of the study**

Their study intended to suggest a method that can protect servers and media information, which requires security. The access control method suggested here uses a way that grants users authority by grade and authenticates users through Two-Factor Authentication method.

#### ➤ **Method of the study**

1. ACAS (Access Control Authentication Server), which is suggested by this study and uses Multi-Factor, consists of AMA (Authentication Management Agent) and FMA (File Management Agent). Though they could additionally apply various factors such as PKI or PIN while authenticating users, they used user ID/Password and OTP only in this study.
2. A user requests authentication to Server using his authentication certificate and Server identifies Access Control List of LogDB with that user's authentication certificate, generates OTP value falling to user's grade and sends it to Server, and the user requests the document falling to his grade while having access to File Server by using the issued OTP value.

#### ➤ **Result of the study**

Access control method using Multi-Factor, suggested by their study uses ID/Password and OTP for user authentication. When they compare the user authentication method with the existing single factor authentication method, it has the strength of excellent security and powerfulness. Their results of comparing expense, safety and speed among ID/Password method, PKI method and the system suggested by their study is really good which will add additional security to user authentication.

### ➤ **Conclusion**

In user authentication and access control system, Access Control Method using Multi-Factor, suggested in their study, is thought to have more excellence than other authentication-based systems, in the aspects of expense, efficiency and processing speed. And, Access Control Method also shows big difference in regard with safety when we compare the case with other cases using only authentication certificate.

Therefore, even if they don't use PKI system, which uses public key structure, they may be able to apply the most effective access control system with just low expense. From the viewpoint of software, there can't exist a perfect security system, and we need to make security environment stronger.

### **2.2.2 A PROPOSED MODEL FOR DATA STORAGE SECURITY IN CLOUD COMPUTING USING KERBEROS AUTHENTICATION SERVICE**

(Yaser Fuad Al-Dubai and Dr. Khamitkar S.D; Swami Ramanand Teerth Marathwada University, India. 2013)

#### **Objective of the study**

The purpose of this paper was to focus on the security management of cloud computing data used in cloud computing via their proposed model for data storage security in cloud computing using Kerberos authentication service.

#### **Method of the study**

- i. The basic approach for cloud computing with Kerberos authentication is as follows: a cloud customer should supply a ticket. A ticket for a cloud service is a series of bits with the attribute that it has been enciphered using the private key for that cloud service. That private key is known only to the cloud service itself and to Kerberos. The cloud service can be confident that any information that exists within the ticket originated from Kerberos. Kerberos will have placed the identity of the cloud customer inside the ticket so the cloud service that receives a ticket has a Kerberos authenticated opinion of the identity of the cloud customer. To help ensure that one customer does not steal and reuse another customer's tickets, the cloud customer accompanies the ticket with an authenticator. (In addition, tickets expire after a specified lifetime, which is usually within a few hours.)
- ii. The cloud customer gets a ticket by sending a message to Kerberos naming the principal identifier of the desired cloud service, the principal identifier of the (alleged) cloud customer and the reference to the current time of day. Anyone can send such a message or intercept its response that response however is usable only to the cloud customer named in the original request because Kerberos seals the response by enciphering it in the private key of that cloud customer. The response contains three parts: the ticket (which itself is further sealed in the private key of the cloud service) a newly minted key for use in this cloud customer, server session, and a timestamp issued by the Kerberos server.
- iii. The cloud customer will be able to unseal this message, obtain the ticket and session key and verify that the timestamp is current (thereby preventing replays of old responses). No other customer without the named cloud customer's private key can correctly decrypt the reply to produce the sealed tickets and corresponding session key.
- iv. Once a cloud customer gets a ticket and sends it to a cloud service and the cloud service has identified the cloud customer further use of the fact of authentication is specific to the protocol of the cloud service. One application maybe use the session key (Kerberos seals a copy in the ticket) for secure end to end encryption, while at the other extreme, another application maybe throw

everything but the source network address away and assume that all further requests coming on the connection from this particular network address are from the same cloud customer.

- v. The authenticator mentioned above is a simple mechanism designed to discourage tries at unauthorized reuse ("replay") of tickets by someone who notices a ticket sending by on the network and makes a copy. The authenticator contain of among other things the cloud customer's principal identifier, network address, and the current time of day all sealed with the key that Kerberos minted for this session. After the cloud service decrypts the ticket it uses the session key found in that ticket to decrypt the authenticator. If the principal ID of the authenticator matches the one in the ticket the network address in the authenticator is the same as the one that sent the packet and the time in the authenticator is within the last few minutes the authenticator is probably not a response and the cloud service accepts the associated ticket. That is because authenticators expire in a short time that all the cloud customers and servers in a Kerberos realm need to have their clocks loosely synchronized.
- vi. If a private key has been compromised another party may successfully pose as the principal until the private key is changed and all tickets previously issued under it expire. If a session key is breakthrough another party may successfully pose as the principal until the previously issued tickets expire.
- vii. One more mechanism rounds out the complete Kerberos process. If a cloud customer uses several cloud services a distinct ticket is needed for each. Not all the cloud services to be used may be known at the beginning of a login session but that is when the user provides the password used as a private key to decrypt tickets. To avoid storing the private key in the workstation memory for the entire duration of the session, at login time the user obtains a single ticket, useful only for a service provided by Kerberos itself, the ticket-granting cloud service. Whenever the cloud customer goes back to Kerberos for an additional service specific ticket, the response is actually enciphered in the session key of the ticket granting cloud service. Thus the private key is needed only for the initial

ticket and the workstation software can immediately destroy its copy of that private key after being used once.

A representation of network architecture for cloud data storage from the Kerberos AS is illustrated in figure 2.1. Seven different network entities can be identified as follows:

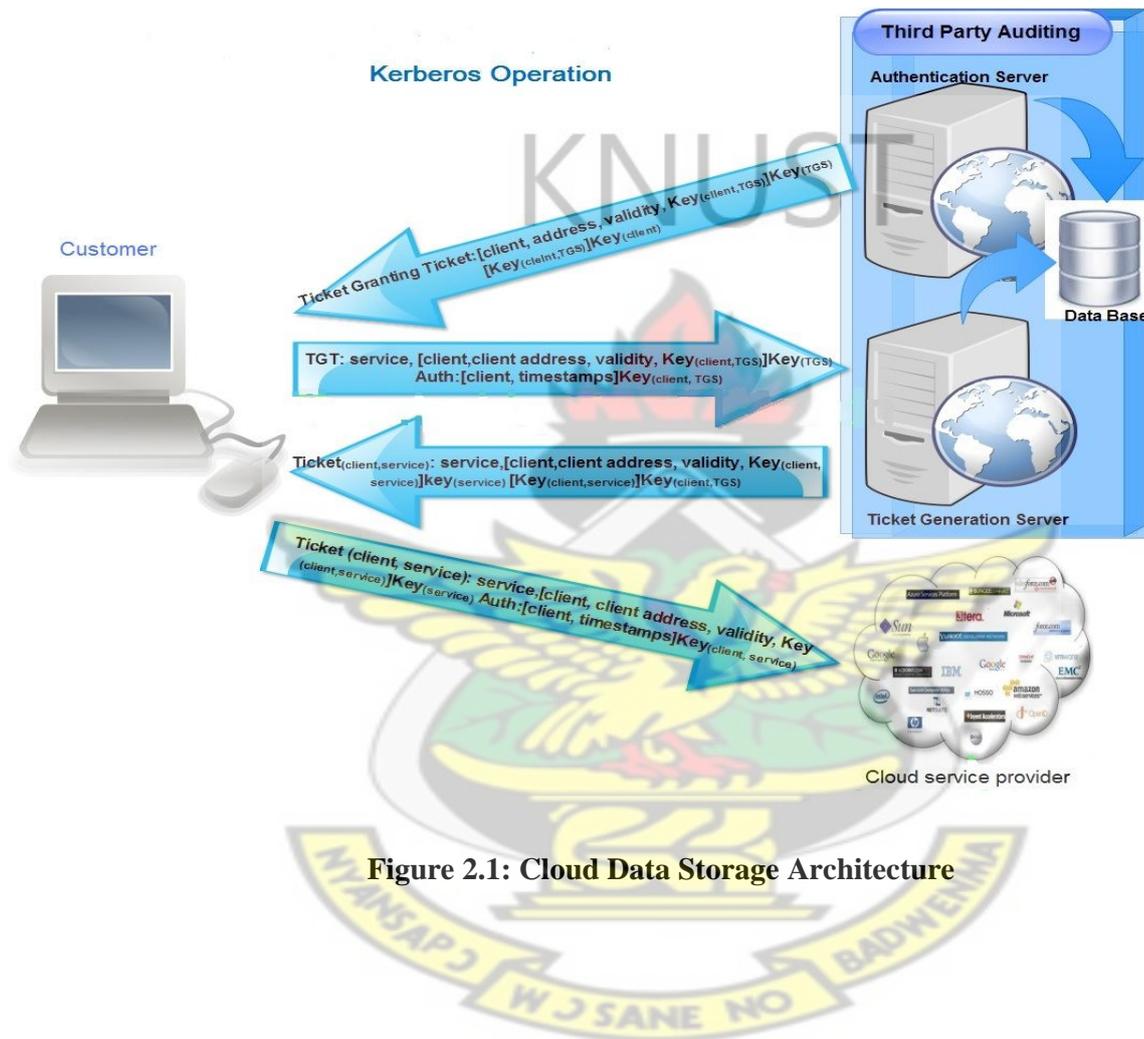


Figure 2.1: Cloud Data Storage Architecture

➤ **Result of the study**

**Authentication Scenario**

The first step of the Key Distribution Centre is the AS. Cloud customer (principal) initially requests a ticket to the KDC by giving it its name, an expiration time until when the authentication will remain valid, the cloud service required (tgs) and some other information, is not mentioned here for clarity.

- i. The KDC if found the cloud customer in its database, replies with two steps:

- ii. Cloud customer ticket contains a session key SA, KDC, the expiration time and it is tgs cloud service name, all encrypted using the secret key of the principal KA. The expiration time usually working day or eight hours, gives a period of time during which the tickets will be valid.
- iii. Granting ticket contains the session key SA, KDC, the expiration time and the name of the cloud customer, all encrypted using the secret key for the KDC KKDC. This is what is known as a TGT. The principal unable to decrypt the TGT, and will be used later to request tickets for the other cloud services. As it is encrypted the cloud customer cannot read the data inside. If tries to modify it, the KDC will not be able to decrypt it and it will be refused.

### **Ticket Granting Cloud Service (TGCS) Scenario**

- i. The second step of the KDC is the distribution of tickets it called the TGCS. Once authenticated the cloud customer who requests a specific application such as telnet or FTP first asks the KDC. It does not query the cloud service directly. This request to the KDC it contains several fields:
- ii. An Authenticator consist of: a timestamp and checksum encrypted with the session key SA,
- iii. KDC, which was obtained earlier in the KDC, shared between the cloud customer and the KDC. This proves the identity of the cloud customer since he is the only one to know this session key. The checksum proves the authentication message has not been modified during the transiting. The timestamp confirms the message is recent, and is used to prevent "reply" attacks, since anyone can interception of data across the network and use it at a later time.
- iv. Typically, the KDC must responds within five minutes for a message to be accepted. This is why it is important to have a good time synchronization across your network where is implemented the Kerberos AS to the cloud computing. Consider the use of Protocol such as NTP (Network Time Protocol) to keep it accurate.
- v. TGT received during the authentication exchange with the KDC. It is used by the KDC to verify the cloud customer's name. If the cloud customer name present in the TGT does not match with related the session key and this means the cloud customer has been impersonated and the KDC is unable to

decrypt the authenticator. Also the KDC verifies the validity by checking the expiration time of the authentication.

- vi. The Cloud Service name to which the cloud customer wants to establish a connection.
- vii. An expiration time for the TGT. The KDC responds to the cloud customer (principal) with two tickets:
- viii. The cloud customer ticket contains a new session key SA, B that the cloud customer and the cloud service will be used to verify each other's identity and to encrypt their sessions. The ticket also encloses the cloud service name and the expiration time of the new ticket. All of these items encrypted using the key SA, KDC shared between the cloud customer and the KDC, known only to the cloud customer.
- ix. The server ticket that contains the same session key SA, B as mentioned above, the cloud customer's name and time of the expiration of the ticket. The server ticket being encrypted with the cloud service's secret key KB, only known to the server.
- x. It is then under the responsibility of the cloud customer to send a server ticket to the cloud service. Therefore, in order for the cloud customer to request access to the cloud service, you must first decrypt the cloud customer ticket and extract the session key SA, B. Once extracted, the cloud customer uses this key to encrypt his authenticator, and consists of a timestamp and. Thus the cloud customer sends this encrypted authenticator and the server ticket to the cloud service. Note that the cloud service does not have the session key SA, B yet. It will get it only if it is able to decrypt the ticket accompanying authenticator, which is the server ticket. It has been sent by the KDC to the cloud customer, encrypted with the cloud service secret key KB, and now are forwarded by the cloud customer to the Cloud Service. As it is encrypted no one except the cloud service is able to see what has this ticket contains, not even the cloud customer. This is how the cloud service receives the session key SA, B to verify the cloud customer's identity and to share with it. It also verifies the validity of the ticket by checking the expiration time enclosed in the server ticket.

- xi. Optionally, the cloud service replies to the cloud customer with a timestamp encrypted with their session key SA, B. This is how the cloud customer verifies and validates the identity of the server; since the cloud customer and the server are the only one to know this session key. Again, the timestamp is used to prove the message is recent, and that it is not previous packet being sent again.
- xii. Table 2.1 shows how they implemented the scenario.

# KNUST

**Table 2.1 Summary of Kerberos Message Exchange in Cloud Service**

<p><b>(A) AS Exchange: to obtain TGT</b></p> <ol style="list-style-type: none"> <li>1. AS_REQ – {cloud customer name, expiration time, tgs cloud service name, ...}</li> <li>2. AS_REP – {SA, KDC, expiration time, tgs cloud service name ...}. KA + {SA, KDC, expiration time, cloud customer name ...}. KKDC.</li> </ol>
<p><b>(B) Ticket Granting Sever Exchange: to obtain Server Granting Ticket</b></p> <ol style="list-style-type: none"> <li>3. TGS_REQ – {timestamp, checksum ...}.SA, KDC + { SA,KDC, expiration time , cloud customer name, ...}. KKDC. + cloud service name + expiration time</li> <li>4. TGS_REP – {SA,B , cloud service name, expiration time, ...}.SA, KDC + {SA, B ,cloud customer name, expiration time,...}. KB</li> </ol>
<p><b>(C) Customer/Server Authentication Exchange: to obtain Cloud Service</b></p> <ol style="list-style-type: none"> <li>5. CS_REQ – {timestamp, checksum ...}.SA,B + {SA,B , cloud customer name, expiration time, ...}. KB</li> <li>6. CS_REP – {timestamp}.SA,B</li> </ol>

➤ **Conclusion**

This paper proposed a model for data storage security in cloud computing using Kerberos; they also presented the problem of data security which effected on cloud data storage, which is essentially a distributed storage system. To ensure the accuracy of customer's data in cloud data storage and accuracy of customers who can access cloud server, they proposed flexible and an effective distributed system with dynamic data support including Kerberos authentication service.

Kerberos provides a centralize Authentication Server whose function is to authenticate customer to cloud server and vice versa. Any customer to be access the cloud server first must make customer ID and password then it can use the cloud server with an increase in qualifying. As known, the unique attribute of the network is security. In an unprotected network environment the customer can be able to apply in any cloud server to service but the process for Kerberos with make use of RSA or DES instead of elaborate protocol can provide the authentication service. In their opinion this model is novel model in era of cloud data storage domain.

### **2.2.3 T-CLOUD: A Multi – Factor Access Control Framework for Cloud Computing**

(Sultan Ullah, Zheng Xuefeng and Zhou Feng, March, 2013)

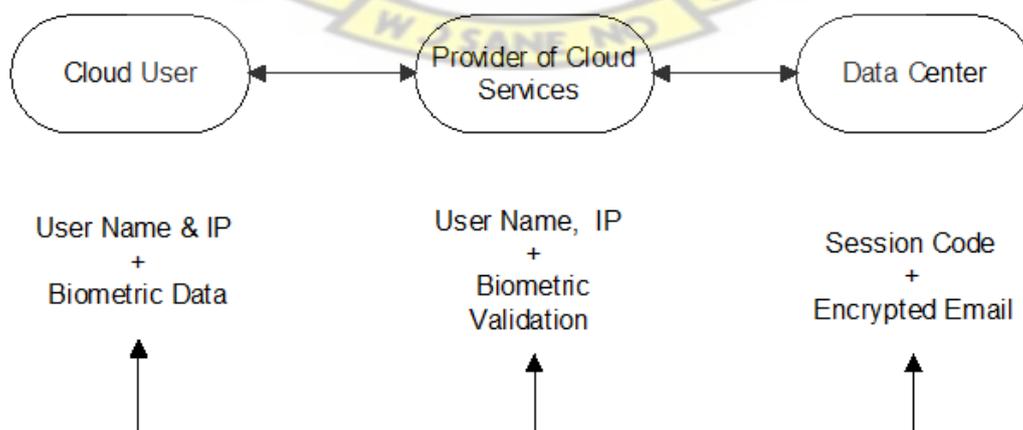
#### **➤ Objective of the study**

This paper the featured of various access control mechanisms are discussed and a novel framework of access control was proposed for cloud computing, which provides a multi - step and multifactor authentication of a user.

#### **➤ Method of the study**

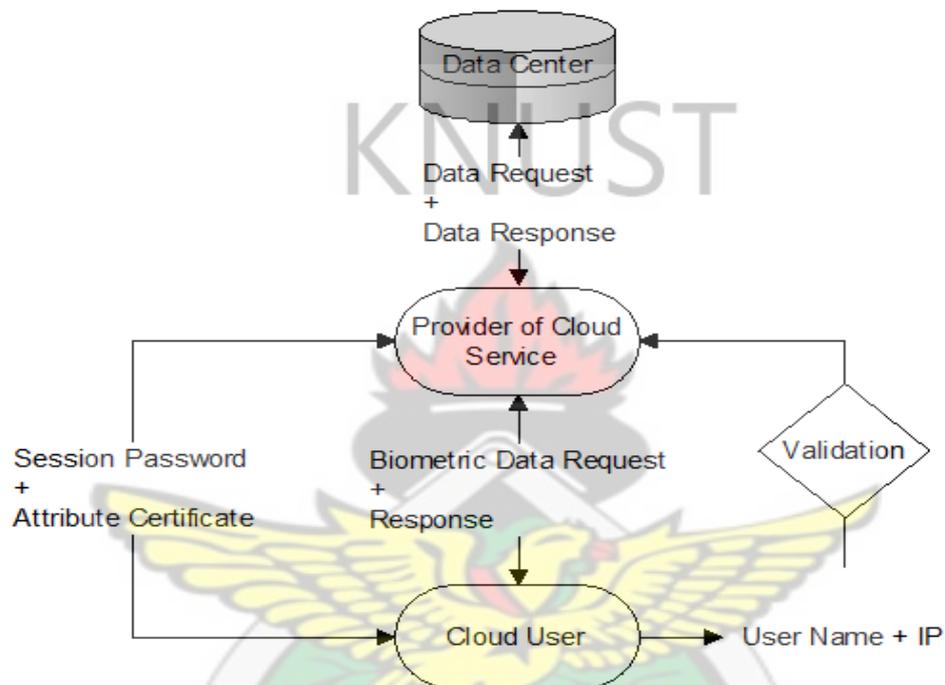
In their proposed model, it was assumed that the components which make the system operational was composed of an owner of the data, a lot of entities which was used was the data created by the owner of the data called the user of the data and the provider of the cloud services and data centre. The authentication of the user was a multistep process, and after the successful authentication the user will only access the data file store by the owner, in a confidential manner by the implementation of

the digital certificate. The owner comes online when it needs to register a new user or make some update to the certificate available on provider of cloud service, and the provider of cloud services was assumed to be online all the time to provide access to the data store at data centre. Another assumption that they made was that the owner of the data will be able to perform / implement binary codes at cloud services for administration of data along with the storing of such data in encrypted type. The idea behind the choice of RSA encryption involving user and the cloud service was, since RSA encryption will need all associations to generate digital certificate for all users, and which is only one of its kind for every user. The user was refraining from accessing other's data files, as the access provided by the owner has put some limitation on the user because of their capabilities. The data which contained request of the user for the data and his identification credentials were sent to cloud services, and checked it with the available information in the validation information at data centre and also the communication between owner and provider of cloud services ought to be secured during the transaction, to oppose any attack. Then the user will be asked to enter the biometric (Finger Prints) data. After completing the login process every user will get an authorization certificate based the attribute. The high level diagram of the proposed framework is shown on Figure 2.2 and the detail diagram is shown in Figure 2.3. The steps required for the registration of new users comprised a request for registration send to the owner of the data with identity details i.e. User ID, IP address / Terminal ID, timestamp and the right to access data file.



**Figure 2.2 High Level Diagram of the Proposed Scheme**

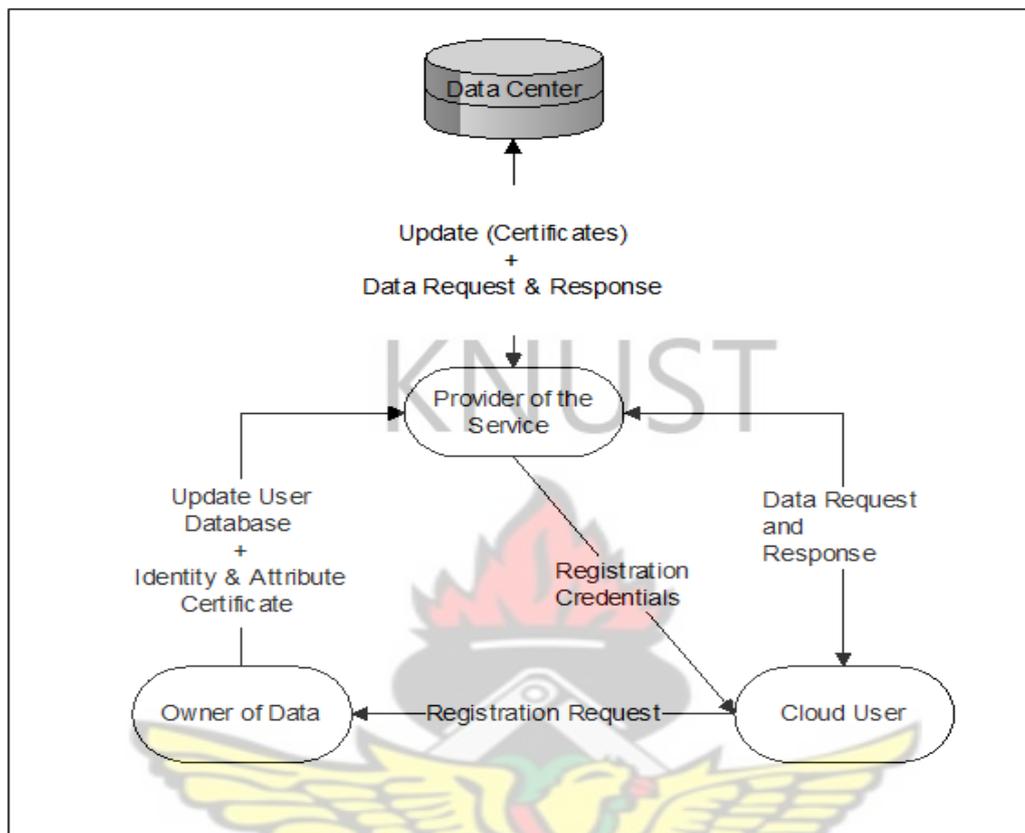
The validity of the request was checked by the owner of the data after the formal receipt of the data. The new user information was now updated by the owner at the service provider. After the updating of information at cloud service and data centre, the data centre now generate sends a reply message to the clients encrypted by MD5. The owner of the data encrypts every file available in the frame of reference using MD5. The data integrity and confidentiality



**Figure 2.3 Multi – Step Authentication of Cloud User**

The owner of the data subsequently propels the whole lot of data, encrypted with his private key, and after that employing the service provider public key in order to maintain the confidentiality and authentication among the provider of the cloud service and owner of the data. After receiving the encrypted files the provider of cloud service will use the public key of the owner of data and its own private key to decrypt the message and send the encrypted files to the data centre for storage. The accomplishment of an improved and protected data access for cloud environment, an exclusive access control was provided by means of another type of digital certificate known as the attribute certificate. The data structure of the attribute certificate was analogous to the identity certificate; on the other hand it does not contain any public key as opposed to identity certificate. If the information

of access control was placed in the extension of the identity certificate then it would have shorten the life span of the certificate, whereas life



**Figure 2.4 User Registration**

Subsequent the successful login of the user on to the cloud services, the certificate issuing authority issues identity, and attribute. The user request for the data access to data centre through cloud services, the cloud service after verification of the attribute and identity certificate with the issuing authority, authorize the user to access data on the data centre. Now the user is able to access the encrypted data as requested by the user. The user needs a decryption gizmo that is present in her/his personal user account. The decryption gizmo will start the generation of the private key for the data encrypted, and then decrypt the data as shown in Figure 2.4.

➤ **Result of the study**

i. **Authentication and Authorization**

All the interactions of the owner of the data and cloud service were also authenticated, the mechanism followed was, the owner uses his private key for the encryption of the scrambled data file, and the Cloud Services uses his public key to authenticate the owner of data. The authentication user of the data was performed with owner's private key when adding a new client, while the owner authentication was performed at cloud service by the private encryption at cloud service with owner private key.

ii. **Data Confidentiality and Integrity**

The provider of cloud service was unable to visualize the original data and digest of the owner as the key was symmetric and only shared among the user and data owner. The data after encryption with symmetric keys was once again encrypted with the private key of the data owner, and public key of the provider of cloud services.

iii. **Access Control Based on Attribute Certificates**

The identity and attribute certificate can be created by owner of the data in certificate issuing authority centre. Because the user needs were different, if one access one data file may not necessary accessed by other client so, creating of access control list for any data object is apparently difficult. In their approach, they used attribute certificate which contain the necessary data structure of the data files for the access control.

➤ **Conclusion**

The model that was proposed in this paper gave power to the owner of the data to implement the security process on the data to be outsourced, and hence retained the control over the data. The model also proposed the combination of cryptography and access control to keep the data safe from vulnerabilities. A multistep, multi – factor authentication approach was employed for the authentication and authorization of the client, which increase the confidentiality and integrity of the data. The paper also presented the private key, hash and public encrypted ciphers among the owner,

the client and the service provider which guarantee the isolation and safe execution of the cloud environment.

#### **2.2.4. Secure Substantiation in Cloud Computing Environment**

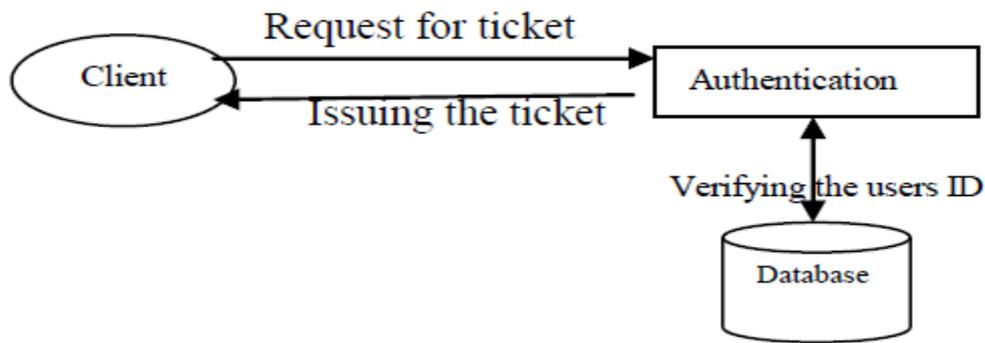
(Raja Shree S. JUNE 2013)

##### ➤ **Objective of the study**

This paper mainly focused on authentication issues in the cloud computing environment, where an enhancement is made to the intrusion login by Kerberos authentication service having fingerprint as its base.

##### ➤ **Method and Results of the study**

Kerberos uses strong encryption and a complex ticket-granting algorithm to authenticate users on a network. Also of interest to many of users, Kerberos has the ability to distribute "session keys" to allow encrypted data streams over an IP network each user for connecting to the cloud at the first should make the profile and user ID. After that it must get the password and also the information of all participating user such as User ID, hashed password will save in the large Database for more secure. All users are register with the Kerberos server. In this method each user want connect to the cloud server at the first time he or she logs on to workstation. Kerberos issues ticket to the client as one ticket per session. Whenever the Client(C) request the ticket to the Authentication Server (AS) with its own identifier (ID<sub>c</sub>) and with the identifier of the ticket granting server(ID<sub>tg</sub>s),the AS responds with a ticket (i.e) encrypted with a key(K<sub>c</sub>) that is derived from users password. When this response arrives, the user decrypts it by using his password. If the correct password is supplied, the ticket is successfully recovered. This is illustrated in Figure 2.5.



**Figure 2.5 Ticket per session**

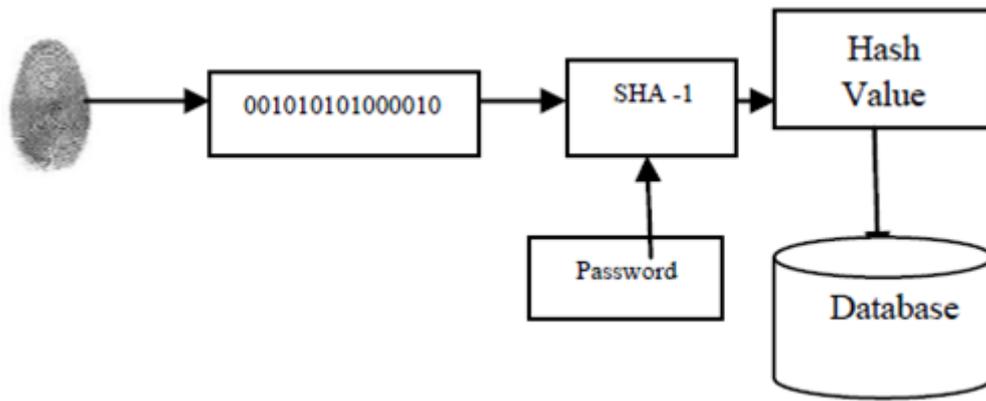
Now this hash value gets registered with the users hash value in AS database. Whenever the user sends request to the AS, the AS verifies the user identity in its data base and then it issues the ticket to the user.

### **Proposed Work / Additions to Kerberos Work**

In Kerberos authentication service the AS stores the hash value of all the users' password. Therefore the hacker may hack the database and retrieves the password. The work being proposed here was, instead of storing the hash value of users password, here we can store the hash value of the particular users finger print along with his password hash value. The ticket gets encrypted with the hash value of the finger print and password. So the user can decrypt the data only by giving his finger print and password. Even if the hacker hacks the hash value then there is no use of that hash value. The session ticket gets decrypted only with the particular person's finger print and password.

### **Computation of Hash Value and storing**

As illustrated in figure 2.6, a fingerprint scanner scans the finger print of the user and converts it into binary form using any modern technique. Take this binary form as input and compute the hash value by using SHA-1 algorithm.



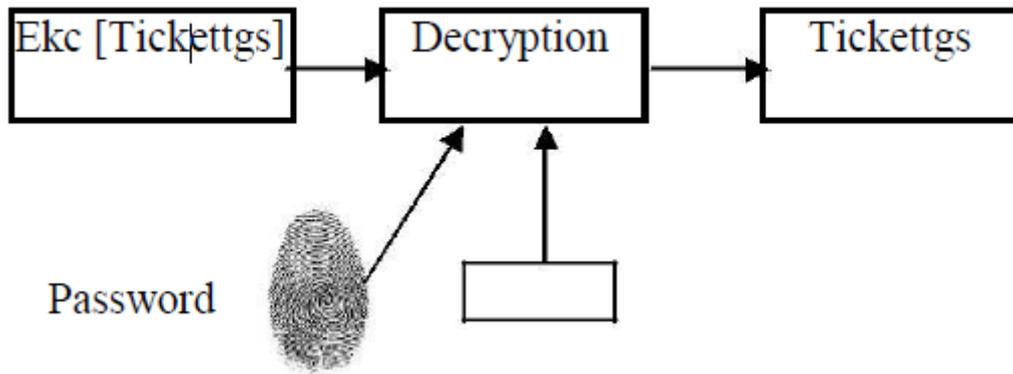
**Figure 2.6 Hash value computations**

**Authentication Process (Figure 2.7 and 2.8)** If the user wants to access the cloud service, the user first request a session ticket to the AS by giving his ID ( $ID_c$ ) and the ticket granting server ID ( $ID_{tgs}$ ). The AS checks the user ID in its database and issues the ticket to the user which is get encrypted with the hash value of the user. The user can able to use the ticket only by decrypting it using his finger print and password. If both matches the ticket gets decrypted .By using this ticket the user can enjoy the services of cloud.  $ID_c || ID_{tgs}$



**Figure 2.7 Requisition Process**

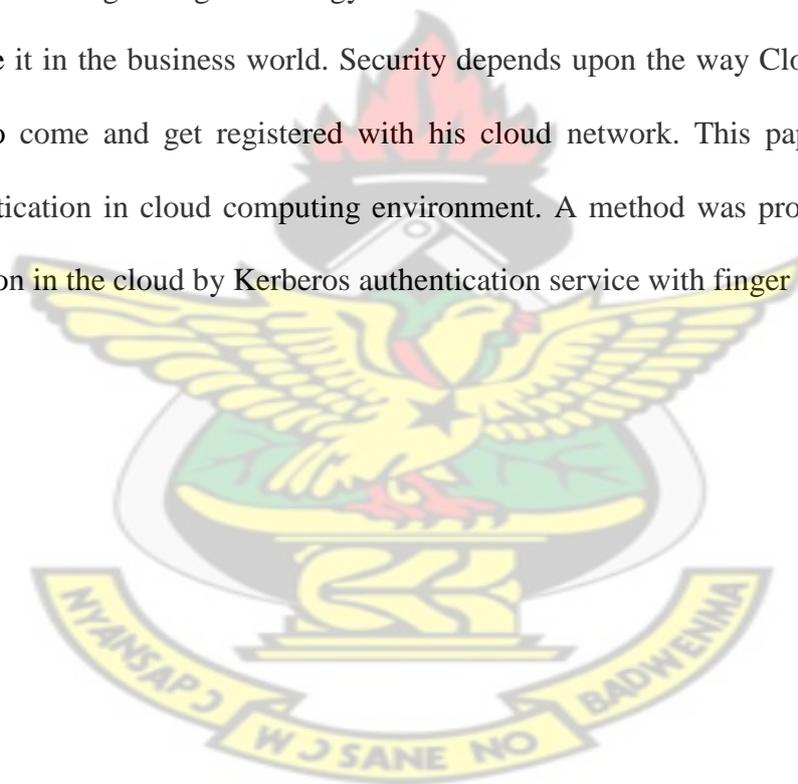
Here the AS stores all its user passwords hash value in a database. The hacker may hack the database and able to retrieve the password from the hash value. Then the hacker may enjoy the service of the ticket.



**Figure 2.8 Decryption Process**

➤ **Conclusion**

Cloud Computing is a fast growing technology in the modern world but it needs some more security features to enhance it in the business world. Security depends upon the way Cloud service provider allows its client to come and get registered with his cloud network. This paper investigated the problem of authentication in cloud computing environment. A method was proposed to ensure the secure authentication in the cloud by Kerberos authentication service with finger print as its base.



**2.2.5 Authentication in the Clouds: A Framework and its Application to Mobile Users**

( R. Chow, M. Jakobsson, R. Masuoka, J. Molina, Y. Niu, E. Shieshi and Z. Song., October, 2010)

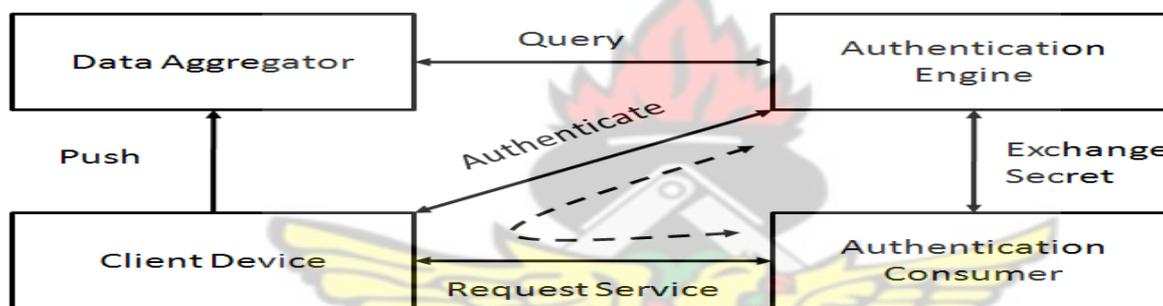
➤ **Objective of the study**

They describe how cloud computing can address a typical handsets that have input constraints and practical computational and power limitations, which must be respected by mobile security technologies in order to be effective.

Their approach was based on a flexible framework for supporting authentication decisions they called TrustCube (to manage the authentication infrastructure) and on a behavioural authentication approach referred to as implicit authentication (to translate user behaviour into authentication scores).

➤ **Method of the study**

**Authentication Flows**



**Figure 2.9 Authentication Flows**

In figure 2.9, Client Device pushes context and activity data to a Data Aggregator; the Authentication Engine queries the Data Aggregator for individual device reports; the Client Device requests a service from the Authentication Consumer; the Client Device authenticates itself through the Authentication Engine; and the Authentication Engine exchanges a secret with Authentication Consumer during authentication (in order to later verify authentication results).

They considered architecture with the following types of participants: client devices, data aggregators, an authentication engine, and authentication consumers.

The authentication engine obtains data from data aggregators, and may request data directly from client devices. It makes authentication decisions based on collected data and authentication policies.

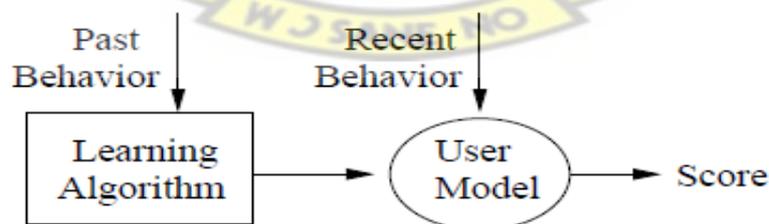
Authentication consumers provide policies to the authentication engine based on end user access requests (e.g., a webpage access request or a payment request). Finally, the authentication consumer responds to a client's request based on the authentication result it receives. The authentication flow is as follows: Before authentication starts, the authentication consumer lists the access requests (e.g., a webpage access request or a payment request) that require authentication. For each request, the authentication consumer will register a policy with the authentication engine. During normal operation, client devices periodically report to the data aggregator. The authentication flow starts when an access request is received by the authentication consumer. Upon receiving the request, the authentication consumer redirects the request to the authentication engine, along with request details.

### Data Analysis and Processing

Both pull and push methods were adopted by client devices to provide data. The push path is from client devices to data aggregators. The main purpose is to constantly report the context and behavior of client devices. The pull path is a request from the authentication engine to client devices and data aggregators to send data back to the authentication engine.

Data aggregators might mine the data received to derive a data and behavioral model for the client device.

From a user's past behavior, they first learnt a user model which characterizes an individual's behavioral patterns. They use this probability as an authentication score.

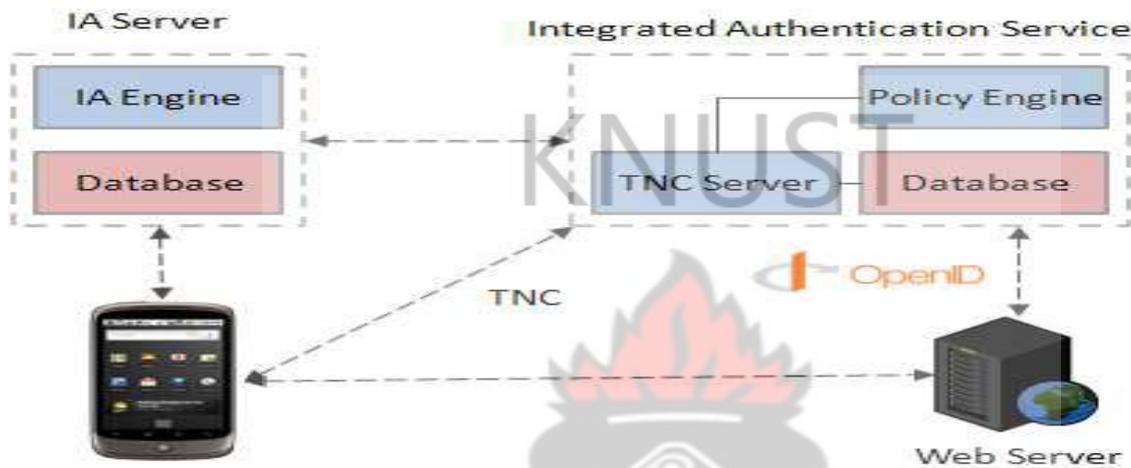


**Figure 2.10: Learning a User Model.**

In figure 2.10 the learning algorithm uses past behavior to create a user model. The user model uses recent behavior to generate a score.

## Implementation Approach

They implemented the authentication framework described above. They called it TrustCube or Trust3 because, unlike traditional user-based authentication, TrustCube supports a wide range of policies, which may include reports on the user, the platform, and the environment of a client device (3 factors). The general architecture is given in Figure 2.11.



**Figure 2.11: The TrustCube architecture**

One implementation of the authentication framework of Figure 2.11, the authentication between the smart phone and integrated authentication server uses the TNC (Trusted Network Connect) protocol. Service requests are redirected to the integrated authentication service using the OpenID protocol.

The client side agent collects two kinds of data; i.e.

First, it collects a user's context and activities, and reports these to the data aggregator regularly (in our system, the data aggregator is the implicit authentication server, or IA server). The collected data consists of phone call and SMS history, browser history, network information, and location. The data is stored locally until it is successfully reported to the data aggregator. This key is device-specific, generated and stored on the device, and never exported. The system cannot infer the actual data from the hashed data, nor can it test a piece of data to see if it agrees with the original data.

Second, during authentication, it collects information about the phone and reports this data to the authentication engine (in our system, the authentication engine is the integrated authentication server). The information it collects is based on the policy provided by the authentication engine, and may include, for example, the applications that are running and installed and the firmware version.

The IA server exposes two web service interfaces: report and query. The report interface allows client side agents to report context and activity information routinely; the query interface allows other entities (e.g., the authentication engine) to get a score for a device which indicates how normal the behavior of the device is at the moment.

The integrated authentication service is the authentication engine. The service exposes two interfaces: a web-based user interface for authentication consumers to define and maintain policies and a web service interface to authenticate client devices. The authentication service scales effortlessly because it itself is a cloud service.

Policies can be easily uploaded, modified, and monitored using the integrated authentication service's user interface. The integrity check rule includes items such as the minimum OS version, acceptable network settings, and a white/black list of installed/running applications.

Finally, the authentication consumer is a web server. They deployed a sample web server that hosts some sensitive medical data. In order to access the site, a user must authenticate using the integrated authentication service.

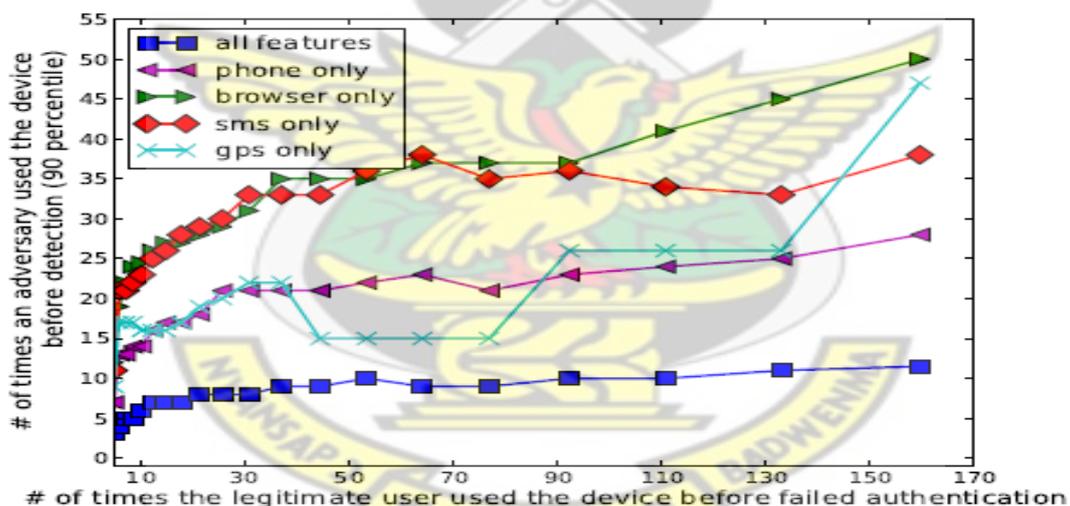
The protocol to redirect an access request between a web server and the integrated authentication service is OpenID, an international standard for federated authentication allowing one ID provider to serve multiple ID consumers. They observe that this protocol supports the separation of the authentication component from a web-based service and it fits our general architecture well. The

OpenID protocol also included methods for a web server and an integrated authentication service to exchange a secret before authentication using the Diffie-Hellman key exchange method. Later, the web server may use the shared secret to verify authentication results. Since information is transferred through open networks in TrustCube, we adopted the TNC (Trusted Network Connect) [11] protocol for trusted reporting. The TNC client is implemented in the client side agent and the integrated authentication service acts as a TNC server.

➤ **Result of the study**

**Case Study: Device Theft**

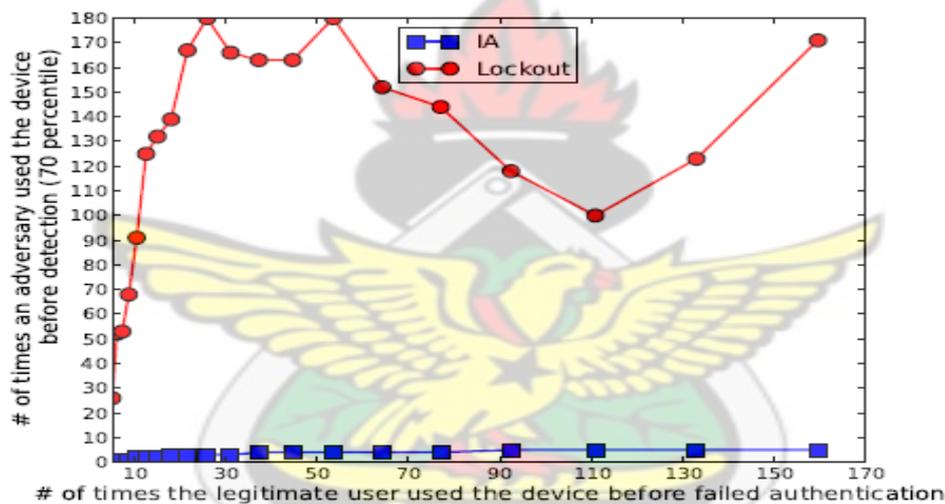
For example, one may expect the detection rates to be much greater in a context where we wish to detect the theft of credit cards than one where we aim to detect the theft of a handset.



**Figure 2.12: False positive and false negative rates**

Figure 2.12 shows x-axis shows the number of successful implicit authentications before a failed authentication for a legitimate user, while the y-axis shows the number of times an adversary can access resources before being locked out with 90% probability.

We emphasize that the ability to detect this sort of theft has to be balanced against the risk of failing legitimate users – the usual balancing act between false positives and false negatives. They then created adversarial attempts, and attempted to detect these. To model an adversarial attempt, they pasted in the activity trace of one user with the activity trace of another user, with care taken to avoid peculiarities such as sudden jumps in terms of location. By varying our acceptance thresholds, they obtain tradeoffs between the number of times between failed legitimate authentication attempts and failed adversarial authentication attempts. This was shown in Figure 4. They compared these rates to those obtained from a simple timed lock-out policy in Figure 5.



**Figure 2.13: Tradeoffs between false positives and false negatives**

Figure 2.13 shows x-axis shows the number of successful implicit authentications before a failed authentication for a legitimate user, while the y-axis shows the number of times an adversary can access resources before being locked out with 70% probability.

They noted that these are preliminary results obtained from an experiment with approximately 50 users, observed over a period of two weeks. They expected that the error rates will be better for large-scale deployments that collect behavioural data for extended periods of time.

## ➤ Conclusion

Cloud computing has brought new challenges and opportunities for authentication. There is increasing demand for usable authentication to access services and data for both enterprises and consumers. When mobile users access the cloud, their behavioural data is starting to be used for applications such as advertising, using the cloud's data-aggregation ability. This system has at its core an authentication service – dubbed TrustCube – that itself resides in the cloud. Any cloud-based service can re-direct authentication to the authentication service via a federated authentication framework such as OpenID.

Their system has the ability to accept various authentication methods in a policy-driven manner, from TCG-style device integrity measurements to passwords. The system is flexible enough to support newer, cloud-oriented authentication techniques. In particular, they have integrated the system with implicit authentication, and they have described several simple end-to-end use cases with our authentication frame work and implicit authentication.

### 2.3. Summary

Some of the works used various techniques such as Multi-factor authentication (called TrustCube), Kerberos Authentication and Encryption of user keys before transmission of data. In this research work, a Two-Factor Authentication approach will be used and the result will be dealt with using a software simulator in order to justify the validity of the algorithm.

## CHAPTER 3

### 3.0 Research Design and Methodology

#### 3.1 Introduction

This chapter of the research discusses the methods used to accomplish the research work. It presents the manner in which the study developed and progressed to answer the aims and objectives of the research in order to draw conclusion. This chapter will also present the most important external factors affecting the authentication process where an enhancement is made to the intrusion login by Kerberos Authentication Service having One-Time Password as its base.

## **3.2 Kerberos Authentication Service**

**3.2.1 What is Kerberos?** Kerberos is a distributed authentication service that allows a process (a client) running on behalf of a principal (a user) to prove its identity to a verifier (an application server, or just server) without sending data across the network that might allow an attacker or the verifier to subsequently impersonate the principal. Kerberos optionally provides integrity and confidentiality for data sent between the client and server. Kerberos was developed in the mid-'80s as part of MIT's Project Athena. As use of Kerberos spread to other environments, changes were needed to support new policies and patterns of use. To address these needs, design of Version 5 of Kerberos (V5) began in 1989. Though V4 still runs at many sites, V5 is considered to be standard Kerberos.

### **3.2.2 Kerberos Authentication Overview**

The Kerberos Authentication System uses a series of encrypted messages to prove to a verifier that a client is running on behalf of a particular user. The Kerberos protocol is based in part on the Needham and Schroeder authentication protocol, but with changes to support the needs of the environment for which it was developed. Among these changes are the use of timestamps to reduce the number of messages needed for basic authentication [3.6], the addition of a "ticket-granting" service to support subsequent authentication without re-entry of a principal's password, and different approach to cross-realm authentication (authentication of a principal registered with a different authentication server than the verifier).

The remainder of this section describes the Kerberos protocol. The description is simplified for clarity; additional fields are present in the actual protocol. Readers should consult RFC 1510 for a

more thorough description of the Kerberos protocol. Though conceptually, Kerberos authentication proves that a client is running on behalf of a particular user, a more precise statement is that the client has knowledge of an encryption key that is known by only the user and the authentication server. In Kerberos, the user's encryption key is derived from and should be thought of as a password; we will refer to it as such in this article. Similarly, each application server shares an encryption key with the authentication server; we will call this key the server key. Encryption in the present implementation of Kerberos uses the data encryption standard (DES). It is a property of DES that if cipher text (encrypted data) is decrypted with the same key used to encrypt it; the plaintext (original data) appears. If different encryption keys are used for encryption and decryption, or if the cipher text is modified, the result will be unintelligible, and the checksum in the Kerberos message will not match the data. This combination of encryption and the checksum provides integrity and confidentiality for encrypted Kerberos messages.

The client and server do not initially share an encryption key. Whenever a client authenticates itself to a new verifier it relies on the authentication server to generate a new encryption key and distribute it securely to both parties. This new encryption key is called a *session key* and the Kerberos ticket is used to distribute it to the verifier. The Kerberos ticket is a certificate issued by an authentication server encrypted using the server key. Among other information, the ticket contains the random session key that will be used for authentication of the principal to the verifier, the name of the principal to whom the session key was issued, and an expiration time after which the session key is no longer valid. The ticket is not sent directly to the verifier, but is instead sent to the client who forwards it to the verifier as part of the application request. Because the ticket is encrypted in the server key, known only by the authentication server and intended verifier, it is not possible for the client to modify the ticket without detection

### **3.2.3 Limitation of Kerberos**

Limitations of Kerberos have been described in the literature. Though most are a matter of preference; or apply to V4 and early drafts of V5, a few are fundamental and are discussed here.

In particular, Kerberos is not effective against password guessing attacks; if a user chooses a poor password, then an attacker guessing that password can impersonate the user.

Similarly, Kerberos requires a trusted path through which passwords are entered. If the user enters a password to a program that has already been modified by an attacker (a Trojan horse), or if the path between the user and the initial authentication program can be monitored, then an attacker may obtain sufficient information to impersonate the user.

Kerberos assumes that each user is trusted but is using an untrusted host on an untrusted network. Its primary goal is to prevent unencrypted passwords from being transmitted across that network. However, if anyone other than the proper user has access to the one host that issues tickets used for authentication — called the Key distribution center (KDC) -- the entire Kerberos authentication system is at risk.

Kerberos can be combined with other techniques, as described later, to address these limitations. To be useful, Kerberos must be integrated with other parts of the system. It does not protect all messages sent between two computers; it only protects the messages from software that has been written or modified to use it. While it may be used to exchange encryption keys when establishing link encryption and network level security services, this would require changes to the network software of the hosts involved. Kerberos does not itself provide authorization, but V5 Kerberos passes authorization information generated by other services. In this manner, Kerberos can be used as a base for building separate distributed authorization services.

### **3.2.4 Kerberos Process**

Kerberos uses strong encryption and a complex ticket-granting algorithm to authenticate users on a network. Also of interest to many of users, Kerberos has the ability to distribute "session keys" to allow encrypted data streams over an IP network each user for connecting to the cloud at the first

should make the profile and user ID. After that it must get the password and also the information of all participating user such as User ID, hashed password will save in the large Data Base for more secure. All users are register with the Kerberos server. In this method each user want connect to the cloud server at the first time he or she logs on to workstation. Kerberos issues ticket to the client as one ticket per session. Whenever the Client(C) request the ticket to the Authentication Server (AS) with its own identifier (IDc) and with the identifier of the ticket granting server(ID tgs), the AS responds with a ticket (i.e.) encrypted with a key(Kc) that is derived from users password. When this response arrives, the user decrypts it by using his own password. If the correct password is supplied, the ticket is successfully recovered. Now this hash value gets registered with the users hash value in AS database. Whenever the user sends request to the AS, the AS verifies the user identity in its data base and then it issues the ticket to the user.

### 3.3. One Time Passwords

A One-Time Password (OTP) is just what the names implies, a password that is only valid for one login. The benefit of OTPs is that it offers much higher security than static passwords, in expense of user friendliness and configuration issues.

OTPs is immune against password sniffing attacks, if an attacker use software to collect your data traffic, video records you when you type on your keyboard, or use social engineering, it doesn't matter since the password that the attacker gets hold on will not be valid to use. [3.7]

A OTP can be generated using different methods [3.7][3.8], and is often used in conjunction with a device that is synchronized with an authentication server:

- **Time-Based OTPs:** In the time-based method, a device with an internal clock generates a password that is depending on the current time. For example, every minute a new password is generated in the device, and the same password is generated at the authentication server. When the user wants to login to a service or system, the current OTP that is displayed on the device is used. The device can also use the current time as a factor when creating a hashed OTPs, where

the other factors usually is a challenge or a PIN-code (Two-factor authentication). The main advantage of the time-based method is that the password is only valid for a short period of time, before it expires. This can however lead to problems if the authentication server and the OTP-generating device is not properly synchronized.

- **Counter-Synchronized OTPs:** In this method, a counter is synchronized between the authentication server and the device. The principle for when a user wants to login is the same as the time-based method; the user enters the current OTP that is displayed on the device. A new OTP will now be generated that the user can use next time to login, and the counter will advance one step in the device and in the server. The drawback of this method is that time is not considered when generating the password, making the password available for a long period of time, it will only be changed upon login. This will lead to serious problems if an attacker gets hold of the OTP-generating device.
- **Seed-Chain OTPs** In this method, a previous entered OTP is used as a seed to generate a new OTP, building a chain of passwords that all depend on the previous password. Some Linux distributions have the support of local login using this method. The passwords will be printed out on a piece of paper, and the user will have to follow the list in the correct order to be able to log in. However, this approach is not very safe since it removes the function of the OTP.
- **Challenge-Based OTPs:** This kind of OTPs is used together with two-factor authentication. A user has to put a challenge into the generating device (often a PIN code) in order to generate the OTP. This kind of method is often used when users log in to online banks.

### 3.4. Two-Factor Authentication with OTP

Since the problems with static passwords, OTP will be used for a two-factor authentication as the login procedure. One time passwords and two-factor authentication are two separate solutions but are most often used together for a better security solution.

### **3.5. Problem to Be Addressed**

How cloud users experiencing and what are service providers doing to addressing privacy issue and authentication in the cloud environment?

The amount of proof of identity that is required to gain access to something is proportionate to the value of what is being sought. It is estimated that only 4% of online transactions use methods other than simple passwords [39]. Security of systems resources generally follows a three-step process of identification, [authentication](#) and [authorization](#). Today, a high level of trust is as critical to Cloud transactions as it is to traditional face-to-face transactions.

### **3. 6. Design of System**

In Kerberos authentication service the AS stores the hash value of all the users' password. Therefore the hacker may hack the database and retrieves the password. The work being proposed here is, instead of storing the hash value of user password on the database, we will rather use password from OTP through a mobile device (Mobile Phone SMS). The ticket gets encrypted with the hash value of the OTP. So the user can only decrypt it using the hash password from the OTP within the stipulated time frame. So even if the hacker hacks the hash value then there is no use of that hash value. The session ticket gets decrypted only with the particular OTP ones the decryption is done, the OTP is destroyed.

#### **3.6.1. Authentication Process**

If the user wants to access the cloud service ,the user first request a session ticket to the AS by giving his ID(IDc) and the ticket granting server ID(IDtgs).The AS checks the user ID in its database and issues the ticket to the user which is get encrypted with the hash value of the user. The user can be able to use the ticket only by decrypting it using OTP sent to the mobile phone through sms. The user

will then enter the password from the sms into the link. If both matches the ticket gets decrypted. By using this ticket the user can enjoy the services of cloud.

### **3.6.2. Problem and Purposes Overview**

Both the user and the cloud provider instance must make sure that whatever requests/response they get is from a trusted source by estimating the authenticity of the person accessing the data or a service on the cloud. This can be done and will be done by implementing a trust based protocol (Kerberos Authentication Service having One-Time Password as its base) that runs between the user and the instance before any “real requests/responses” to the user.

The addition of this extra authentication protocol will add another factor to the authentication process. The Kerberos Authentication Service is based on a One-Time Password which is more secured and can be identified by the person who receives OTP code.

The protocol/model will determine the trust at both the ends by probing each other with challenges and then decide whether the other end is legitimate to handle requests/provide responses

### **3.7. Research Questions**

This paper focused on authentication issues in the cloud computing environment, where an enhancement is made to the intrusion login by the use of Kerberos Authentication Service based on a One-Time Password through a mobile

### **3.8. Research Hypotheses**

Since one-time-password (OTP) can only be used ones, combining it with Kerberos Authentication Service will create a strong base to authenticate users;

Therefore, the addition of OTP feature to an existing Kerberos Authentication models raises a more secure way of increasing the security in the authentication process making it better and more secure than the existing username and password or any of them alone.

In [verification](#) mode the system performs comparison the entered received OTP code with a specific hash template stored in a database within the specified time in order to verify the individual.

Three steps involved in verifying a user.

- [i] The first step will use a username and a password to indicate which template should be used for comparison.
- [ii] Next step is to confirm with OTP SMS code sent to the mobile device
- [iii] The next step, the OTP code is matched with the reference stored hash code of the OTP.
- [iv] The last step is the testing step.

KNUST

### 3.11. Existing Model

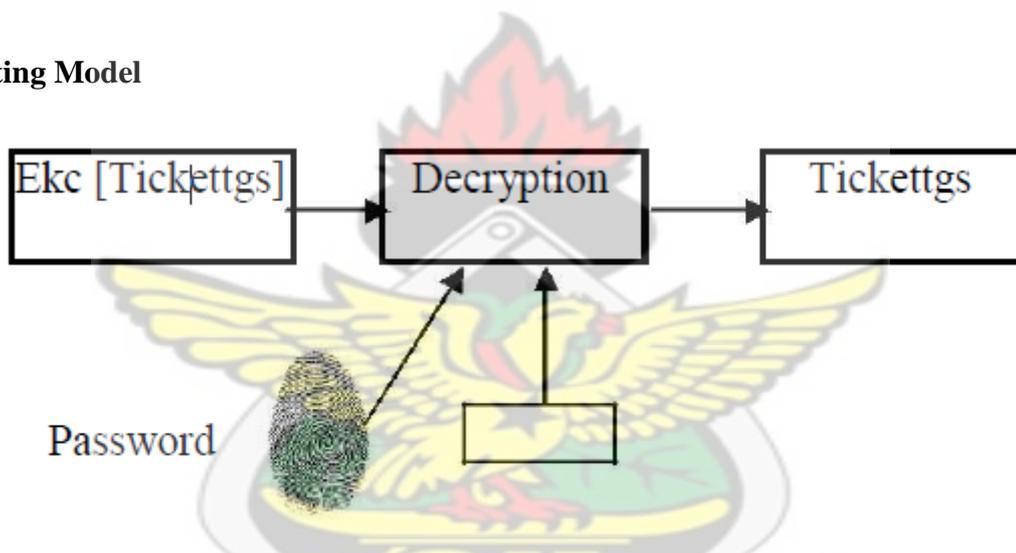
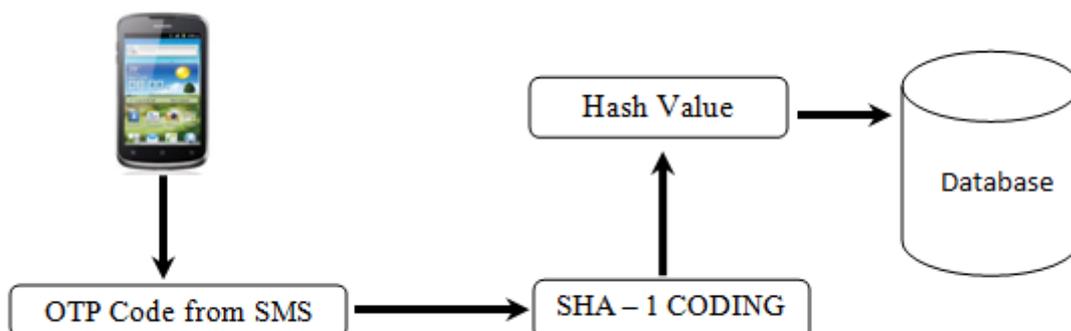


Figure 3.1 Existing Model

Figure 3.1 shows an existing model proposed by [2.5] where a user can decrypt tickets using only his finger print and password.

### 3.12. Proposed Model



### **Figure 3.2 Proposed Model**

Figure 3.2 show the hash encryption of the mobile authentication code for verification with the hash values stored in the database

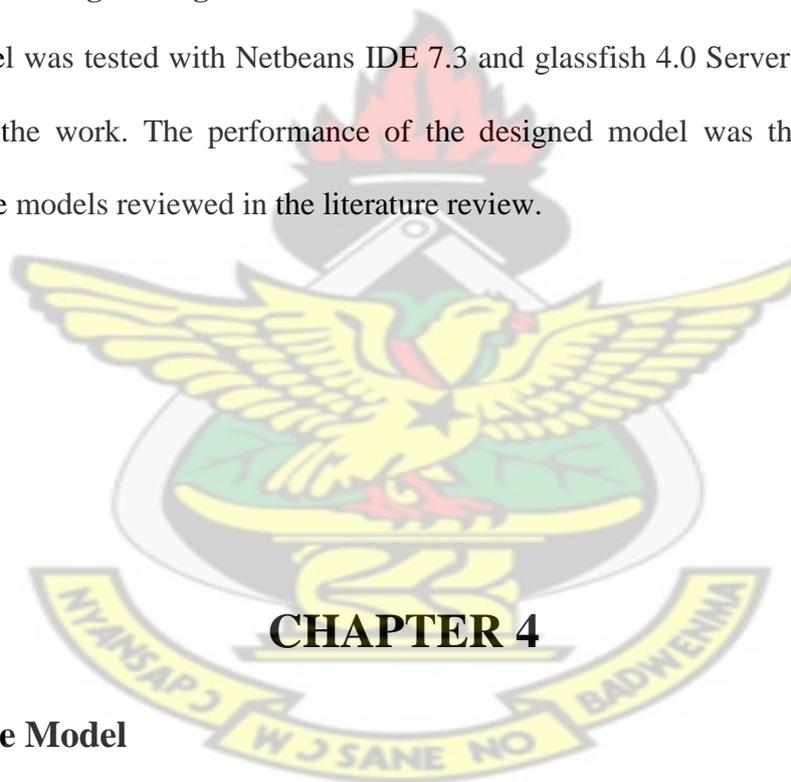
#### **3.11. Tools Needed To Be Used**

Activity diagram, CASE diagram, algorithm, Netbeans IDE 7.3 tools would be used to solve the problem.

#### **3.12. Testing Of the Designed Algorithm**

The designed model was tested with Netbeans IDE 7.3 and glassfish 4.0 Server for developing Java codes to simulate the work. The performance of the designed model was then compared to the performances of the models reviewed in the literature review.

KNUST



## **CHAPTER 4**

### **4.0 Design of the Model**

#### **4.1. Introduction**

The research aimed at adding a second factor authentication to an already existing Kerberos Protocol to strengthen the existing weakness of user authentication in the cloud based protocol.

The work makes use of Kerberos for the first-factor authentication and then makes use of the Kerberos infrastructure to securely achieve the second-factor authentication using an OTP through

Mobile SMS. This chapter deals with how the various methods and tools used in the research are implemented. This section will discuss the implementation and test of the solution.

The proposed Two-Factor Kerberos Authentication consists of four main protocols. These are the Kerberos protocol, SHA-1 encryption algorithm, OTP algorithm and an SMS factor for a second authentication.

#### 4.2. Pre- Authentication

From figure 4.1, if the user wants to access the cloud service, the user first requests a session ticket to the AS by giving his standard Kerberos Username and Password to the AS. AS checks the user ID in its database and issues the ticket to the user which is get encrypted with the hash value of the T-OTP. The user can able to use the ticket only by decrypting it using the Time-OTP password sent via SMS to his Mobile device (Mobile Phone) for the second authentication to be completed. If both matches the ticket gets decrypted. By using this ticket the user can enjoy the services of cloud.

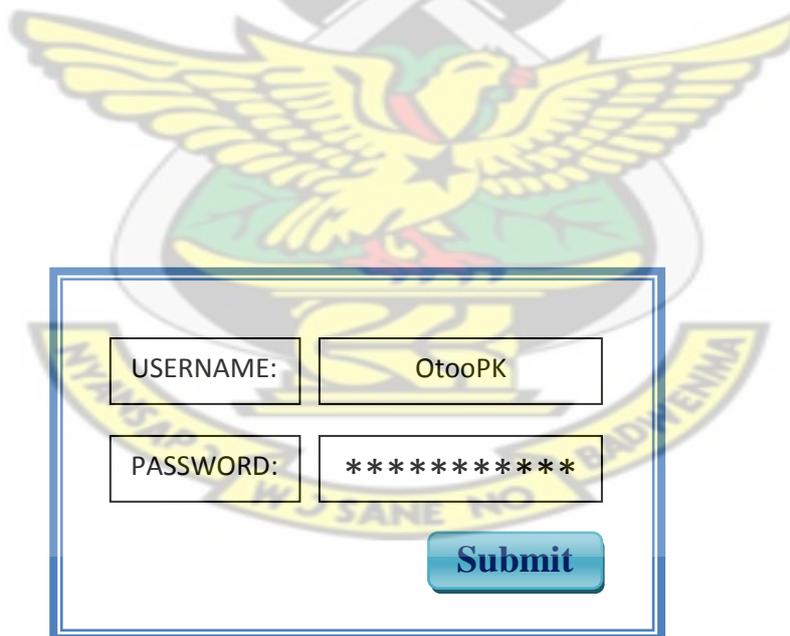


Figure 4.1 1<sup>ST</sup> Factor Authentication

#### 4.3. Computation of OTP Hash Value and Storing

Here (Figure 4.2), an Authentication Server (AS) generate a six digit code using any modern technique and the hash values of the code is stored on the server database and the generated number



is sent to the user via SMS to the mobile device of the user. Take this code form as input and compute the hash value by using **SHA-1 algorithm**.

KNUST

**Figure 4.2 Hash Value Computations**

#### **4.4. Time Synchronization**

It is possible with time- and event-based tokens that the OTP server will lose synchronization with the current time on the network that the sms was sent to. For example, event-based tokens may drift since the counter on the token is incremented every time the token is used, but the counter on the server is only incremented on an authentication. Similarly, the clocks on time-based tokens may drift.

For this reason, the AS timer will remain supreme; i.e. if the AS sent the SMS code, the time range will be allocated to decryption of the ticket sent.

If, within that time frame the ticket is not decrypted, it is destroyed and the user will have to make a new request for a new ticket to be issued, when processing a code fails for this reason (time), then the AS will return a ERROR message. The ERROR will contain a TIME-OUT.

#### 4.5. OTP via SMS Authentication

This is usually the authentication method used when a transaction is verified with an OTP (Figure 4.3). The AS sends you an OTP and you then have a time frame to enter this OTP. This mechanism doesn't need any synchronization process as the OTP is originally generated by the server and send to a third party device. The server expects that you type the correct OTP within generally 5 minutes. If you fail to do it, you just ask a new OTP and then enter it within the given time.

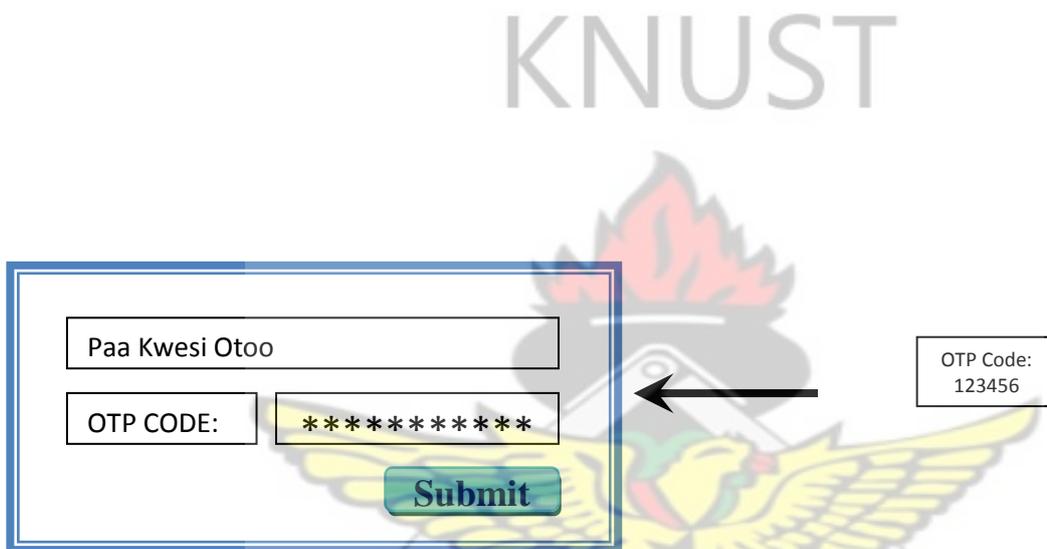


Figure 4.3 2<sup>nd</sup> Factor OTP via SMS Authentication

#### 4.6. Model of Existing Kerberos Protocol

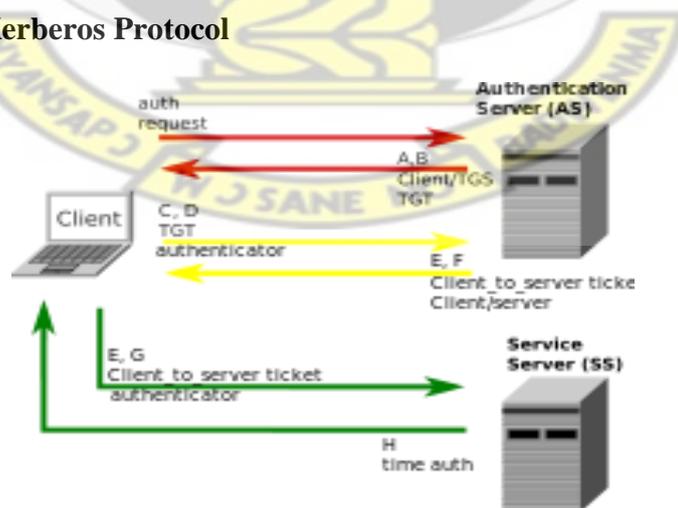
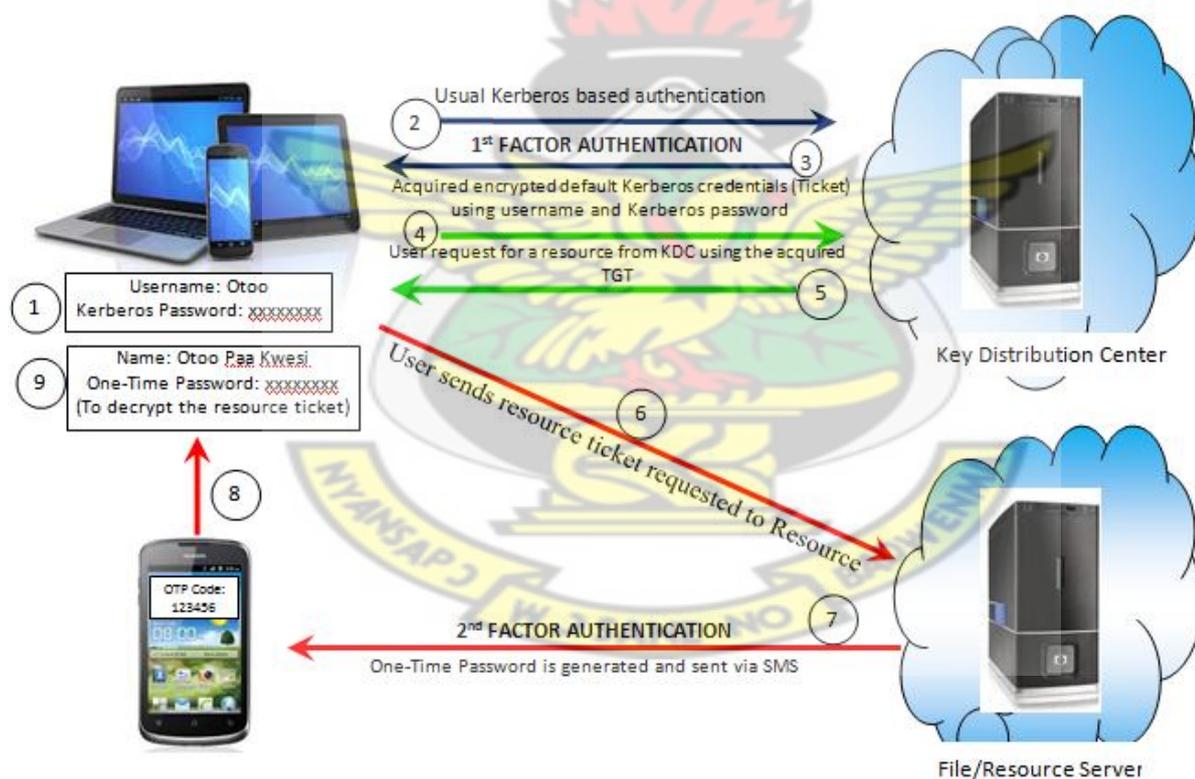


Figure 4.4 Existing Kerberos Protocol

Kerberos operates by encrypting data with a symmetric key. A symmetric key is a type of authentication where both the client and server agree to use a single encryption/decryption key for sending or receiving data. When working with the encryption key, the details are actually sent to a key distribution center, or KDC, instead of sending the details directly between each computer. The entire process takes a total of eight steps, as shown below.

Is all that back-and-forth communication really necessary? When concerning speed and reliability, it is entirely necessary. After the communication is made between the client and server, no further need of transmitting logon information is needed. The client is authenticated until the session expires. Figure 4.4 shows the authentication process.

#### 4.7. Proposed Kerberos Two-Factor Authentication System with OTP



**Figure 4.5 Proposed Kerberos Protocol**

Figure 4.5 describes the design steps for implementing a two-factor authentication system using Kerberos and OTP.

It shows a Kerberized client configured to a Kerberos server system. Either system can be an AIX box with IBM Network Authentication Service installed and configured.

The following explains the steps that can be implemented using a standard Kerberos Protocol to achieve a two-factor authentication login module with OTP.

**Steps 1** - Prompt the user to enter the Kerberos Username and Kerberos Password.

**Steps 2, 3** - Use the Kerberos username and the corresponding Kerberos password (which the user has to remember) to acquire the Encrypted Kerberos Credential (TGT- Ticket Granting Ticket).

If the password entered is incorrect, the authentication fails and the user is not allowed any access.

On successful acquisition of the Kerberos ticket (TGT), the first-factor authentication is completed.

This is similar to any regular login module that based on Kerberos authentication.

**Steps 4, 5** – The AS sends the generated OTP code through SMS to the User's Mobile Handset and again sends the Hash encrypted Ticket to the user's access point for decryption with the OTP code.

**Steps 6, 7** - The above-acquired Kerberos credential and establish a secure authentication based on the OTP application server residing on the system (assuming the OTP server has been Kerberized).

This involves a handshake between the client login module and decrypted TGT ticket with the OTP from the AS. Note that here both the login module and the OTP application running on the Kerberos mechanism.

On successful handshake, a secure authentication is established over the underlying Kerberos security mechanism. This context helps the client login module and the OTP application to communicate securely. If the handshake fails, the login program is terminated.

**Step 8, 9** – The AS grant the user a service ticket to access a requested service. The Hash OTP code residing on the AS is destroyed once the service is granted.

**Note:** These steps do not claim to have completely introduced two-factor authentication in a Kerberos protocol. They can be implemented by practitioners who require a two-factor authentication where the first factor needs to be a regular Kerberos authentication and the second

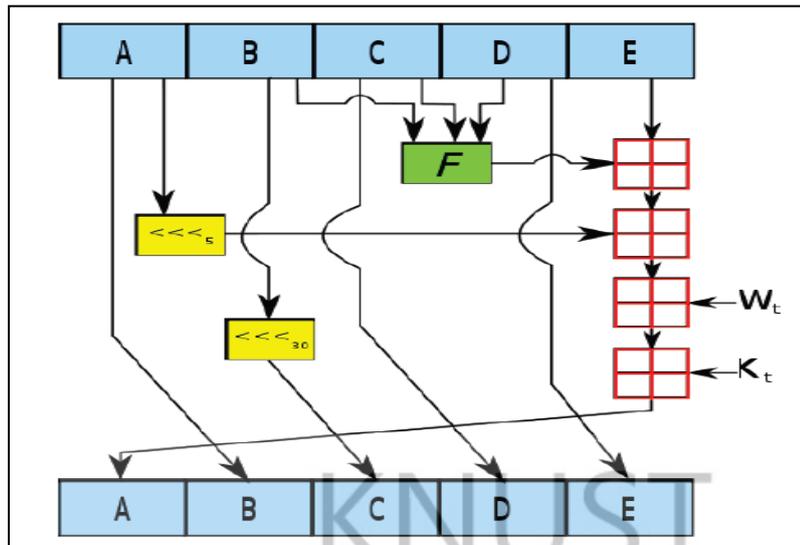
factor needs to be OTP, or they can be implemented as a part of login modules of applications or secure systems.

## 4.8. Algorithms for implementation

### 4.8.1 SHA-1 Algorithm

#### SHA1 Algorithm Description

- Padding
- ❖ Pad the message with a single one followed by zeroes until the final block has 448 bits.
- ❖ Append the size of the original message as an unsigned 64 bit integer.
- Initialize the 5 hash blocks (h0, h1, h2, h3, h4) to the specific constants defined in the SHA1 standard.
- Hash (for each 512bit Block)
- ❖ Allocate an 80 word array for the message schedule
  - Set the first 16 words to be the 512bit block split into 16 words.
  - The rest of the words are generated using the following algorithm
    - Word [i3] XOR word [i8] XOR word [i14] XOR word [i16] then rotated 1 bit to the left.
- ❖ Loop 80 times doing the following. (Shown in Image1)
  - Calculate SHA function () and the constant K (these are based on the current round number.
  - $e = d$
  - $d = c$
  - $c = b$  (rotated left 30)
  - $b = a$
  - $a = a$  (rotated left 5) + SHA function() + e + k + word[i]
- ❖ Add a, b, c, d and e to the hash output.
- Output the concatenation (h0,h1,h2,h3,h4) which is the message digest



**Figure 4.6:** 80 round inter-loops.

Figure 4.6 Shows hash diagram for construction a 512 bit block size and has a maximum message size of  $2^{64} - 1$ bits.

#### 4.8.2 OTP Algorithm

##### Hash-Based Message authentication Code (HMAC)

The current timestamp is turned into an integer time-counter (TC) by defining the start of an epoch (T0) and counting in units of a time step (TS). For example:

$$TC = (\text{unixtime (now)} - \text{unixtime (T0)}) / TS$$

TOTP = HOTP (SecretKey, TC), where the HOTP algorithm is defined below.

**TOTP-Value** = TOTP mod  $10^d$ , where  $d$  is the desired number of digits of the one-time password.

Let:

- $K$  be a secret key

- $C$  be a counter
- $HMAC(K,C) = SHA1(K \oplus 0x5c5c... \parallel SHA1(K \oplus 0x3636... \parallel C))$  be an [HMAC](#) calculated with the [SHA-1](#) cryptographic hash algorithm
- $Truncate$  be a function that selects 4 bytes from the result of the HMAC in a defined manner

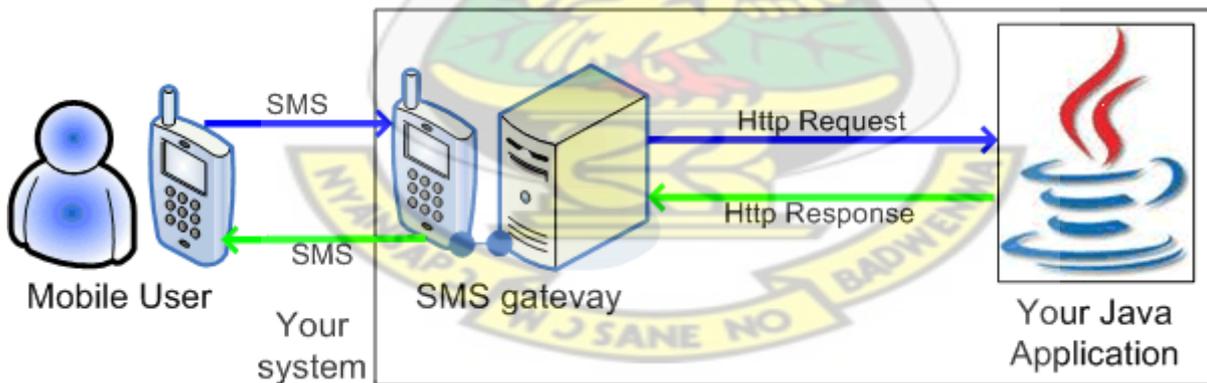
Defined as:

$$HOTP(K,C) = Truncate(HMAC(K,C)) \& 0x7FFFFFFF$$

The [mask](#) is used to disregard the [most significant bit](#) to provide better interoperability between processors.

### 4.8.3 Mobile SMS algorithm

Java has native method calls to submit HTTP requests. This means that HTTP is a good choice to send SMS text messages to mobile phones. If you operate an SMS gateway in your network, for example the Mtn SMS Gateway, you can pass SMS messages to it using HTTP GET or HTTP POST method calls (Figure 1).



Figure

### 4.6 – SMS messages from JAVA through HTTP

From Figure 4.3, we saw that to be able to communicate with mobile phones, you need to pass your messages to the HTTP Gateway. The HTTP Gateway has a built in web server, that provides an [HTTP SMS API](#) that makes it possible to submit messages. After your messages arrive to the SMS

Gateway, they will be sent to the mobile network through one of the channels you have configured previously.

This was done through, the [SMS gateway that operates on my mobile phone attached to my PC with a data cable](#) was used to send and receive your messages. It can also send the messages through the Internet to an SMS service provider.

The java codes for sending SMS as per Compact Disc.

#### 4.8.4 Kerberos Algorithm

##### Description

The client authenticates itself to the Authentication Server (AS) which forwards the username to a [Key distribution center](#) (KDC). The KDC issues a Ticket Granting Ticket (TGT), which is time stamped, encrypts it using the user's password and returns the encrypted result to the user's workstation. This is done infrequently, typically at user logon; the TGT expires at some point, though may be transparently renewed by the user's session manager while they are logged in.

When the client needs to communicate with another node ("principal" in Kerberos parlance) the client sends the TGT to the Ticket Granting Service (TGS), which usually shares the same host as the KDC. After verifying the TGT is valid and the user is permitted to access the requested service, the TGS issues a Ticket and session keys, which are returned to the client. The client then sends the Ticket to the service server (SS) along with its service request.

The protocol is described in detail below (Kerberos negotiations).

##### User Client-based Logon

1. A client enters a username and password on the [client machines](#).
2. The client transforms the password into the key of a symmetric cipher. This either uses the built in key scheduling.

## Client Authentication

1. The client sends a [clear text](#) message of the user ID to the AS requesting services on behalf of the user. (Note: Neither the secret key nor the password is sent to the AS.) The AS generates an OTP key and hashed the key generated using SHA-1 algorithm.
2. The AS checks to see if the client is in its database. If it is, the AS sends back the following two messages to the client:
  - Message A: *Client/TGS Session Key* encrypted using the secret key of the OTP.
  - Message B: *Ticket-Granting-Ticket* (which includes the client ID, client network address, ticket validity period, and the *client/TGS session key*) encrypted using the secret key of the TGS.
  - Message SM: OTP code sent via SMS to client Mobile device
3. Once the client receives messages A, B and SM, it attempts to decrypt message A with the SM generated from the OTP. If the user entered password does not match the password in the AS database, the client's secret key will be different and thus unable to decrypt message A. With a valid password and secret key the client decrypts message A to obtain the *Client/TGS Session Key*. This session key is used for further communications with the TGS. (Note: The client cannot decrypt Message B, as it is encrypted using TGS's secret key.) At this point, the client has enough information to authenticate itself to the TGS.

## Client Service Authorization

1. When requesting services, the client sends the following two messages to the TGS:
  - Message C: Composed of the TGT from message B and the ID of the requested service.
  - Message D: Authenticator (which is composed of the client ID and the timestamp), encrypted using the *Client/TGS Session Key*.
2. Upon receiving messages C and D, the TGS retrieves message B out of message C. It decrypts message B using the TGS secret key. This gives it the "client/TGS session key". Using this key, the TGS decrypts message D (Authenticator) and sends the following two messages to the client:

- Message E: *Client-to-server ticket* (which includes the client ID, client network address, validity period and *Client/Server Session Key*) encrypted using the service's secret key.
- Message F: *Client/Server Session Key* encrypted with the *Client/TGS Session Key*.

### **Client Service Request**

1. Upon receiving messages E and F from TGS, the client has enough information to authenticate itself to the SS. The client connects to the SS and sends the following two messages:
  - Message E from the previous step (the *client-to-server ticket*, encrypted using service's secret key).
  - Message G: a new Authenticator, which includes the client ID, timestamp and is encrypted using *Client/Server Session Key*.
2. The SS decrypts the ticket using its own secret key to retrieve the *Client/Server Session Key*. Using the sessions key, SS decrypts the Authenticator and sends the following message to the client to confirm its true identity and willingness to serve the client:
  - Message H: the timestamp found in client's Authenticator plus 1, encrypted using the *Client/Server Session Key*.
3. The client decrypts the confirmation using the *Client/Server Session Key* and checks whether the timestamp is correctly updated. If so, then the client can trust the server and can start issuing service requests to the server.
4. The server provides the requested services to the client.

## **4.9. Testing**

In the testing phase, the program was executed with a set of test cases and the output of the program for the test cases was evaluated to determine if the program performed as expected. Testing is the process of executing a program code with the intent of finding errors.

### **4.9.1 Test Priorities**

During the testing of the algorithms, the following qualities were tested.

- ✓ Kerberos Algorithm- whether the required functions are working as expected.
- ✓ SHA-1 Algorithm - whether the required functions are working as expected.
- ✓ OTP Algorithm - whether the required functions are working as expected.
- ✓ SMS Algorithm - whether the required functions are working as expected.

#### **4.9.2 Test Environment/ System Specification**

##### **Hardware and Software**

The test environment will consist of:

- ✓ Processor: Pentium IV Intel Processor
- ✓ Processor Speed: 2.10 GHz
- ✓ Memory (RAM): 3.00 GB
- ✓ System Type: 32/64 – bit Operating System
- ✓ Hard Disk Drive (HDD): At least 20 GB
- ✓ CD ROM Drive: 52X
- ✓ USB: 2.0
- ✓ Network Interface Card (NIC)
- ✓ Operating System: Windows XP/Visa/Windows 7 Ultimate.

#### **4.10. Simulation Platform**

This section describes the tools for the implementation, programming language and framework for the proposed two-factor Kerberos authentication with an OTP via mobile SMS.

#### **4.11. Tools for implementation**

##### **4.11.1 Integrated Development Tool (IDE) - Netbeans**

Netbeans IDE version 7.3 was used for the development of the model because Netbeans is designed as a modular developer tool for a wide range of development tasks. The base IDE includes an

advanced multi-language editor, debugger and profiler integration, file versioning control and unique developer collaboration features.

#### **4.11.2 GSM Modem (Mtn 3G Wireless modem)**

A GSM modem is a wireless modem that works with a GSM wireless network. A wireless modem behaves like a dial-up modem. The main difference between them is that a dial-up modem sends and receives data through a fixed telephone line while a wireless modem sends and receives data through radio waves. Like a GSM mobile phone, a GSM modem requires a SIM card from a wireless carrier in order to operate. Both GSM modems and dial-up modems support a common set of standard AT commands. You can use a GSM modem just like a dial-up modem.

### **4.12. Language**

#### **4.12.1 Java (Web Technology)**

Java is designed to enable development of portable, high-performance applications. Java is a powerful platform that includes a complete set of APIs for distributed applications. It allows programs to run anywhere on the network (i.e. Java is platform independent, secure, multi-threaded and dynamic programming language. It gives freedom to run application on any operating system).

#### **4.12 Summary**

This chapter has presented the design of the security services of the proposed two-factor Kerberos authentication with an OTP via mobile SMS and its corresponding implementation algorithms and protocols. It includes an existing Kerberos model, proposed two-factor Kerberos model, SAH-1 algorithm, OTP algorithm, SMS algorithm, system specification, simulation platform which includes tools for implementation, language and framework.

## **CHAPTER 5**

## Test and Evaluation of System

### 5.1 Introduction

This chapter presents the overall evaluation of the proposed Two-Factor Kerberos Authentication System with Mobile OTP security system from two perspectives: integration and security. Integration demonstrates how the proposed security services can be integrated within a cloud environment. Security demonstrates how securely the services are delivered to service requesters.

### 5.2. Java Security and Integration Advantages

Java has gained enormous popularity since it first appeared. Its rapid ascension and wide acceptance can be traced to its design and programming features, particularly in its promise that you can write a program once, and run it anywhere.

As stated in Java language white paper by Sun Microsystems: "Java is a simple, object-oriented, distributed, interpreted, robust, secure, architecture neutral, portable, multithreaded, and dynamic."

The advantages of Java are as follows:

1. **Java is easy to learn:** Java was designed to be easy to use and is therefore easy to write, compile, debug, and learn than other programming languages.
2. **Java is object-oriented:** This allows you to create modular programs and reusable code.
3. **Java is platform-independent:** One of the most significant advantages of Java is its ability to move easily from one computer system to another. The ability to run the same program on many different systems is crucial to World Wide Web software, and Java succeeds at this by being platform-independent at both the source and binary levels.
4. **Java is distributed:** Java is designed to make distributed computing easy with the networking capability that is inherently integrated into it. Writing network programs in Java is like sending and receiving data to and from a file.
5. **Java is secure:** Java considers security as part of its design. The Java language, compiler, interpreter, and runtime environment were each developed with security in mind.

6. **Java is robust:** Robust means reliability. Java puts a lot of emphasis on early checking for possible errors, as Java compilers are able to detect many problems that would first show up during execution time in other languages.
7. **Java is multithreaded:** Multithreaded is the capability for a program to perform several tasks simultaneously within a program. In Java, multithreaded programming has been smoothly integrated into it, while in other languages, operating system-specific procedures have to be called in order to enable multithreading.

Because of Java's robustness, ease of use, cross-platform capabilities and security features, it has become a language of choice for this thesis

### **5.3. Evaluation of System Security and Success of the Problem Solved**

Security evaluation is based on the attack-oriented threat model. Threat model gives a formal approach to order potential security issues that makes the system security evaluation easy to understand.

The proposed security system is analyzed for possible security threats, taking into account security considerations for Kerberos authentication systems, highlighted in Chapter 3.

In particular, Kerberos is not effective against password guessing attacks; if a user chooses a poor password, then an attacker guessing that password can impersonate the user. This in particular has been prevented using randomly generated OTP code via SMS.

Similarly, Kerberos requires a trusted path through which passwords are entered. If the user enters a password to a program that has already been modified by an attacker (a Trojan horse), or if the path between the user and the initial authentication program can be monitored, then an attacker may obtain sufficient information to impersonate the user. With this regards, the introduction of the second factor authentication will not give full access to the impersonator.

### **5.4. Weakness of the program**

Kerberos assumes that each user is trusted but is using an untrusted host on an untrusted network. Its primary goal is to prevent unencrypted passwords from being transmitted across that network. However, if anyone other than the proper user has access to the one host that issues tickets used for authentication — called the Key distribution center (KDC) - the entire Kerberos authentication system is at risk.

### 5.5. Summary

This chapter has presented the evaluation of the proposed security services from two aspects: integration and security. All advantages and disadvantages adopted from the Java technology have been highlighted for our system. Finally, it has been shown that the system is resistant to all potential security issues associated with the possible treat mentioned in chapter 3

## CHAPTER 6

### Conclusions and Future Work

This chapter summarizes the overall investigation of this thesis and recommends some future work in the research area.

#### 6.1. Comparison Table

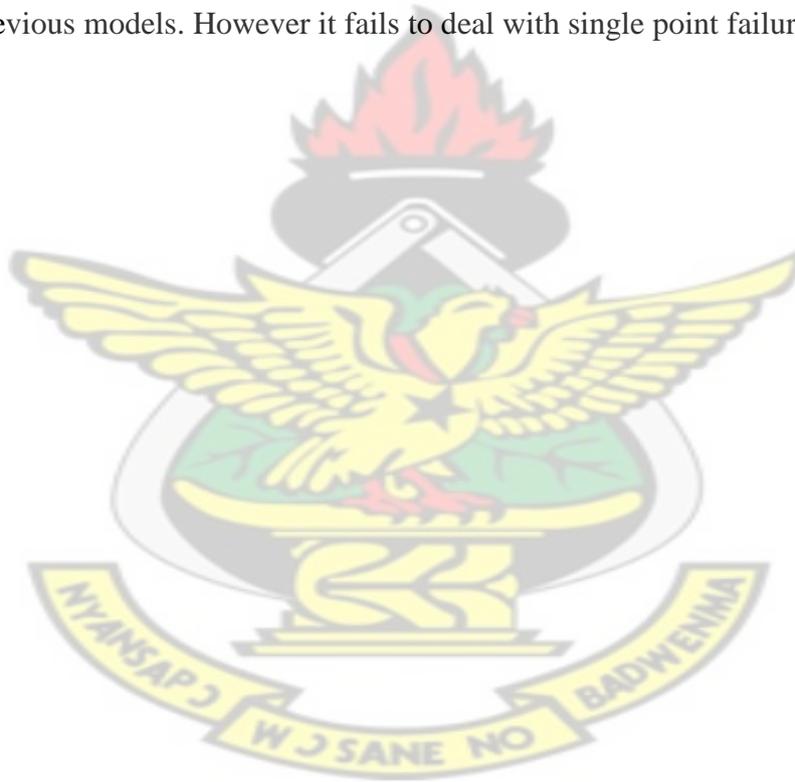
**Table 1: Comparison To Related Works**

Common Password & Authentication Attacks	Access Control System [4]	Data Storage Security [5]	T-CLOUD [6]	Secure Substantiation [7]	TrustCube or Trust [8]	Proposed Model
Replay Attack	✓	✗	✓	✓	✗	✓
Dictionary / Password Guess Attacks	✓	✗	✓	✓	✓	✓
Secure Host	✗	✗	✗	✗	✗	✗
Trusted Path	✗	✗	✗	✗	✗	✓

Impersonation	✓	✗	✓	✓	✓	✓
Single Point of Failure	✗	✗	✗	✗	✗	✗
Brute-force Attack	✓	✗	✓	✓	✓	✓
Password Encryption Over Network	✓	✓	✗	✓	✗	✓

Table 1 Compares the possible attacks and authentication issues to the previous related models proposed in section 2 to the proposed model in section 4.

Table 1 compares the possible password/authentication attacks of the previous related models to the proposed model. The proposed model deals better with password and authentication issues as compared to the previous models. However it fails to deal with single point failure and secure host.



## 6.2. Conclusion

Cloud Computing is a fast growing technology in the modern world but it needs some more security features to enhance it in the business world. Security depends upon the way Cloud service provider allows its client to come and get registered with his cloud network. This paper investigates about the problem of authentication in cloud computing environment. A method is being proposed to ensure the secure authentication in the cloud by Kerberos authentication service with OTP via Mobile SMS as its base.

Implementations of cloud security solutions under the concept of Security as a Service are in their awaking phase. The proposed cloud security system based on that concept and made contributions in the area of authentication services for a cloud environment. A problem has been solved and the goals have been achieved.

## 6.3. Future Work

In this research a cloud security system has been proposed for managing authentication services applying quite new cloud service paradigm, such as Security as a Service. As such, there is a need to do more comprehensive observations and activities within this area and here are some of them:

- Kerberos assumes that each user is trusted but is using an untrusted host on an untrusted network. Its primary goal is to prevent unencrypted passwords from being transmitted across that network. However, if anyone other than the proper user has access to the one host that issues tickets used for authentication — called the KDC -- the entire Kerberos authentication system is at risk. More needs to be done to improve this area.
- Cloud-based security service providers deal with end-users whose privacy should not be violated at all. Although the system promises that from theoretical perspective according to

the applied security techniques and approaches, there is a need to conduct focused practical activities within this area in order to see the real picture of the security system robustness against potential privacy vulnerabilities.

At the same time all security credentials are stored in the central security system, which makes it possible to link and trace end-user activities by cloud identity service provider.

- Centralization of the authentication services for a cloud environment represents another two issues: single point of failure and single target of attack. Therefore, there is a need to conduct extra work related to data replication and protection for solving those mentioned problems.
- Security evaluation of the proposed security services is based on security considerations associated only with Authentication security risks.

That is why there is a necessity to make additional security evaluations against security issues associated with other system aspects, such as hardware, software, etc.

- System performance should be evaluated in a scalable environment in order to measure how responsive it is in case of large amount of service requests. This will also show how resistant the system is against denial of service attacks.
- The proposed system supports delivery of only two identity services. Therefore, more identity service features can be added, such as session refreshment, etc.

## REFERENCES:

- [1] [http://en.wikipedia.org/wiki/Cloud\\_computing](http://en.wikipedia.org/wiki/Cloud_computing). [Accessed on: 13<sup>th</sup> September 2013]

- [2] <http://cloudtimes.org/2013/03/05/rackspace-survey-cloud-computing-startups-increase-profits/>. [Accessed on: 13<sup>th</sup> September 2013]
- [3] "Cloud Computing entry". NetLingo. Retrieved 15 January 2014.
- [4] <http://www.crn.com/news/cloud/240150619/the-100-coolest-cloud-computing-vendors-of-2013.htm>. [Accessed on: 13<sup>th</sup> September 2013]
- [5] "The NIST Definition of Cloud Computing". National Institute of Standards and Technology. Retrieved 24 July 2013.
- [6] "What is Cloud Computing?". *Amazon Web Services*. 2013-03-19. Retrieved 2013-03-20.
- [7] "Baburajan, Rajani, "The Rising Cloud Storage Market Opportunity Strengthens Vendors," infoTECH, August 24, 2011". It.tmcnet.com. 2011-08-24. Retrieved 2011-12-02.
- [8] Andreas Tolk. 2006. What Comes After the Semantic Web – PADS Implications for the Dynamic Web. 20th Workshop on Principles of Advanced and Distributed Simulation (PADS '06). IEEE Computer Society, Washington, DC, USA
- [9] "Cloud Computing: Clash of the clouds". *The Economist*. 2009-10-15. Retrieved 2009-11-03.
- [10] "Gartner Says Cloud Computing Will Be As Influential As E-business". Gartner. Retrieved 2010-08-22.
- [11] Voorsluys, William; Broberg, James; Buyya, Rajkumar (February 2011). "Introduction to Cloud Computing". In R. Buyya, J. Broberg, A. Goscinski. *Cloud Computing: Principles and Paradigms*. New York, USA: Wiley Press. pp. 1–44. ISBN 978-0-470-88799-8.
- [12] "Tony Shan, "Cloud Taxonomy and Ontology"". February 2009. Retrieved 2 February 2009.
- [13] "ITU-T NEWSLOG – CLOUD COMPUTING AND STANDARDIZATION: TECHNICAL REPORTS PUBLISHED". International Telecommunication Union (ITU). Retrieved 16 December 2012.

- [14] Amies, Alex; Sluiman, Harm; Tong, Qiang Guo; Liu, Guo Ning (July 2012). "Infrastructure as a Service Cloud Concepts". *Developing and Hosting Applications on the Cloud*. IBM Press. ISBN 978-0-13-306684-5.
- [15] Salvatore Distefano, Giovanni Merlino, Antonio Puliafito: Sensing and Actuation as a Service: A New Development for Clouds. 2012 IEEE 11th International Symposium on Network Computing and Applications, pp 272-275 [2]
- [16] Platform-as-a-Service Architecture for Real-Time Quality of Service Management in Clouds [3]
- [17] Hamdaqa, Mohammad. *A Reference Model for Developing Cloud Applications*.
- [18] Chou, Timothy. *Introduction to Cloud Computing: Business & Technology*.
- [2.1] Keunwang Lee Chungwoon and Haeseok Oh : Research on Access control Method by User Authority using Two-Factor Authentication , *Dept.of Multimedia Science, Chungwoon University & Dept of Computer Science, Gachon University South Korea, ASTL Vol. 24, pp. 172 - 175, 2013.*
- [2.2] Zhao.F, Nishide.T, Sakurai.K, Multi-User Keyword Search Scheme for Secure Data Sharing with Fine-Grained Access Control, *Lecture Notes in Computer Science, No.7259, pp.406-418(2012).*
- [2.3] Yaser Fuad Al-Dubai And Dr. Khamitkar S.D; Swami Ramanand: A Proposed Model For Data Storage Security In Cloud Computing Using Kerberos Authentication Service , *School Of Computational Sciences - Teerth Marathwada University, India., International Journal of Computer Engineering and Technology (IJCET), Volume 4, Issue 6, November - December (2013).*
- [2.4] Sultan Ullah, Zheng Xuefeng and Zhou Feng; Tcloud: A Multi – Factor Access Control Framework For Cloud Computing, *School of Computer and Communication Engineering, University of Science and Technology, Beijing. International Journal of Security and Its Applications Vol. 7, No. 2, March, 2013.*
- [2.5] Raja Shree S: Secure Substantiation in Cloud Computing Environment

*Jerusalem College of Engineering, Anna Univeristy Narayanapuram, India,*  
*International Journal of Modern Engineering Research (IJMER) www.ijmer.com Pp-42-*  
*46 ISSN: 2249-6645.*

- [2.6] R. Chow, M. Jakobsson, R. Masuoka, J. Molina, Y. Niu, E. Shieshi and Z. Song.,  
Authentication in the Clouds: A Framework and its Application to Mobile Users, *School  
of Computer and Communication Engineering, University of Science and Technology,  
Beijing*, International Journal of Security and Its Applications Vol. 7, No. 2, March, 2013.
- [2.7] S. Ullah and Z. Xuefeng, "Cloud Computing Research Challenges", In Proceedings of 5th  
IEEE International Conference on Biomedical Engineering and Informatics, (2012), pp.  
1397-1401.
- [2.8] E. -J. Yoon and K. -Y. Yoo, "Robust id-based remote mutual authentication with key  
agreement scheme for mobile devices on ecc", In Computational Science and Engineering,  
2009, CSE'09, International Conference on, vol. 2, IEEE, (2009), pp. 633-640.
- [2.9] J. Wiebelitz, S. Piger, C. Kunz and C. Grimm, "Transparent identity-based firewall  
transition for eScience", In E-Science Workshops, 2009 5th IEEE International  
Conference on, IEEE, (2009), pp. 3-10.
- [2.10] D. Zissis and D. Lekkas, "Addressing cloud computing security issues", Future  
Generation Computer Systems, vol. 28, no. 3, (2012), pp. 583-592.
- [2.11] Cloud Security Alliance, Security Guidance for Critical Areas of Focus in Cloud  
Computing V2.1, <http://www.cloudsecurityalliance.org/>, December 2009
- [2.12] B. Clifford Neumann, Theodore Ts'o —Kerberos: An Authentication Service for  
Computer Networks, IEEE Communications Magazine, Volume 32, Number 9, pages 33-  
38, September 1994.
- [3.0] SALTZER, J.H. and SCHROEDER, M.D., 1975. The protection of information in  
computer systems. Proceedings of the IEEE, 63(9), pp. 1278-1308.
- [3.1] REID, B., 1991. Reflections on some recent widespread computer break-ins, Computers  
under attack: intruders, worms, and viruses 1991, ACM, pp. 145-149.

- [3.2] BISHOP, M., 2005. Realigning Usability and Security. *Security and usability*, , pp. 103-128.
- [3.3] ADAMS, A. and SASSE, M.A., 1999. Users are not the enemy. *Communications of the ACM*, 42(12), pp. 40-46.
- [3.4] DE ALVARE, A. and SCHULTZ JR, E., 1988. A framework for password selection.[*Password recommendations*], .
- [3.5] FIPS 112 - Password Usage.1985. FIPS 112 - Password Usage. [ONLINE] Available at: [http:// www.itl.nist.gov/fipspubs/fip112.htm](http://www.itl.nist.gov/fipspubs/fip112.htm). [Accessed 23 September 2012].
- [3.6] RENAUD, K., 2012. Blaming Noncompliance Is Too Convenient: What Really Causes Information Breaches? *Security & Privacy, IEEE*, 10(3), pp. 57-63.
- [3.7] NORMAN, D.A., 2009. THE WAY I SEE IT when security gets in the way. *Interactions*, 16(6), pp. 60-63.
- [3.8] KAMP, P., GODEFROID, P., LEVIN, M.Y., MOLNAR, D., MCKENZIE, P., STAPLETON-GRAY, R., WOODCOCK, B. and NEVILLE-NEIL, G.V., 2012. LinkedIn Password Leak: Salt Their Hide. *Queue*, 10(6), pp. 20.
- [3.1] <http://www.wikipedia.com/security-central/gartner-seven-cloud-computing-security-risks-8>
- [3.9] Splash Data, 2012. Scary Logins: Worst Passwords of 2012 — and How to Fix Them. Available at:<http://splashdata.com/press/PR121023.htm>. [Accessed 17 March 2013].
- [3.10] SASSE, M.A., BROSTOFF, S. and WEIRICH, D., 2001. Transforming the ‘weakest link’—a human/computer interaction approach to usable and effective security. *BT technology journal*, 19(3), pp. 122-131.
- [3.11] GRANGER, S., 2001. Social engineering fundamentals, part I: hacker tactics. *Security Focus*, December, 18.
- [3.12] BONNEAU, J. and PREIBUSCH, S., 2010. The password thicket: technical and market failures in human authentication on the web, *Proc. WEIS 2010*.
- [3.13] BestBuy, 2013. Password Help [ONLINE] Available at:

<http://www.bestbuy.com/site/Using-My-Account/Password-Help/pcmcat204400050052.c?id=pcmcat204400050052> [Accessed 20 June 2013].

- [3.14] Cardinal, 2012. Two-for-one: Amazon.com's Socially Engineered Replacement Order Scam: HTMLList.com, A Web Development Blog by Synapse Studios. [ONLINE] Available at: <http://www.htmlist.com/rants/two-for-one-amazon-coms-socially-engineered-replacement-order-scam/>. [Accessed 17 March 2013].
- [3.15] <http://www.wikipedia.com/security-central/gartner-seven-cloud-computing-security-risks-8>. [Accessed 17 April 2013].
- [4.1] <http://en.wikipedia.org/wiki/File:SHA1.svg> [Accessed 17 March 2013].
- [4.2] [http://en.wikipedia.org/wiki/Kerberos\\_protocol](http://en.wikipedia.org/wiki/Kerberos_protocol). [Accessed 17 December, 2013].
- [4.3] [http://en.wikipedia.org/wiki/Time-based\\_One-time\\_Password\\_Algorithm](http://en.wikipedia.org/wiki/Time-based_One-time_Password_Algorithm). [Accessed 17 December 2013].
- [4.4] B. Clifford Neuman and Theodore Ts'o (September 1994). "[Kerberos: An Authentication Service for Computer Networks](#)". *IEEE Communications* 32 (9): 33–8. doi:10.1109/35.312841.
- [4.5] John T. Kohl, B. Clifford Neumann, and Theodore Y. T'so (1994). "[The Evolution of the Kerberos Authentication System](#)" (Postscript). In Johansen, D.; Brazier, F. M. T. *Distributed open systems*. Washington: IEEE Computer Society Press. pp. 78–94. ISBN 0-8186-4292-0.
- [4.6] "[Kerberos Overview: An Authentication Service for Open Network Systems](#)". Cisco Systems date=19 January 2006. Retrieved 15 August 2012.
- [4.7] "[How Kerberos Authentication Works](#)". Learn-networking.com. 28 January 2008. Retrieved 15 August 2012.
- [4.8] Gil Held: "Data over Wireless Networks." pages 105–11, 137–38. Wiley, 2001.
- [4.9] <https://cs.uwaterloo.ca/research/tr/2007/CS-2007-42.pdf>. [Accessed 20th March 2014].
- [4.10] <http://thejavamonkey.blogspot.com/2008/07/.html>. [Accessed 20th March 2014].

## REFERENCES:

- [19] [http://en.wikipedia.org/wiki/Cloud\\_computing](http://en.wikipedia.org/wiki/Cloud_computing). [Accessed on: 13<sup>th</sup> September 2013]
- [20] <http://cloudtimes.org/2013/03/05/rackspace-survey-cloud-computing-startups-increase-profits/>. [Accessed on: 13<sup>th</sup> September 2013]
- [21] "Cloud Computing entry". NetLingo. Retrieved 15 January 2014.
- [22] <http://www.crn.com/news/cloud/240150619/the-100-coolest-cloud-computing-vendors-of-2013.htm>. [Accessed on: 13<sup>th</sup> September 2013]
- [23] "The NIST Definition of Cloud Computing". National Institute of Standards and Technology. Retrieved 24 July 2013.
- [24] "What is Cloud Computing?". *Amazon Web Services*. 2013-03-19. Retrieved 2013-03-20.
- [25] "Baburajan, Rajani, "The Rising Cloud Storage Market Opportunity Strengthens Vendors," infoTECH, August 24, 2011". It.tmcnet.com. 2011-08-24. Retrieved 2011-12-02.
- [26] Andreas Tolk. 2006. What Comes After the Semantic Web – PADS Implications for the Dynamic Web. 20th Workshop on Principles of Advanced and Distributed Simulation (PADS '06). IEEE Computer Society, Washington, DC, USA
- [27] "Cloud Computing: Clash of the clouds". *The Economist*. 2009-10-15. Retrieved 2009-11-03.
- [28] "Gartner Says Cloud Computing Will Be As Influential As E-business". Gartner. Retrieved 2010-08-22.
- [29] Voorsluys, William; Broberg, James; Buyya, Rajkumar (February 2011). "Introduction to Cloud Computing". In R. Buyya, J. Broberg, A.Goscinski. *Cloud Computing: Principles and Paradigms*. New York, USA: Wiley Press. pp. 1–44. ISBN 978-0-470-88799-8.
- [30] "Tony Shan, "Cloud Taxonomy and Ontology"". February 2009. Retrieved 2 February 2009.

- [31] "ITU-T NEWSLOG – CLOUD COMPUTING AND STANDARDIZATION: TECHNICAL REPORTS PUBLISHED". International Telecommunication Union (ITU). Retrieved 16 December 2012.
- [32] Amies, Alex; Sluiman, Harm; Tong, Qiang Guo; Liu, Guo Ning (July 2012). "Infrastructure as a Service Cloud Concepts". *Developing and Hosting Applications on the Cloud*. IBM Press. ISBN 978-0-13-306684-5.
- [33] Salvatore Distefano, Giovanni Merlino, Antonio Puliafito: Sensing and Actuation as a Service: A New Development for Clouds. 2012 IEEE 11th International Symposium on Network Computing and Applications, pp 272-275 [2]
- [34] Platform-as-a-Service Architecture for Real-Time Quality of Service Management in Clouds [3]
- [35] Hamdaqa, Mohammad. *A Reference Model for Developing Cloud Applications*.
- [36] Chou, Timothy. *Introduction to Cloud Computing: Business & Technology*.
- [2.1] Keunwang Lee Chungwoon and Haeseok Oh : Research on Access control Method by User Authority using Two-Factor Authentication , *Dept.of Multimedia Science, Chungwoon University & Dept of Computer Science, Gachon University South Korea, ASTL Vol. 24, pp. 172 - 175, 2013.*
- [2.2] Zhao.F, Nishide.T, Sakurai.K, Multi-User Keyword Search Scheme for Secure Data Sharing with Fine-Grained Access Control, *Lecture Notes in Computer Science, No.7259, pp.406-418(2012).*
- [2.3] Yaser Fuad Al-Dubai And Dr. Khamitkar S.D; Swami Ramanand: A Proposed Model For Data Storage Security In Cloud Computing Using Kerberos Authentication Service , *School Of Computational Sciences - Teerth Marathwada University, India., International Journal of Computer Engineering and Technology (IJCET), Volume 4, Issue 6, November - December (2013).*
- [2.4] Sultan Ullah, Zheng Xuefeng and Zhou Feng; Tcloud: A Multi – Factor Access Control Framework For Cloud Computing, *School of Computer and Communication Engineering, University of Science and Technology, Beijing. International Journal of Security and Its Applications Vol. 7, No. 2, March, 2013.*

- [2.5] Raja Shree S: Secure Substantiation in Cloud Computing Environment  
*Jerusalem College of Engineering, Anna Univeristy Narayanapuram, India,*  
*International Journal of Modern Engineering Research (IJMER) www.ijmer.com Pp-42-46 ISSN: 2249-6645.*
- [2.6] R. Chow, M. Jakobsson, R. Masuoka, J. Molina, Y. Niu, E. Shieshi and Z. Song.,  
Authentication in the Clouds: A Framework and its Application to Mobile Users, *School of Computer and Communication Engineering, University of Science and Technology, Beijing*, International Journal of Security and Its Applications Vol. 7, No. 2, March, 2013.
- [2.7] S. Ullah and Z. Xuefeng, "Cloud Computing Research Challenges", In Proceedings of 5th IEEE International Conference on Biomedical Engineering and Informatics, (2012), pp. 1397-1401.
- [2.8] E. -J. Yoon and K. -Y. Yoo, "Robust id-based remote mutual authentication with key agreement scheme for mobile devices on ecc", In Computational Science and Engineering, 2009, CSE'09, International Conference on, vol. 2, IEEE, (2009), pp. 633-640.
- [2.9] J. Wiebelitz, S. Piger, C. Kunz and C. Grimm, "Transparent identity-based firewall transition for eScience", In E-Science Workshops, 2009 5th IEEE International Conference on, IEEE, (2009), pp. 3-10.
- [2.10] D. Zissis and D. Lekkas, "Addressing cloud computing security issues", Future Generation Computer Systems, vol. 28, no. 3, (2012), pp. 583-592.
- [2.11] Cloud Security Alliance, Security Guidance for Critical Areas of Focus in Cloud Computing V2.1, <http://www.cloudsecurityalliance.org/>, December 2009
- [2.12] B. Clifford Neumann, Theodore Ts'o —Kerberos: An Authentication Service for Computer Networks, IEEE Communications Magazine, Volume 32, Number 9, pages 33-38, September 1994.
- [3.0] SALTZER, J.H. and SCHROEDER, M.D., 1975. The protection of information in computer systems. Proceedings of the IEEE, 63(9), pp. 1278-1308.
- [3.1] REID, B., 1991. Reflections on some recent widespread computer break-ins, Computers under attack: intruders, worms, and viruses 1991, ACM, pp. 145-149.

- [3.2] BISHOP, M., 2005. Realigning Usability and Security. Security and usability, , pp. 103-128.
- [3.3] ADAMS, A. and SASSE, M.A., 1999. Users are not the enemy. Communications of the ACM, 42(12), pp. 40-46.
- [3.4] DE ALVARE, A. and SCHULTZ JR, E., 1988. A framework for password selection.[Password recommendations], .
- [3.5] FIPS 112 - Password Usage.1985. FIPS 112 - Password Usage. [ONLINE] Available at: <http://www.itl.nist.gov/fipspubs/fip112.htm>. [Accessed 23 September 2012].
- [3.6] RENAUD, K., 2012. Blaming Noncompliance Is Too Convenient: What Really Causes Information Breaches? Security & Privacy, IEEE, 10(3), pp. 57-63.
- [3.7] NORMAN, D.A., 2009. THE WAY I SEE IT when security gets in the way. Interactions, 16(6), pp. 60-63.
- [3.8] KAMP, P., GODEFROID, P., LEVIN, M.Y., MOLNAR, D., MCKENZIE, P., STAPLETON-GRAY, R., WOODCOCK, B. and NEVILLE-NEIL, G.V., 2012. LinkedIn Password Leak: Salt Their Hide. Queue, 10(6), pp. 20.
- [3.1] <http://www.wikipedia.com/security-central/gartner-seven-cloud-computing-security-risks-8>
- [3.9] Splash Data, 2012. Scary Logins: Worst Passwords of 2012 — and How to Fix Them. Available at:<http://splashdata.com/press/PR121023.htm>. [Accessed 17 March 2013].
- [3.10] SASSE, M.A., BROSTOFF, S. and WEIRICH, D., 2001. Transforming the ‘weakest link’—a human/computer interaction approach to usable and effective security. BT technology journal, 19(3), pp. 122-131.
- [3.11] GRANGER, S., 2001. Social engineering fundamentals, part I: hacker tactics. Security Focus, December, 18.
- [3.12] BONNEAU, J. and PREIBUSCH, S., 2010. The password thicket: technical and market failures in human authentication on the web, Proc. WEIS 2010.
- [3.13] BestBuy, 2013. Password Help [ONLINE] Available at:

<http://www.bestbuy.com/site/Using-My-Account/Password-Help/pcmcat204400050052.c?id=pcmcat204400050052> [Accessed 20 June 2013].

- [3.14] Cardinal, 2012. Two-for-one: Amazon.com's Socially Engineered Replacement Order Scam: HTMLList.com, A Web Development Blog by Synapse Studios. [ONLINE] Available at: <http://www.htmlist.com/rants/two-for-one-amazon-coms-socially-engineered-replacement-order-scam/>. [Accessed 17 March 2013].
- [3.15] <http://www.wikipedia.com/security-central/gartner-seven-cloud-computing-security-risks-8>. [Accessed 17 April 2013].
- [4.1] <http://en.wikipedia.org/wiki/File:SHA1.svg> [Accessed 17 March 2013].
- [4.2] [http://en.wikipedia.org/wiki/Kerberos\\_protocol](http://en.wikipedia.org/wiki/Kerberos_protocol). [Accessed 17 December, 2013].
- [4.3] [http://en.wikipedia.org/wiki/Time-based\\_One-time\\_Password\\_Algorithm](http://en.wikipedia.org/wiki/Time-based_One-time_Password_Algorithm). [Accessed 17 December 2013].
- [4.4] B. Clifford Neuman and Theodore Ts'o (September 1994). "[Kerberos: An Authentication Service for Computer Networks](#)". *IEEE Communications* **32** (9): 33–8. doi:10.1109/35.312841.
- [4.5] John T. Kohl, B. Clifford Neumann, and Theodore Y. T'so (1994). "[The Evolution of the Kerberos Authentication System](#)" (Postscript). In Johansen, D.; Brazier, F. M. T. *Distributed open systems*. Washington: IEEE Computer Society Press. pp. 78–94. ISBN 0-8186-4292-0.
- [4.6] "[Kerberos Overview: An Authentication Service for Open Network Systems](#)". Cisco Systems date=19 January 2006. Retrieved 15 August 2012.
- [4.7] "[How Kerberos Authentication Works](#)". Learn-networking.com. 28 January 2008. Retrieved 15 August 2012.
- [4.8] Gil Held: "Data over Wireless Networks." pages 105–11, 137–38. Wiley, 2001.
- [4.9] <https://cs.uwaterloo.ca/research/tr/2007/CS-2007-42.pdf>. [Accessed 20th March 2014].
- [4.10] <http://thejavamonkey.blogspot.com/2008/07/html>. [Accessed 20th March 2014].

## Appendix A SIMULATION OF RESULT

Fig 3 shows the **Kerberos at Test Run** for the proposed model.

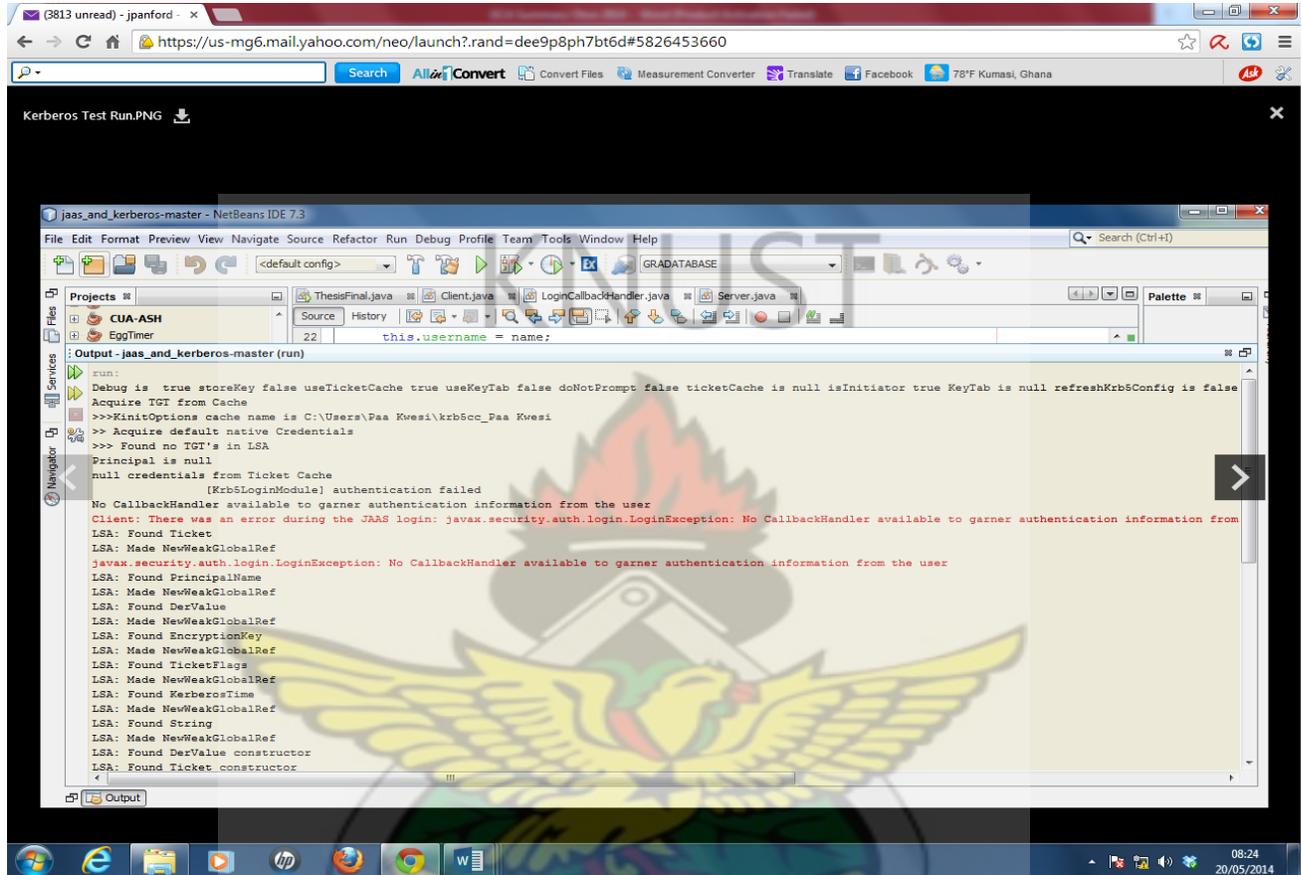


Fig 3: Kerberos at Test Run

## Appendix B SIMULATION OF RESULT

Fig 4 shows the **OTP at test run** for the proposed model.

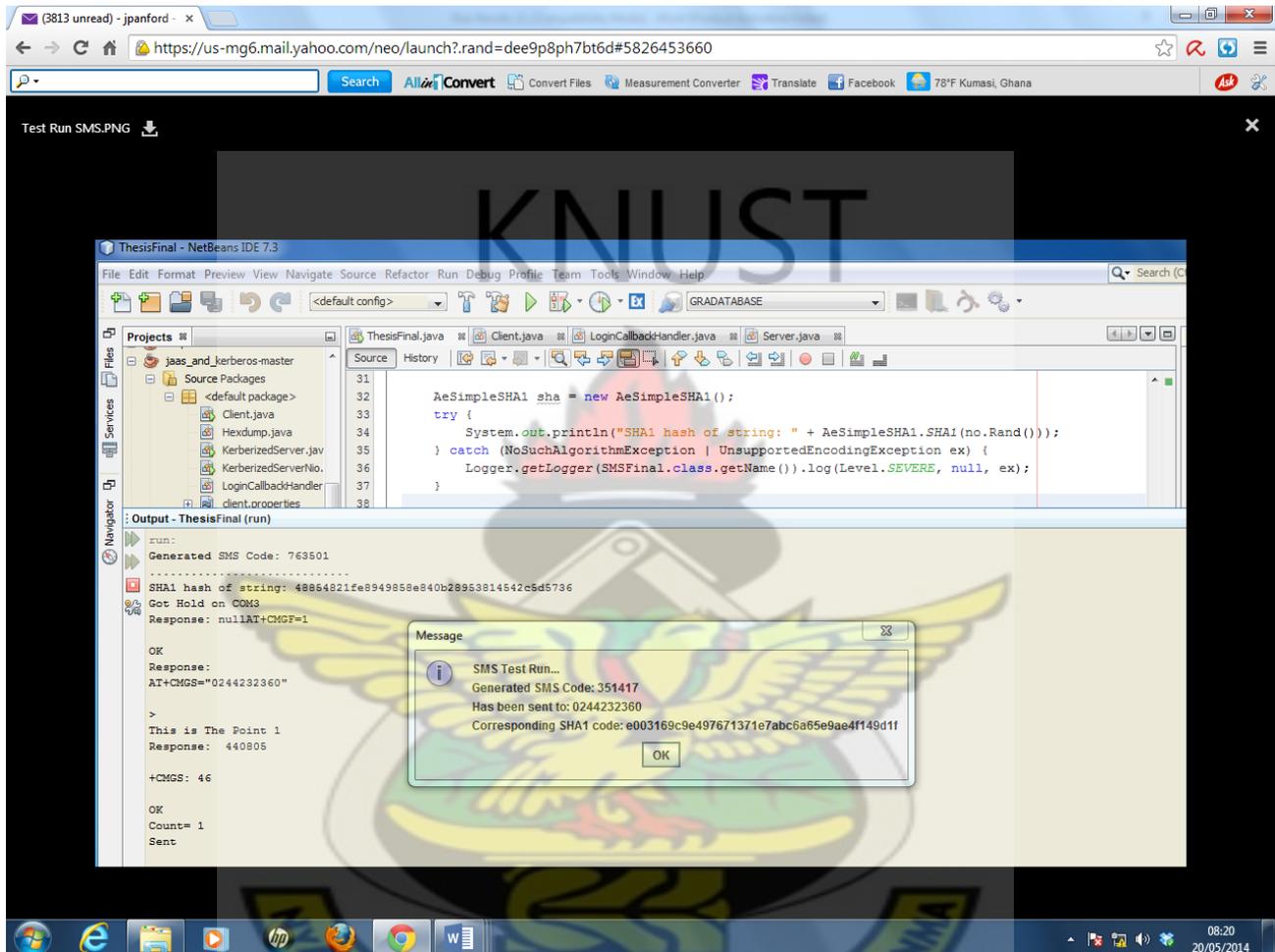


Fig 3:OTP SMS at Test Runtime

## Appendix C SIMULATION CODE

Find attached CD for code and Server Configuration Instructions on Kerberos and (OTP)SHA-1 code (Netbeans 7.3).