# AN AUTHENTICATION FRAMEWORK FOR SECURING GUISET INFRASTRUCTURE

**NOLUTHANDO BRILLIANT MHLONGO**

**(20042343)**

(BSc. Hons. Computer Science)

(University of Zululand)

A dissertation submitted in fulfillment of the requirements for the degree

of

Master of Science in Computer Science

**Supervisor: Prof. M. O. Adigun**

Department of Computer Science,

Faculty of Science and Agriculture

University of Zululand

2011

# DECLARATION

This dissertation represents the author's own research work and the work has not been submitted in any form to another tertiary institution for another degree or diploma. All the material used as source has been acknowledged in the text.

---

**Signature**

# DEDICATION

I dedicate this work to my family for believing in me and supporting me through the course of this work.

# ACKNOWLEDGEMENTS

I would like to thank the Lord for giving the strength, courage and guidance throughout the course of this work. I would also like to extend my gratitude to my supervisor Prof M. O. Adigun for providing me with this opportunity, guiding, supporting and making this work realizable. I would also like to thank Prof S. S. Xulu, for assisting me in finishing up my work. I would like to give many thanks to Mr. E. Jembere, who has been very supportive and has sacrificed his time in guiding me throughout my work.

I would also like to thank my research colleagues at the Center for Mobile e-Services for providing a stimulating and fun environment in which to learn and grow. I wish to give many thanks to my Sponsors, Telkom for funding me, providing me with the opportunity to achieve my goal.

# TABLE OF CONTENTS

# LIST OF FIGURES

## LIST OF TABLES

## LIST OF ACRONYMS

| | |
|---|---|
| **AS** | Authentication Server |
| **CA** | Certification Authority |
| **GAMA** | Grid Account Management Architecture |
| **GA** | GUISET Administrator |
| **GSI** | Grid Security Infrastructure |
| **GT** | Globus Toolkit |
| **GUISET** | Grid-based Utility Infrastructure for SMMEs Enabling Technology |
| **HTTPS** | Hypertext Transfer Protocol |
| **IBAG** | Identity-based Architecture for Grid |
| **IBC** | Identity-based Cryptography |
| **IBE** | Identity-based Encryption |
| **IBS** | Identity-based Signature |
| **ID-PKC** | Identity-based Public Key Cryptography |
| **KGCA** | Key Generation and Certification Authority |
| **LDAP** | Light Directory Access Protocol |
| **PECF-GSI** | Password-enabled and certificate free grid infrastructure |

| | |
|---|---|
| **PKI** | Public Key Infrastructure |
| **RA** | Registration Authority |
| **RCP** | Resource Consumer Proxy |
| **RPP** | Resource Provider Proxy |
| **RSA** | Rivest, Shamir, Adleman |
| **SMMEs** | Small, Medium and Micro Enterprises |
| **SSL** | Secure Socket Layer |
| **SOAP** | Simple Object Access Protocol |
| **TA** | Trusted Authority |
| **TLS** | Transport Layer Security |
| **UPKI** | Unified Public Key Infrastructure |
| **VO** | Virtual Organization |
| **XML** | Extensible Mark-up Language |


## ABSTRACT

Grid computing has appeared as a progressive and substantial area which has gained considerable attention from academic and business environments. It provides large, flexible resource sharing environment where resources and services are distributed over distributed administrative domains. Although there are many corporations benefiting from the grid computing technology, SMMEs still face some difficulties in exploiting these capabilities because of their lack of understanding the benefits that e-commerce enabling technology can provide. GUISET was developed to enable the integration of skills and resources by the SMMEs so that they can share and collaborate between themselves and independent business associates.

With GUISET technology based on an open resource sharing paradigm, and provision of unlimited access to end users, comes with a great challenge of security.

The widely deployed technology for securing these shared grid resources exploited by end users is the public key infrastructure. It proves the identities of grid users through the use of certificates. GSI is one of the implementations of the PKI technology and delivers essential requirements in a security infrastructure and those include delegation services, mutual authentication and single sign-on. However, the GSI has been discovered to suffer from poor scalability because of the use of certificates.

In addressing the above-mentioned challenge, an authentication infrastructure for GUISET has been developed. The model employs the distribution of certification authorities granting identification to users requiring access to GUISET resources. An implementation was performed for evaluating the performance of the proposed model. The results showed that the newly developed authentication framework for securing GUISET resources provides more scalability than the previously developed grid security infrastructure.

# CHAPTER ONE

# INTRODUCTION

## 1.1 Background

The continuing developments to computing power and storage capacity enable computing technologies of precedently unheard levels of sophistication (Lim, 2006). However, the existence of large and new problems result into high solicitation of accessing more computational power and resources. Consequently, grid has appeared as a promising technology to tackle these

requirements. The focus of grid computing has been described by (Foster et al, 2001) as "resource sharing and problem solving in a dynamic and multi-institutional virtual organization". Grid computing basically focuses on enabling individuals to utilize these shareable resources which may be databases, storage, applications, and many other hardware and software devices.

One sector that stands to benefit from the emerging collaborative approach to computing are the Small, Medium and Micro Enterprises (SMMEs). Many of these enterprises engage in e-commerce with the hope of exploiting their own competence and to profit from partners' abilities and capabilities through collaboration (Volker, 2001). By doing so, the SMMEs face competition in the global electronics markets. Although there are many studies on the benefits and problems associated with the adoption of e-commerce, most of them have focused on the big corporations, with little reference or attention to the SMMEs. The adoption rate of e-commerce is even much slower among deep rural SMMEs as well as their communities. While globalization provides many opportunities to these deep rural SMMEs, they have not taken full advantage of its potential and to compete strongly in such a global market. This is because of various problems overwhelming them such as insufficient knowledge on making good decisions about utilizing the technology and market features limiting the technology to be implemented.

Existing literature on SMMEs and IT as a facilitating tool for e-commerce is predominant with the challenges the rural SMMEs are encountering. These challenges also include the cost of IT, lack of IT knowledge, deficiency of using independent consultants and vendors, inadequate administration views, being unenlightened about the advantages that e-commerce enabling technologies can offer and determining the benefits, and lack of formal planning or control procedures (Burgess, 2002).

To address this, a solution that is technologically tailored towards meeting local needs, has been proposed to provide a Grid-based Utility Infrastructure for SMME enabled Technology (GUISET) (Adigun et al, 2006). The fundamental significance of GUISET is to propose an e-infrastructure where SMMEs (especially deep rural SMMEs) are able to integrate their resources and knowledge for sharing and collaboration among themselves and their independent business associates (Kabanda et al, 2007). The idea here is motivated by an ongoing technological convergence between grid and web services emerging service-oriented architectures which is creating a trend towards IT services provisioning as utilities. This includes provisioning of computing, data information and knowledge capabilities (Kabanda et al, 2007).

GUISET is a three-tier layer architecture consisting of the grid infrastructure layer, the middleware infrastructure layer, and the multi-modal interface. The Grid infrastructure layer provides services and resources, which are managed by the middleware infrastructure. The Grid infrastructure layer provides a service user with repository of business support services, which are accessible on demand with the aid of an interface.

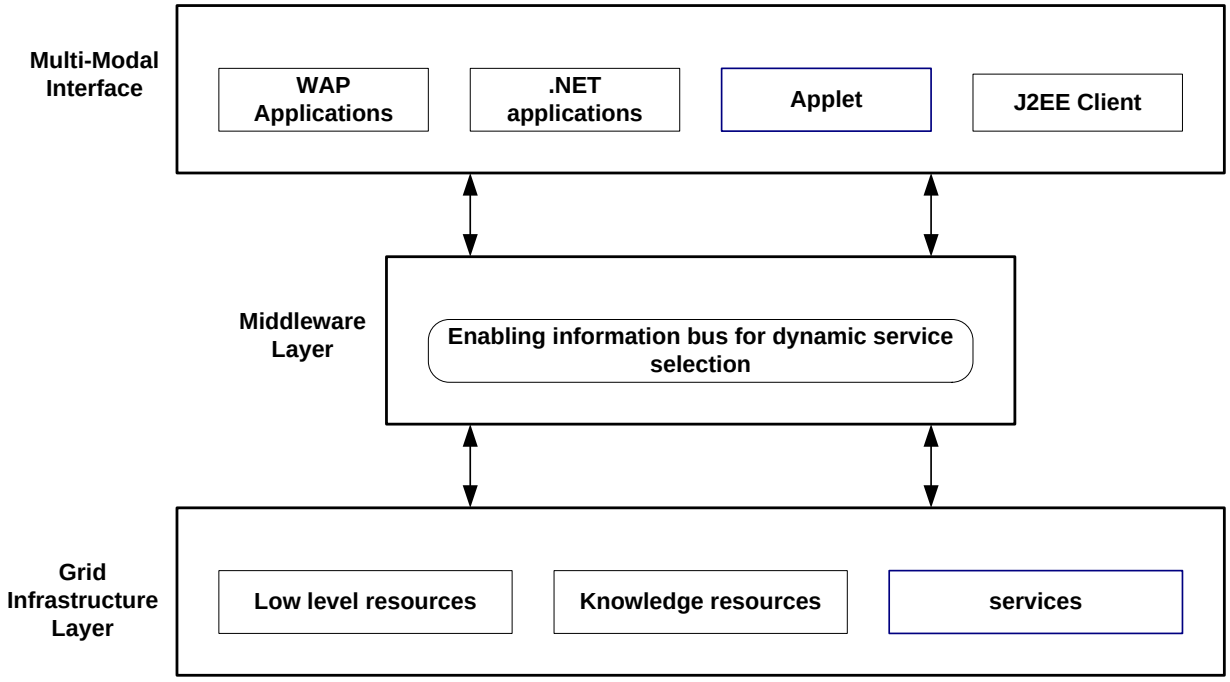

Figure 1.1 the GUISET Architecture (Adigun et al, 2006)

The middleware infrastructure layer is responsible for administering the services and resources provided by the grid infrastructure layer. It has the utility broker, which enables services to be consumed by utility services. The multimodal interface then provides support for universal accessibility to all services of device type, user diversity and execution environments. GUISET as a grid-based infrastructure may also involve a number of participants, each with different expertise. This results in security complications which are not well addressed by existing grid system technologies.

Provisioning of appropriate security appears to be easier in conventional distributed systems than it is in grid environments like GUISET. The Public Key Infrastructure (PKI) technology is the currently implemented technology in many grid infrastructures because of its widely sustained consecutiveness and sophistication and can be collaborated with diverse applications on heterogeneous platforms (Lim and Paterson, 2005). The Globus Toolkit (GT) is the leading toolkit used in implementing many grid security systems. The GT uses the standard X.509 public

key certificates, and thus has designed the proxy certificates in addition to the X.509. These certificates deal with the disadvantages prevailing in the traditional PKI structure and to provide more capabilities that coordinate with the requirements for secure communications among grid users within effectively changing environment. There are two advantages in using proxy certificates: (i) long-term credentials disclosure, and, (ii) enabling single sign-on and delegation services (Lim, 2006).

One infrastructure that adopted the PKI technology is the Grid Security Infrastructure which uses the SSL Authentication Protocol (SAP) to accomplish mutual authentication amongst the user proxy and resource proxy. Consequently, certificates under the organizations of the standard certificate-based Public Key Infrastructure have been determined by the user proxy and resource proxy. However, the scalability concern results from this pervasive utilization of certificates in the hierarchical PKI organization within a vigorously fluctuating grid environment. And this causes the limitation in the number of resource allocation sessions that can be created by the user proxy, which in turn limits the degree of performance grid computing services available to the user (Lim and Paterson, 2005).

With the widely adopted security mechanism having a scalability pitfall, we presume that a more lightweight authentication mechanism than a traditional PKI is required within the grid environment, in this case, the GUISET. This research work is specifically aimed at enhancing the Grid-based Utility Infrastructure for SMME enabled Technology by providing it with a more scalable authentication mechanism. We want to determine if our certificate based approach can present a more sufficient alternative than the PKI-based GSI.

## 1.2 Statement of the Problem

GUISET is a technology enabling the sharing and collaboration of resources between SMMEs. This thus means GUISET comprises of many sites with computational resources from which a virtual organization (VO) of high performance services can be integrated for users in need. With the GUISET's technology based on an open resource sharing paradigm, and the provision of extensive power and information access to GUISET entities, comes a great challenge of federated security. Currently the public key infrastructure is the widely deployed technology for proving the identities of grid users through the use of certificates when accessing grid resources. The deployment of the PKI based technology delivers essential requirements in a security infrastructure and those include delegation services, mutual authentication and single sign-on.

One infrastructure that has employed the PKI technology is the Grid Security Infrastructure, which uses the X.509 certificates (Foster, 1998). The use of public key certificates support authentication and key agreement protocols such as TLS. The certificates enable single sign-on and delegation. The certificates have so far been a legitimate mechanism for ensuring the entities' identities. Though the passwords can be a strong authentication mechanism, in practice they get misplaced or the password files' management is easily compromised.

While there is no doubt in the effectiveness of a digital certificate in guaranteeing the authentication, the main concern is to develop a mechanism that will be scalable in order to manage the increase of public digital certificate requests. This is because, for grid systems like GUISET, resource users may grow from time to time. If the authentication mechanism employed is not scalable, GUISET users may be unable to access the resources efficiently e.g. delay in certification acquisition. It has been conceded that the existing GSI has a drawback of poor

scalability which thus limits the number of sessions of a protocol that can be run by a user (Chen et al, 2005). This means a user can only be able to use a certain number of resource contributing sites.

This study, therefore, aims at improving the traditional PKI-based GSI solution by developing a secure certificate based mechanism which will enable scalability for the grid-based infrastructure, GUISET.

## 1.3 Research Questions

An investigation of the existing methods identified the following concerns which this research work will address:

1. How can we classify GUISET users?
2. Can this classification be used to formulate a mutual authentication framework for GUISET?
3. How can scalability be achieved in a PKI-based authentication infrastructure?

## 1.4 Goal and Objectives

### 1.4.1 Goal

The Goal of this research was to develop a scalable authentication framework for securing GUISET infrastructure.

### 1.4.2 Objectives

The above goal was formulated as the equivalent of some lower-level objectives, which are:

i. To design a scalable authentication framework for GUISET infrastructure based on the security requirements.
ii. To implement the proposed framework
iii. To evaluate the performance of the implemented framework.

## 1.5 Rationale of the study

This research work contributes to the GUISET, which is a grid-based infrastructure providing services to SMMEs. Due to the dynamic nature and heterogeneity of any grid environment, security (specifically the authentication aspect of it) still remains a challenge within these environments. The Grid Security Infrastructure (GSI) addressed security but poor scalability results in their current solution due to the extensive use of certificate. This degrades the performance of the whole grid system. We hope that our work will ensure accurate authentication while enabling scalability as the number of GUISET users may increase with time.

## 1.6 Research Methodology

The above research objectives were accomplished by using the following methodologies, namely literature survey, framework development, and proof of concept.

### 1.6.1 Literature Survey

This part of the research methodology included conducting an extensive survey of existing methods for security models in grid environments. During the survey, we specifically investigated the authentication aspect of security, seeking for the authentication mechanisms that

have been proposed and developed. The survey also engrossed authentication mechanisms which enable scalability, how they were developed, why and which design criteria was used when developing them. Knowledge acquired from this survey was used in the evaluation of existing authentication approaches for grid environments, and also for the identification of the metrics to use during the evaluation of the proposed model.

### 1.6.2. Framework development

Framework development needed an analytical approach to evaluating what has already been accomplished in the research for security in grid environments. After thorough investigation of the existing works, significant research results were scrutinized with the intention of identifying the strong points of existing methods. The knowledge achieved here provided the main insight needed for the target framework development.

### 1.6.3 Proof of concept

A prototype of the proposed authentication model for securing the GUISET infrastructure was implemented and evaluated as a proof of concept. Experiments were conducted to evaluate the performance of the proposed model. And an appropriate scalability performance parameter was used for evaluation.

## 1.7 Organization of the Dissertation

The dissertation is organized as follows:

**Chapter 1**: This consists of the introduction of the research work as a whole. It includes the overview of the research, the research questions, statement of the problem, rationale of the study, the goal, objectives and the methodology.

**Chapter 2**: The chapter consists of the background concepts in the area of grid computing, GUISET infrastructure, and grid security

**Chapter 3**: The chapter comprises of literature review on securing grid resources. This includes the concepts that form the foundation and concentrate on frameworks that have been done by different researchers, specifically looking at the gaps.

**Chapter 4**: Here we present the design and the development of a framework that secures the GUISET Infrastructure.

**Chapter 5**: The chapter describes the implementation layout and discusses the sets of experiments conducted.

**Chapter 6**: The final chapter draws the conclusion of the whole dissertation and the research's future work.

# CHAPTER TWO

# BACKGROUND

## 2.1 Introduction

Grid computing is a technology which allows diverse organizations to collaborate for the development and execution of computational resources to the benefit of all the engaged stake owners. The ability to share resources becomes a fundamental concept on grid systems like GUISET. GUISET technology is based on an open resource sharing paradigm and the provision of extensive power and information access to GUISET entities. Therefore, resource security becomes a primary concern since both resources and the users are distributed all over the network. And with the growing size of the GUISET infrastructure enabling more SMMEs to share resources; a comprehensive security solution, capable of responding to any attack on GUISET resources is required. This work, as mentioned in chapter 1, aims at developing a framework for securing GUISET infrastructure resources, while enabling scalability. Therefore, the chapter introduces the background concepts of grid computing security which form the basis of our research work.

In Section 2.2 we briefly discuss grid computing in general, demonstrating the grid architecture. This is followed by Section 2.3 where we elaborate on GUISET. We highlight Grid Computing Security concepts in Section 2.4 which include Authorization, Integrity, Confidentiality, and Authentication. We give a concise summary of this chapter in Section 2.5.

## 2.2 Grid Computing

It is an evolutionary technology which has been developed from existing technologies such as distributed computing, web services, and other virtualization technologies. Grid was initially designed to assist pervasive and distributed scientific computing that needs to use extensive quantities of data and computational resources that are spread in various autonomously managed networks. However, (Lazos et al, 2003) says that "use of grid computing has increased recently to constitute the deployment of grid technologies within the context of business". This considerably lengthens grid technologies' significance. Business services' standard interfaces have also been influenced by grid computing. Different areas that have been targeted by grid computing include finance, medicine, decision making, collaborative design, and utility computing (Vivas et al, 2009). The concentration is currently on coordinated resource sharing distributed across virtual organizations.

Grid computing leverages IT infrastructure to provide high throughput computing by using many networked computers to design virtual computer architectures. It offers the capability to vigorously tie the resources together for the execution of large-scale and distributed applications. These resources can include all elements of computing, such as: hardware, software applications, networking services, and pervasive devices (Fellestein et al, 2004). Grid computing technology enables mainly effective emerging business solution techniques. The difference between the grid computing technology and the traditional distributed computing is that, grid concentrates on large-scale pervasive resource sharing, virtual and pluggable high performance orientation.

This resource sharing is highly administered, with resource providers and consumers, giving description on what should be shared, who should share, and the specifications on which sharing

occur. A set of individuals and institutions define such sharing principles from what is referred to as virtual organizations (Foster et al, 2000). (Magoules et al, 2009) define virtualization as the combination of geographically distributed and heterogeneous systems.

The essence of the definitions above is captured by (Foster et al, 2002) by defining a simple checklist, in which he defines a grid as a system that*:*

> *Coordinates resources that are not subject to centralized control.* A Grid incorporates and coordinates resources and users that live within different control domains. It discusses the issues of security, policy, payment, membership, and so forth that arise in these settings.
>
> *Uses standard, open, general-purpose protocols and interface.* A Grid is developed from multi-purpose protocols and interfaces that discuss such fundamental issues as authentication, authorization, resource discovery, and resource access. It is important that these protocols and interfaces be standard and open.
>
> *Delivers nontrivial qualities of service*. A Grid enables its essential resources to be utilized in a coordinated fashion to produce distinct features of service, relating for example to response time, throughput, availability, and security, and/or co-allocation of multiple resource types to meet heterogeneous user demands, so that the effectiveness of the incorporated system is significantly greater than that of the sum of its parts.

There are conditions associated with resource sharing: resources should always be made available by the resource owners, accountable to constraints on when, where, and what can be done. A new technology is required for the development, administration, and utilization of the effective, cross-organizational VO sharing relationships. (Foster et al, 2001) forms the analysis of this technology in terms of a grid architecture, which classifies the basic system components, and demonstrates the interaction of these components with each other.

The grid architecture introduces a way categorizing the protocols: service, application programming interfaces, and software development kits according to their roles in enabling resource sharing (Foster, 2001). It portrays the interoperability between resource providers and resource requestors in creating the resource sharing.

The grid architecture can be viewed as a layered architecture, comprising of constituents in every layer sharing standard features but developed on abilities and behaviors attained from a lower layer. We show this architecture in figure 2.1 using the *hourglass model* (Foster, 2001).

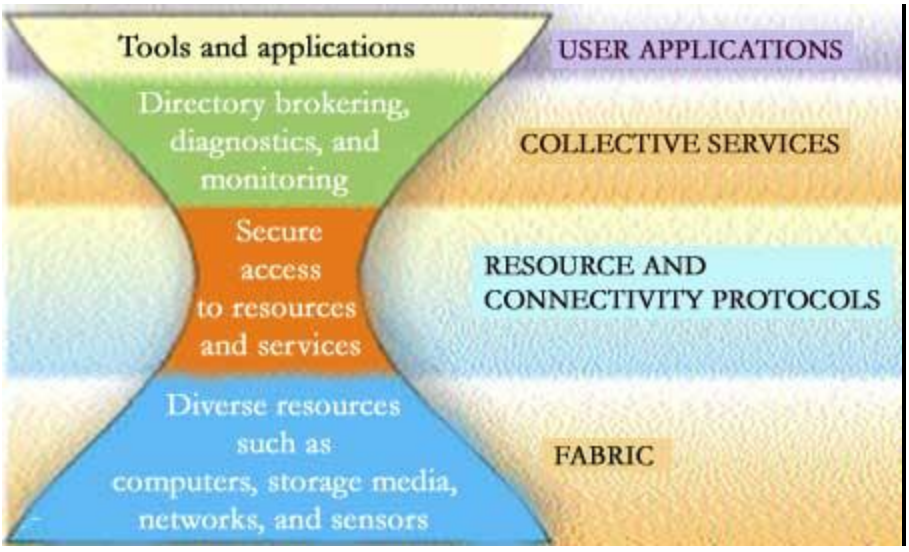


Figure 2.1The Hourglass model for Grid Architecture (Foster, 2001)

The Hourglass model for Grid Architecture has four layers: the fabric layer, the connectivity layer, the collective layer, and the application layer. These layers are described below

(i) *The Fabric Layer*: the fabric layer deals with the provision of resources with which grid protocols are responsible for resolving shared access. A resource may be a logical entity such as a distributed file system, computer cluster, distributed computer pool; in such instances, internal protocols may be incorporated when resources are being implemented. The local, resource-specific processes that occur on specific resources are implemented by the components found on this layer. And this results because of sharing operations at higher levels.

(ii) *Connectivity Layer*: the main communication and authentication protocols are defined in this layer. These protocols are needed for grid specific network transactions. The authentication protocols deal with the provision of cryptographically secure mechanisms which prove the illegitimacy of the users' identity and resources. The alteration of data among the fabric layer resources is achieved through the communication protocols. And lastly, the access resources are controlled by the resource layer, based on the connectivity and authentication protocols.

(iii) *Collective Layer*: scheduling services, data replication services, diagnostic/monitoring services, and directory brokering services are dealt with in this layer. These services only deal with apprehending interactions over resource collections and are not affiliated with a particular resource. The implementation of the collective components can thus result on the resources being shared, since

these collective components constitute exclusively on the resource and connectivity layer in the protocol hourglass.

(iv) *Application Layer*: user applications functioning within a VO environment are found in this layer. These applications are implemented in terms of, and by calling upon services defined at any layer.

## 2.3 Grid-based Utility Infrastructure for SMME enabled Technology (GUISET)

Grid computing has become the buzzword in both industry and academic community, and many grid infrastructures have been developed to address different ICT challenges. We introduce a grid based infrastructure which tackles the challenge experienced by the deep rural SMMEs. This infrastructure transparently networks the local business community of SMMEs and links them up to trading partners and third party services, thereby enhancing their business offerings. SMMEs play an important role in increasing jobs and promoting economic growth. However, they are overwhelmed with many challenges. These challenges include lack of e-commerce enabling technologies and/or limited understanding of e-commerce technologies, lack of sufficient information on market characteristics, and little or no information to inform decision making. Therefore, grid based solutions required are the ones with a simplified interface for end-users through knowledge management approaches.

Grid-based Utility Infrastructure for SMME Technology (GUISET) was proposed by (Adigun et al, 2006). GUISET is based on open-resource sharing paradigm and provides e-commerce solutions to SMMEs, particularly those in rural areas. This infrastructure effectively meets the business needs of rural SMMEs. It behaves logically utilizing knowledge-based approaches,

encompassing semantic-rich operation and interoperation. The abstraction of GUISET develops from the technology that is within the individual's means is already available; therefore a relevant method to achieve affordable technologies using utility approach to service delivery is rather required. The GUISET project motivation comes from the constant technological integration between grid and web services and the emerging Service Oriented Architectures. This creates a trend towards IT service provision utilities, and includes provision of computing data, information, and knowledge capabilities (Beco et al, 2006).

GUlSET is a three-tier layer architecture comprising the Grid-infrastructure layer, middleware layer and multi-modal interface layer (Figure 1.1). The Grid infrastructure layer provides services and resources, which are managed by the middleware infrastructure. It provides a service user with repository of business support services, which are accessible on demand with the aid of an interface. The middleware infrastructure is made up of the utility broker, which enables services to be consumed as utility services. The multimodal interface layer provides support for universal accessibility to all services irrespective of device type, user diversity and execution environments.

As discussed in Section 2.2, virtual organizations (VOs) are a group of geographically distributed individuals having permanent or temporary existence created to share resources and services among themselves. This resource/service sharing is ruled by a set of policies that define the conditions for that sharing. The dynamic nature of VOs makes the problem of implementing security in grids a challenging one.

Security has been a fundamental concern in grid technology from the onset, and has been considered to be the highly substantial defiance for grid computing. With the growing size of the

GUISET infrastructure, enabling more SMMEs to share resources; a comprehensive security solution, capable of responding to any attack on GUISET resources is required. We discuss more about security in grid computing in the next section.

## 2.4 Security in Grid Computing

Grid computing assists us overcome heterogeneity in terms of computing elements, operating systems, policy decisions and environments (Chakrabarti, 2007). Accordingly, resource sharing and virtualization are the main benefits of grid computing. In essence, grid computing exploits and combines remote systems from local personal computers to super computers to let end users share the resources and data across geographical and organizational boundaries (Ramakrishnan, 2004). Unfortunately, security issues become an obstacle when requiring the grid solution. This necessitates the development of solutions to overcome these impediments.

Security in a grid system is robustly connected to the notion of integrity, and such a system is one that we depend on to carry out its services (Vizienis et al, 2004). The characteristics of dependability involve availability, reliability, safety and maintainability. Nevertheless, confidentiality and integrity should also be considered when developing a trusted grid system. One way of looking at security in a grid system is to attempt protecting the grid services/data against security threats which are as follows:

(i) *Interception*- refers to the situation when an unauthorized individual gains access to a resource or service.

(ii) *Interruption*- refers to the situation when a resource becomes unavailable.

(iii) *Modification*- refers to the situation when there is an unauthorized tampering with a service so that it no longer adheres to its original specification.

(iv) *Fabrication*- refers to the situation in which additional data is generated that would normally not exist.

What is mostly recommended when developing a secure system is a security policy. A security policy specifically illustrates the actions that an individual is allowed to take and not to take within a grid infrastructure. This enables us to focus on the security mechanisms by which a policy can be implemented, and those mechanisms are:

(i) *Encryption*: changes data into something that cannot be understood by an intruder, presenting some alternative ways of implementing data confidentiality, allows us to check whether there has been any data modification and also provides integrity checks' support.

(ii) *Authentication*: used to verify whether the entities are whom they say they are.(we talk about authentication in section 2.5).

(iii) *Authorization*: ensures that an entity gets only those access rights to the resources in a system it is entitled to.

(iv) *Auditing*: provides tools which can be used to trace which clients accessed what, and in which way.

Grid comes with its own security challenges emanating from the fact that since the users and the shared resources may be coming from different domains, some of these may be malicious. There is thus a need to review widespread security areas that perform an important role in describing security in grids which are authorization, integrity, confidentiality, and authentication. These are discussed in the following subsections.

### 2.4.1 Authorization

Authorization identifies and controls the permissions of users accessing the resources and services. Through evaluating the identity of users and security challenges, and according to reasonable authorized strategies, authorization determines and controls which client can use what resources under what conditions. Simply, it specifies the details of who is allowed to access which resources and the conditions that an individual must comply with in order to access those resources. Authorization mechanisms include identity-based, role based, and rule based. These outlined mechanisms do not have the ability to fulfill the access requirements of distributed services on their own. This is because access relies on many other constituents like privacy requirements of the requesters, authentication requirements of the service, trust relationships with the request etc.

Authorization in grid environments is concluded as a result of assessing the request of an authenticated user against differing policies like privacy policy, trust policy etc. Fundamentally, the conclusions made for authorization are derived from the authorization data produced by authorities. For these authorities to make such conclusions, they should have a direct or a delegated relationship with either authorization subject (e.g., user or organization member to which the authorization is issued), or with the target resource for the request that invoked the authorization (e.g., owner or administrator of a resource), or with both. A trust mechanism based on some cryptographic method (i.e., by using some asymmetric or symmetric key mechanism) is thus used for executing these relationships, or they can also be executed completely off-line (e.g., by some other trusted delivery mechanism). This observation leads to the definition of the three basic high level entities involved in authorization which are subject, resource, and authority (Baker et al, 2004) and are described as follows:

(i) *Subject*: a subject is an entity that can request, receive, acquire, transfer, cancel, or entrust electronic authorization to exercise certain right. It may also be a process that performs on behalf of a user and as such holds access rights that were delegated to it from the user.

(ii) *Resource*: a resource is a system component which provides services or host services and places a set of rules and policies which are for accessing these services and these rules are presented by entities that an authoritative for that particular resource. Typical resources in Grid environments might be a computer providing compute cycles or data storage through a set of services it offers. A resource or some entity (a policy enforcement point, gateway) that is located between a resource and the requestor can itself enforce access to resources and thus protecting the resource from being accessed in an unauthorized fashion.

(iii) *Authority*: this is a managing entity responsible and legitimate for issuing, validating and revoking an electronic certificate such that the named subject of the granted electronic means is authorized to exercise a particular right or assert a particular attribute. Right(s) may be implicitly or explicitly present in the electronic proof. A set of policies may determine how authorizations are issued, verified, etc. based on the contractual relationships the Authority has established.

The authorization process involves messaging sequences of pull and push. We depict these sequences in the following subsections.

#### 2.4.1.1 The authorization push sequence

In the push sequence, an authorization is requested by the subject from the Authority. The Authority will then take an authorization decision and responds with a message secured (token or certificate), that operates as a proof of right if granted (Authorization Assertion). That proof of right granted comes with validity time window, and may consequently be utilized by the Subject to request a particular service by contacting the Resource. The authorization proof granted is accepted or rejected by the resource and this is reported back to the requesting Subject.

#### 2.4.1.2 The authorization pull sequence

With the pull sequence, the Resource is directly approached with a request by the subject. The Resource consults with its Authorization Authority first for admitting and denying the service request. The Authorization Authority then makes an authorization decision and sends the results in the form of a message. The service is thus granted or denied by the resource, which returns a result message to the subject. Examples of such systems are found in the network world with systems using the RADIUS protocol where requests typically are carried “in-band” (Rigney et al, 2000). In the Grid environment this sequence is implemented in the PERMIS authorization systems (Chadwick et al, 2002) and Akenti authorization systems (Thompson et al, 2003).

### 2.4.2 Integrity and Confidentiality

In a grid system, messages are used as a form of passing considerable amounts of data through networks. There are two types of attacks that content within the message is subject to, and those are replay and man-in-the-middle. Interruption and modification of messages occurs as a purpose

of altering the effects they have on their recipients. For the benefit of another party, messages may be used more than one time. The use of integrity and confidentiality mechanisms can prevent an eavesdropper from monitoring messages, whose aim was to get the data that should not be available.

Message integrity has two requirements. First, the data received must be the same as the data sent. In other words, data integrity systems must ensure that a message did not change in the transfer process. The second requirement for message integrity is that at any time in the future, it is possible to prove whether different copies of the same document are identical. For the verification of any alteration in the message content, digital signatures are used. A document is signed by the service requestor with the sender's private key and is sent with the message. The signature with the sender's public key is then used by the service provider to check if there has been any interference within the message.

Message integrity ensures that information that is being transmitted has not been altered. Secure transactions are the typical mechanisms that ensure that there has not been any message content's alteration when transferred from one point to another. This process is referred to as integrity and is attained by using algorithms and digest codes signed digitally. Confidentiality should also be achieved through these secure transactions. Confidentiality is the ability to ensure that messages and data are available only to those who are authorized to access them. Confidentiality can be achieved by making sure the connection between the participating parties cannot be intercepted. Though Standard SSL encryption enables point-to-point data privacy among service requestors and service providers, it may perhaps not be best suited for avoiding connection interception because in many cases, providers are not always the ultimate destination for the message. In the cases where a service provider acts a service requestor and sends portions of data to many

services, the XML encryption standard can be utilized to permit encryption of message portions. This can be achieved while enabling the header and other data to be clear text, thus encrypting the susceptible payload. Susceptible data can then be encrypted to the final destination, enabling true end-to-end data privacy.

### 2.4.3 Authentication

For a certain computer system to be able to offer a specific service to a certain entity, it requires identification and verification. We define identification as the process in which a requesting entity claims a certain identity, and verification is when that claim is confirmed. This identification and verification has been studied and illustrated in principle comprehensively (Burrows et al, 1992). Authentication is a process whereby an individual declares a set of credentials to a system. The purpose of authentication is to verify that the specific information presented represents a request to be authentic from a specified entity (Burrows et al, 1992). This is imperative, for verifying the identity of an entity as the basis for all future rights and privileges granted to the entity (Anderson et al, 1991).

What happens during an authentication process is that an individual that requires a certain service or resource in a grid system/ distributed system provides credentials. This normally consists of an account ID and additional information which may include postal address, email address etc. which confirms that the request is made by a rightful owner of the ID. This has been used for many years. A closely related alternative is the use of biometrics technology. This is based on the concept of representing who you are. Biometric attributes are fingerprints, retinal scan, pupil images etc. It is basically the same objective of presenting the unique information for identity confirmation. The advantage of biometrics as stated by (Conklin et al, 2004) is that, “for

most cases you don't leave home without them, and they cannot be forgotten". It is argued that the disadvantage of the biometrics involves their inability to allow transformation if need be. And they also cannot be used for all functions since they are not secret and cannot be used by other entities requiring authentication, human or not. What should be noted is that the existing extensive diversity of authentication schemes is caused by the decisions taken by individuals suitable to specific conditions/ requirements at the time of design of a specific system.

The necessity for differing levels of security is based on an evaluation of the threat associated with a particular system. Below we briefly analyze different types of authentication schemes which are: Shared secret/password based authentication, public key based authentication, and authentication based on a third party. These schemes have been used for several years to provide a secure but flexible way of identifying users in grid environments. The first scheme is maybe the most widespread one, where a shared secret password is used for authentication purposes. The second scheme is dynamic, however it may limit scalability. The third scheme, which uses Certification Authorities, is one of the most popular authentication mechanism used.

#### 2.4.3.1 Shared Secret/password based Authentication

Passwords have become the most recognized technique for authenticating entities requiring access to the resources. Nevertheless, when entities use password-based authentication, they may be exposed to attacks, for instance when a user generally select short and easily memorize passwords to access a resource on insecure communication channels like internet. In the meantime, composite passwords might be misplaced or stolen when written down and may put a strain on entities that might have to memorize them. (Conklin et al, 2004) proposed a conceptual model of password security across multiple systems connected by user activity, with which he

emphasized the result of user generated schemes to help in the users' management of IDs and passwords.

There are other issues involved in the password-based authentication. Let's consider a scenario in which two entities, X and Y, are trying to authenticate each other through a password protocol. An opponent can intercept messages between the entities and inject his own messages. In this man-in-the-middle attack, an opponent's goal is to play the role of X in the messages he sends to Y and the role of Y in the messages he sends to X. This reveals the vulnerability of password-based schemes though they are effective to some extent. To detail a system which uses the password setting as shown in figure 2.2, a user server token is entailed. The server token produces response towards a new password request for a user. The password is then generated by integrating a secret pass code made known to only the user with the token. The password setting system also involves a communication model which allows a token to be sent to a personal communication device owned by the user. The user integrates the secret pass code with the token as to accomplish a valid password that he /she can use to gain access to a secure system. In view of that, access to the system is based on nonsecret information known to the user; which may involve the user ID, pass code and the information granted to the user through an object infatuated by the user.

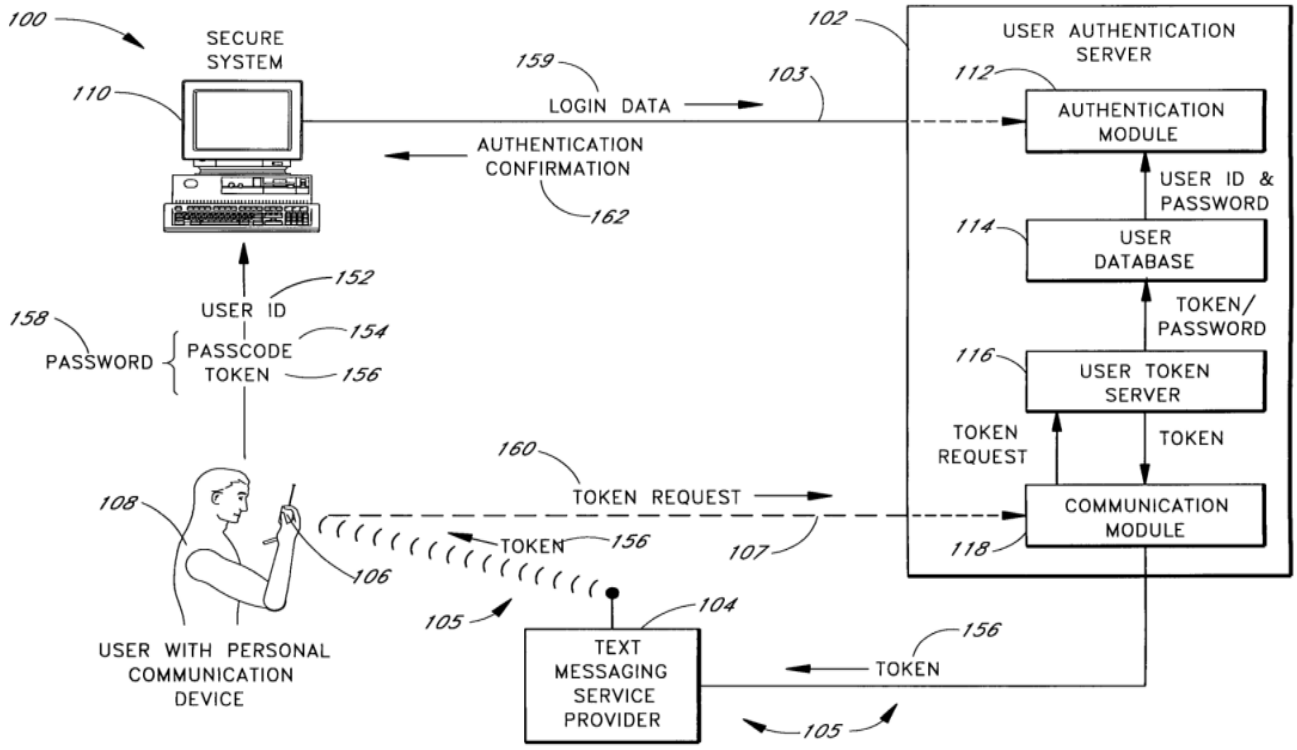


Figure 2.2 shared secret authentication (Chakrabarti, 2007)

#### 2.4.3.2 Public key based authentication

In public key based authentication, an entity contains both the public encryption key and the private decryption key. The public key is not stored as a secret like the private decryption key which is only known by the receiver. The public key and the corresponding private key are used to encrypt and decrypt the messages respectively. These keys correlate mathematically, however the private key cannot be formulated from the public key.

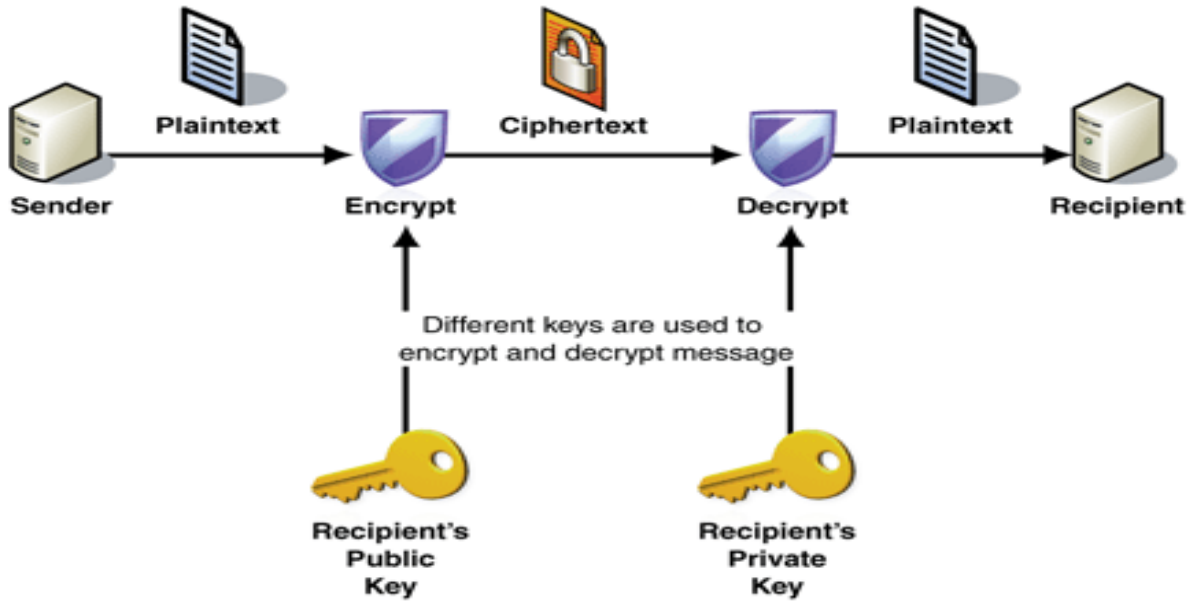


Figure 2.3 public key authentication (Chakrabarti, 2007)

We use an example (Haidar et al, 2009) to clearly depict a public key scenario. Consider A as a user with a public key *pk* and a private key *sk* respectively. Another user B requiring to send a message m to A, gets A's public key, uses some algorithm e.g. RSA to obtain the encryption c=RSA *pk*M and transmits c to A. to decrypt c, A applies the algorithm and obtains the original message RSA *sk* (RSA *pk*M) =M. this basically means that, when certain public key is utilized by a user for the encryption of a particular message, then the other user on the receiving end can only decrypt that message only if he has a corresponding private key.

A fundamental setback for public key authentication is verifying that a public key is dependable, and has not been altered with or substituted by a malicious third party. A common approach normally used to overcome this problem and of which we will be making use of this work is the public key infrastructure; which employs one or more third parties, known as Certificate Authorities for certifying ownership of key parts. We will discuss it in Section 2.4.3.3

#### 2.4.3.3 Authentication based on a third party

A good example of this sort of authentication is that of a person entering a new country. That person is ordered to present a passport and visa by the emigration office of the country. Particularly, the emigration office does not know the person, but believes some third party granting the passport and the embassy granting the visa. This is a typical case of third party authentication where the authenticator has no information about the user, however exploit a third party credential for authentication purposes.

The most common case is when Certificate Authority (CA) which acts as a third party grants a certificate to the user. The CA generates, publish, revoke, and archive the certificates. When the certificates are issued, they are then signed with a private key by the CA, thus ensuring that any alteration on the certificate is discovered. Since the public key of the CA is widely recognized, consequently the authenticator would not encounter any problem in validating the certificate and thus authenticating the user to access the system based on certificates. Nevertheless, such a system requires that every user provides a public key which can be confirmed by the Certificate Authority. This means that there is a need for Public Key Infrastructure (PKI) to make the above scheme work.

The PKI development guarantees mutual client-server authentication with the Secure Sockets Layer/ Transport Layer Security taking care of the server side authentication only. The PKI basically institute message integrity, user authentication and confidentiality before any vulnerable information is exchanged. The sender may use a private key to sign the messages digitally, and the recipient can check the signature using the associated public key constrained in the user's certificate issued by a Certificate Authority within the PKI (Schopf, 2002). This assists

in verifying to the recipient that the sender is who he claims to be since the digital signature has been verified by the CA and overcome any sort of 'man in the middle attacks'.

Conversely, PKI relies on trust and the users must protect the uniqueness of the private key (Hurley, 2003). Figure 2.4 depicts the security infrastructure with the Proxies and delegation, PKI and SSL.

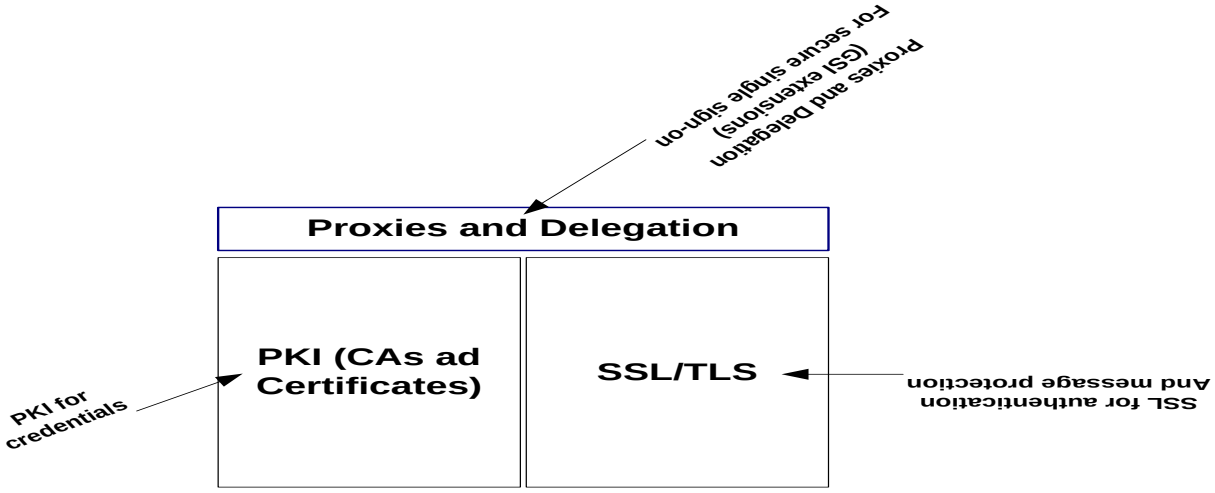


Figure 2.4 Grid security Infrastructure (Schopf, 2003)

SSL was extended to further incorporate self signed X.509 proxy certificates for single sign-on and delegation which encompasses similar functionality to that of Kerberos (The Globus Project, 2003). The development of the X.509 proxy certificate enables the user to dynamically assign anew X.509 identity to an entity and then delegate some subset of their rights to that identity, allowing new credentials and identities to be produced quickly. Thus, proxy certificates can represent the user in all authentication processes without the need for additional sign-on and delegate on behalf of the user (Raghunathan, 2005). Despite its strength, proxy certificates also have their weakness, e.g. "limited lifetime" of proxy certificates exist to secure communications

but time is wasted in generating proxy certificates each time it expires (Locks, 2002). Certificates are information about the an individual, hashed and then signed by the CA's private key (Chakrabarti, 2007).The main characteristic of a typical certificate, defined by (Haidar and Abdallah, 2009) consist of names of the subject and issuer, a public key correlated to the subject, a validity period, and an identifier for the cryptographic algorithms used by CA to sign the certificate; and another identifier for the public key algorithm with the public key on the certificate is used.

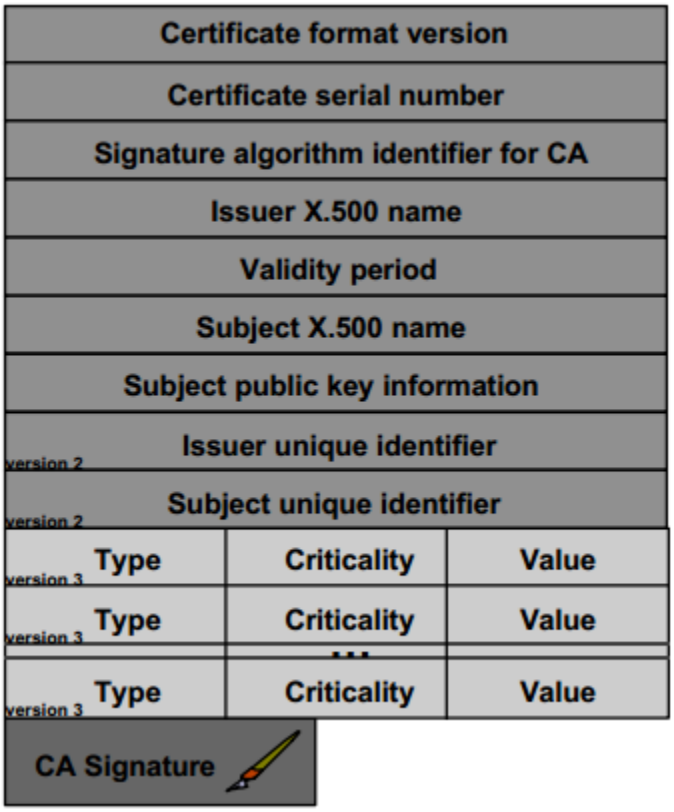


Figure 2.5 An X.509 certificate (Curry, 1996)

**a. Grid Security Infrastructure**

GSI is a set of protocols, libraries, and tools that allows users and applications to securely access grid resources. It is part of Globus Toolkits and is based on a Public Key

Infrastructure (highlighted in Section 2.4.4.3). GSI defines an entire architecture that provides the necessary functionalities for the implementation in grids, which have been developed to meet some of the specific needs of grids. These functionalities are: single-sign on, mutual authentication, delegation of privileges, uniform credentials, and scalability. And their description is as follows:

(i) **Single Sign-On**-A single sign on environment is defined as an authentication infrastructure that presents a single authentication authority for all authentication processes by giving out authentication processes to a specific infrastructure that manages authentication for all applications. (Clercq, 2002) defined it as "the ability for a user to authenticate once to a single authentication authority and then access other protected resources without re-authentication". A proxy certificate is created the first time the user authenticates. The proxy certificate maybe utilized to constantly authenticate for a period mentioned in the duration of the proxy certificate.

(ii) **Mutual Authentication**-Also called two-way authentication is method in which both entities in a communication link authenticate each other. In a grid environment, the client authenticates the server and the server authenticates the client. In this way, the grid users can be guaranteed that they are getting the grid resources from the legitimate entities and resource providers can be certain that all users are seeking to gain access for rightful purposes.

(iii)**Delegation of privileges**- Delegation is a process when a program is able to run on behalf of a user. Basically this means that the program is able to access the

resources on which the user is granted the rights of access. The program should be able to provisionally delegate a subset of its rights to another program.

(iv) **Uniform credentials**- As grid require inter-domain interaction; there must be a uniform way to represent credentials of the entities in the grid. GSI uses X.509 certificate format for representing the credentials of the entities in the grid.

(v) **Scalability**-It is predicted that a grid system agree to the development of flexible sharing relationships between grid users and resource providers through various collaborative structures such as client-server and peer-to-peer, along with more complex models like sharing via intermediaries or brokers (Foster and Kesselman, 2003). This means a grid system might grow from few resources to thousands. It thus raises issues concerning the performance degradation as the size of grid system increases. Scalability security then becomes a vital concern for such a scalable grid. A grid security system should therefore be able to handle practically a large quantity of users with the minimal performance degradation.

## 2.5 Summary

This chapter has presented the background for this research work. With GUISET being the grid enabled infrastructure permitting users to access and share resources, security has to intervene. We have introduced basic concepts including grid computing, security in grid computing, the grid architecture and the GUISET fundamentals. Since security is diverse, we focus on authentication in specific. In the next Chapter, we significantly analyze authentication and its mechanisms by exploring the existing works done previously.

# CHAPTER THREE

# LITERATURE REVIEW

## 3.1 Introduction

The most challenging task in grid systems is the provision of relevant security. Fundamentally, a grid environment should have security mechanisms. These mechanisms may include authentication, authorization, data encryption, and so on (Guo et al, 2005). Authentication is the basis of security in grid (Kent et al, 2007). Basically, authentication among two entities on remote grid nodes suggests that each party sets up a level of trust in the identity of the other party. The identity of an entity is normally some token or name that exclusively describes the entity. A secure communication channel is established by an authentication protocol among the authenticated parties. This is done to enable consecutive messages to be sent without the repetition of authentication steps.

There are many existing authentication technologies that have been proposed for identifying entities in grid environments. Identity based authentication is still the most widely recognized authentication mechanism. This is because it is easy to use and can be easily understood and implemented by end users (Crampton et al, 2007). Identity based authentication is regarded as one of the easiest, understandable and most suitable authentication method. Alternatively, identity based authentication protocols are very vulnerable to replay, password guessing and stolen-verifier attack (Lin et al, 2003).

Kerberos (Needham and Schroeder, 1978) enables the verification of the identities of principals on an open network. Needham and Schroeder declares that the verification of identities achieved

without depending on assertions by the host operating system, without trusting the host addresses, without requiring physical security of all the hosts on the network, and assuming that packets transmitted along the network can be read, altered, and inserted at will. The shared secret cryptography is used by Kerberos which acts as a trusted third party service to accomplish authentication under the above mentioned conditions.

Public Key Infrastructure (PKI) technology currently implemented in most grids is recognized as a sophisticated technology which is widely supported and that can be easily incorporated with different applications on different platforms (Lim, 2006). The PKI has challenges though which include difficulties of using certificates for unenlightened users, scalability due to the use of certificates, etc. In view of these challenges, the traditional PKI cannot satisfy all the requirements of a secure authentication system required by the GUISET infrastructure.

This chapter brings to light the outcome of the descriptive problem analysis phase of our research. Existing scholarships about the issues under discussion in this dissertation are reviewed. The authentication schemes to be discussed in this chapter are Identity-based authentication technology, Kerberos authentication technology, PKI-based authentication technology and hybrid-based authentication technology. The insight from this analysis will assist us to formulate our research approach. Subsequent sections of this chapter are arranged as follows: Section 3.2 discusses Identity-based authentication technology. Section 3.3 presents the Kerberos authentication technology. Section 3.4 reviews the PKI-based authentication technology. This is followed by Section 3.5 which contains the Hybrid-based authentication technology. We then conclude the chapter in Section 3.6

## 3.2 Identity based Authentication Technology

Shamir, 1984 tried to address the shortcoming of significant overheads when he presented an idea of identity based cryptography. Shamir implemented an instructive proof of concept of an identity based digital signature scheme using Rivest, Shamir, and Adelman (RSA) algorithm together with a one-way function. The RSA was somehow eliminated as a possible algorithm for the identity based encipherment, considering only the development of an encryption scheme for future research. In the conventional public key cryptography model, both public and the correlated seem arbitrary; differently in the ID based model, the public key pair is created from publicly verifiable data. Shamir denoted that, if the public key is developed from an identifier of the entity to whom the key is assigned, there would be no need for certificate usage and the administration of public keys in a fielded system would be simplified.

Until 2001 when Boneh and Franklin proposed a practical and proficient encryption scheme, the main drawback of ID based cryptography was that while Shamir and many others could propose different ID based signature schemes, no competent ID based encryption was known. An identity based encryption scheme was presented by Boneh and Franklin, 2001 established from the characteristics of the Weil and Tate pairings on elliptic curves. Their scheme emerged as the foremost fully functioning, competent and provably secure identity based encryption scheme (Paterson, 2004). This mechanism enables users' public-key to be a simply calculated function of his identity, whereas a Trusted Authority is used for other users' private keys' calculation. To be efficient, it is thus advisable to have an identity based signature scheme which can use the same underlying computational primitives and perhaps the same keys.

Lim and Robshaw, 2004 explored the idea of applying identity based cryptography (IBC) within a grid security infrastructure. However, conventional drawbacks of IBC such as escrow and the necessity to issues private keys through secure channels still holds back the dynamic use of identity-based keys. In addition, the proposals by Lim and Robshaw left out some of the necessary security requirements needed in the Globus Toolkit. These include the usage of proxy credentials for single sign-on and delegation.

Mao, 2004 revisited the GSI proposed for Globus Toolkit2 and made use of (Sakai et al's, 2000) non-interactive identity-based key distribution technique within the GSI authentication framework. This technique completely decreased the communication overheads among the two key sharing entities. However, this may not apply in the recent versions of the GT. A grid entity can always delegate its credential to a resource (or broker) which can act on the user's behalf.

Chen et al, 2005 also revisited the GSI in the Globus Toolkit V2 and proposed some improvements to the security architecture. Their work is closely related to the work by (Lim and Robshaw, 2005) where each entity contains a constant long-term credential that can be utilized by other entities to acquire dynamic public keys on-the-fly. However, Chen et al changed the security protocols implemented by (Foster et al, 1998) and greater performance resulted from the newly improved protocols. Additionally, an intriguing application was implemented for aggregate signature to save computational costs in confirming chained signatures. As with (Lim and Robshaw, 2005), however, for a dynamic public key to be computed and utilized, an entity should have an authentic certificate of the determined communicating party.

Afterward, Huang et al, in 2005 combined the works of Lim et al, 2005 and Mao, 2004, to develop an identity-based security infrastructure that appeared to be functioning somewhat contradictorily to the GSI. Although Huang et al, 2005 enlightened how credential delegation can be done among two entities, each run of their delegation protocol involved additional secure communication with private key generator.

Lim and Paterson, 2005 proposed a fully identity-based security infrastructure for grid applications using (Boneh et al, 2001) identity-based public cryptography (ID-PKC). In this method, credential management is easier because certificates are not used and key sizes are rather small. For example, when appropriate system parameters are used the communication bandwidth requirement for mutual authentication and delegation among two users can be decreased by up to 90%.

To enhance Lim and Paterson's work, (Crampton et al, 2007) proposed a password-enabled and certificate-free grid security infrastructure (PECF-GSI) only passwords can be used to authenticate entities. In this case authentication takes place among two entities and centralized authentication server. This approach is beneficial because both the client and the server are not needed when authenticating. The PECF-GSI proposal fully eliminated the requirement for long term user public keys, and consequently the requirement for a revocation mechanism for the public keys too. As an alternative users are given short-lived, identity-based credentials by authentication server upon successful authentication. All consequent security services are carried out using these credentials on behalf of users, without requiring direct user involvement.

(Hongweia et al, 2008) have proposed an identity-based authentication protocol for grid on the basis of the identity- based architecture for grid (IBAG) and corresponding encryption and

signature schemes. Commonly, grid authentication frameworks were attained by means of applying the standard SSL authentication protocol (SAP). Because of the difficulty in the authentication process, grid entities suffered complexities both in communication and computation. Being certificate-free, the authentication protocol coordinated well with the demands of grid computing. Simulation testing showed the authentication protocol was more lightweight and efficient than SAP. That contributed to the better grid scalability.

One identity based cryptography mechanism that has received considerable attention is by (Zhenga et al, 2008). These authors attempted to design a secure and efficient method for grid authentication by means of utilizing identity-based cryptography (IBC). An identity-based signature (IBS) was initially proposed for the generation of private key during authentication. Based on the theoretical analysis of the IBS's model and protocol, it can be concluded that the security and the efficiency of the grid authentication has been improved when compared to some IBC models.

The potential of the identity based technology to provide greater flexibility to entities within a security infrastructure and its certificate-free approach may well match with the dynamic qualities of grid environments. While ID-based models may not satisfy all the requirements as an authentication technology, they might prove to be a complementary technology (Nalla et al, 2003) by offering more lightweight and flexible key usage and management methods within the grid security infrastructures.

## 3.3 Kerberos Technology

**Kerberos** (Neuman et al, 1994), developed based on (Needham-Schroeder, 1978) key establishment protocol was an early implementation of authentication and authorization services using symmetric key techniques. (Neuman et al, 1994) disputed that the password-based authentication is not appropriate for use on computer network. Their argument was based on the fact that when passwords are transferred across the network, they can be intercepted and consequently be used by eavesdroppers to pose as the legitimate users. They advocated that stronger authentication methods should be based on cryptography. We depict the Kerberos authentication protocol in figure 3.1

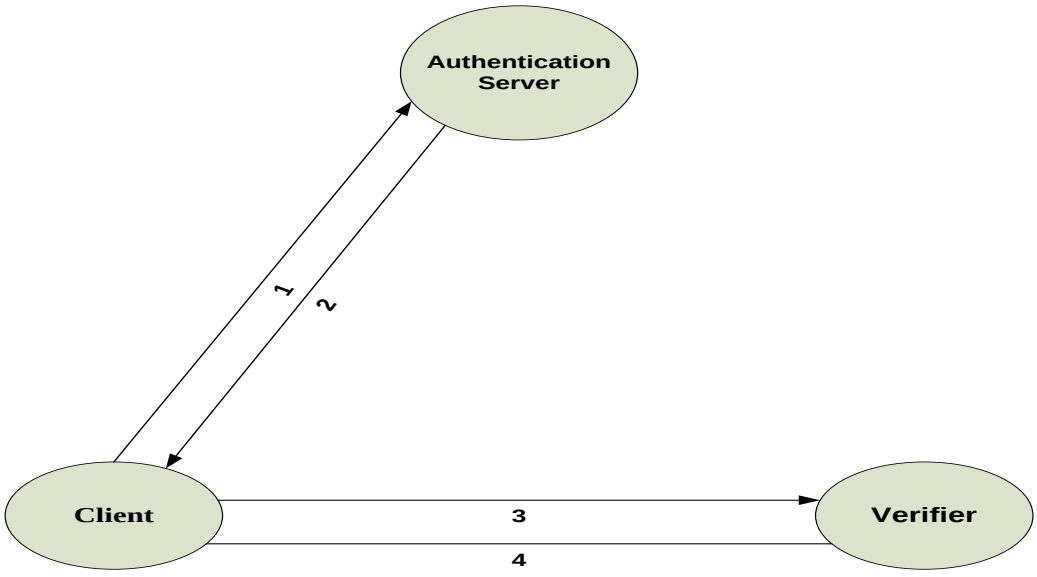


*Figure 3.1 Basic Kerberos authentication protocol (simplified) (Neuman et al, 1994)*

Briefly, the figure shows that when a client wants to gain access to a certain service, his identity must be established. This is done by presenting a ticket to the server, accompanied by the proof that the ticket has not been stolen but issued by the client. Three phases to authentication were outlined through Kerberos. The first phase involves a client obtaining credentials for gaining

access to services. The second phase involves a client requesting authentication for a particular service and the last phase involves the client presenting those credentials to the server.

To extend Kerberos, a variant of Kerberos Version 5 was proposed as Yaksha (Ganesan, 1995). Yaksha uses as its building blocks, a generalization of the RSA cryptosystem. The user's private key is split into two arts; one part is becoming the user's password and the other part the AS password for that particular user. Together, the user and the AS can digitally sign messages.

Though it is known that Kerberos is efficient (Crampton et al, 2007), it is recognized as an inflexible mechanism because it cannot develop interdomain trust relationships or create new entities without involving a specific site administrators (Welch et al, 2003) for a highly dynamic environment, like computational grid. Therefore, a Kerberised client side program called KX.509 assist the Kerberos grid projects in obtaining X.509 certificates utilizing a client's existing ticket (Kornievskaia et al, 2001).

## 3.4 PKI based Technology

Software, encryption technologies, and services that allow enterprises to protect their communications and business transactions on networks produce the Public key Infrastructure (PKI) based technology (Kuhn et al, 2001). An enterprise wide security architecture consisting of digital certificates, public key cryptography, and Certification Authorities is the foundation of a PKI. The main practices that a PKI includes are the issuance, renewal and revoking of digital certificates to grid users and servers. This PKI concept originated from the public key cryptography. Digital signature techniques exist from this technology and consist of particular characteristics that make it as the foundation for security properties in grid systems.

As mentioned in Section 3.1, PKI appears to be the main extensively utilized security infrastructure for grid developments. Globus Toolkit's Grid Security Infrastructure (GSI) was one of the early developments of a PKI-based security infrastructure for grid applications. GSI is discussed more in the next subsections.

Primarily, the PKI is certificate-based. The difference between the PKI and the identity based technology (discussed in Section 3.1) is the binding among the private/public keys and the individual (Lim et al, 2006). The only way this binding can be attained is by utilizing a certificate in the conventional PKI. In the identity based technology, the Trusted Authority (TA) is used for the management of the binding among the private key and the individual, and the public key is bound to the transmitted data. A public key derived from an identifier can be developed even before its corresponding private key is computed. In relation to the generation of the key, the traditional PKI enables a user of his CA to create public/private key pairs. nevertheless, the private keys in the identity-based structure can only be computed by the TA. This means that an identity based mechanism experiences an escrow facility which is not required. In the following subsections 3.4.1 and 3.4.2, we discuss the architectures that use the PKI based mechanism and then discuss the credential storages as they are used for securing the certificates employed in PKIs.

However, the most extensively accepted and applied grid authentication was on the basis of the public key infrastructure (PKI) and X.509 certificate. The PKI and X.509 certificate made the system to have lesser processing efficiency and poor anti-attack capability.

### 3.4.1 Architectures employing PKI based mechanism

Guaranteeing the authentication of the public key of the entities is a known obstacle within the actual use of a public key cryptosystem. For this to be attained, public key certificates are utilized to bind public key with their corresponding owner's identities using digital signatures computed by a trusted third party.

The first most notable security architecture for grids based on a third party was developed by (Foster et al, 1998). Foster et al developed the Grid Security Infrastructure (GSI) standard for security in grids as part of the Globus Toolkit (http://www.globus.org/toolkit). The GSI offered comprehensive security services. This was accomplished by employing public-key cryptography, cryptographic protocol methods and essential infrastructural supporting services in which public key authentication framework was the key component. Although this was a significant development for the provision of security services for distributed computing, the GSI still had shortcomings in implementing the grid feature of resource sharing.

The vital services that GSI has are: entity authentication, user authorization, message confidentiality, data integrity and non-repudiation. These were attained through innovative applications of standard public cryptographic solutions [(Foster et al 2002) and (Foster, 2001)]. For entity authentication, message confidentiality and data integrity, both resource and client side mechanisms were used by the GSI. GSI security mechanisms comprise X.509 credentials in the resource side, for identifying and assuring the resource. In the client side, these contain services to generate temporary credentials called a proxy, for performing single sign-on and delegation. Mutual authentication is performed by the client with the target resource using the certificates

and creates a secure communication channel by applying the Transport Layer Security (TLS) protocols (McDonald et al, 2003).

According to Laganier et al, 2005 security mechanisms used currently for securing grid infrastructures do not scale well with most cooperating domains and entities. This is due to the use of the public key infrastructure global to the grid environments, and a combination of global and local access control policies for making an authorization decision. They instead proposed an approach which combined the network and operating system virtualization with the Host Identity Protocol (HIP) and a simple public key infrastructure (SPKI) delegation/ authorization certificates. This allowed virtual trust domains to be implemented onto different shared computer nodes jointed by an untrusted network. This mechanism differed from the GSI on the implementation point of view. Functionally, infrastructures allow for the dynamic deployment of secure virtual organization overlays, while handling entity identification and rights delegation.

Computational grids are anticipated to present heavy throughput computing. It is therefore recommended that the grid security mechanisms implemented must be planned with constant concentration to performance and interoperability concerns, cautiously reducing their overhead. The provision of very high performance computational services by the grid based infrastructure is the result of the large number of resource contributing sites involved. Authentication mechanisms should thus be flexible in case there is growth in the number of grid resource users.
Certificate based public key infrastructures have been the basis of many of the fundamental advances in the evolution of security solution. It has been used in a variety of distributed applications ranging from e-commerce and web services applications to complex systems such as

grid computing and virtual organizations. Despite its widespread adoption, certificate based PKI still suffers from some uncertainty. Poor scalability is the main factor that has badly influenced the use of certificate based PKI authentication on a large scale.

Although application of X.509 PKI standard made GSI efficient, it still had a weakness. (Mao, 2004) identified a "weakness of poor scalability due to heavy interactions between user client and many contributing sites". In his work, the scalability problem is solved by applying a novel cryptographic method which allows authenticated session key sharing among two parties without any prior communication.

Lacceti et al, 2007 adopted certificate based PKI and PMI infrastructures. They introduced an architectural framework for authentication before accessing grid resources. Their architecture framework complied as much as possible with GSI security principles. The infrastructure was deduced from the effective Grid Security Infrastructure (GSI) and Certificate Security Protocol (CSP). Lacceti et al removes the disadvantage of scalability and expressiveness within the recent grid infrastructures by expanding the authentication and access control mechanism at the operating system layer. This can thus result on them being fully interoperable with the authentication frameworks used for the grid tiers.

As mentioned in previous chapters, our work acknowledges the certificate based GSI (PKI technology) features and develops an infrastructure deduced from (Foster, 1998). With the authors of GSI conceding that the existing grid method has a poor scalability and believe that the scalability problem develops from utilizing the standard X.509 certificate based PKI

authentication framework; our research work promises to solve this problem of scalability within a grid infrastructure. Certificate based public key infrastructure uses online credential storages to secure user's certificates and we shall discuss below.

### 3.4.2 Credential Storages

Administering certificates can be difficult and tedious for grid users. Several credential storages have been developed to store users' credentials thus securing them from malicious attacks. The SPX (Tardo et al, 1991) was developed as a server for keeping users long-term PKI credentials, encrypted with the user's passwords, and users authenticate to the SPX server to get their long-term credentials which they use for signing short-term credentials that are kept unencrypted on the local file system to be utilized for the rest of the session.

One main project, almost similar to the SPX and employed by the PKI-based GSI is the MyProxy (Novoty et al, 2001) credential storage. The initial development of the MyProxy online repository was to serve the purpose of delegating the grid credentials to trusted grid portals so that they can execute some authenticated operations on behalf of users without changing the standard web browser. To be precise, MyProxy is a virtual smart card development in the grid computing point of view. Individuals can keep their long-term credentials in MyProxy and get back short-term proxy credentials which enables single sign-on and delegation. The credentials are encrypted with a symmetric cryptographic key formulated from a password known only to that related entity. On the other hand, users can store their long-term credentials and delegate short-term proxy-credentials to MyProxy. Managing grid credentials within the MyProxy, becomes very convenient, enabling the entities to have rights of admission to a grid from diverse

locations easily. The difference between MyProxy and SPX was that the SPX included transfer keys for delegation, and MyProxy avoids transferring keys by adding a new proxy certificate.

Elaborated by (Lorch et al, 2004), users log in to the portal by entering a MyProxy server name, username, and password that the portal can utilize to get short-term proxy credentials for the user. As an alternative to accumulating long term user credentials on the web server, the MyProxy method utilizes a different, committed credential server for the security of the long-term credential aligned with the web server cooperation. It becomes normal to entities to sign on from machines which are not familiar with the need of credential mobility. This is so because of the broadening of the MyProxy to support the mobility and the renewal of grid credentials. In specific, the users need not to worry about replicating their long-term credentials to the different machines previously mentioned.

(Lorch et al, 2004) continues to articulate that MyProxy can be combined with the Certificate Authority such that new user credentials are generated by the CA and stored into the MyProxy storage with a predetermined passphrase. Since we are following the preceding approach, we basically are enabling the users to get their credentials without encountering the tedious process of managing their credentials. The MyProxy server then potentially handles and takes control of credentials since it keeps the long-term key and restricts clients/users from gaining access only to short-term credentials.

There are other approaches that enhance the credential storages. SHEMP (Marchesini, 2005) one of the credential storage which is basically a centralized repository based on MyProxy and benefits from the server-side secure hardware for the management of user credentials. Here, SHEMP server stores the users' long term credentials which thus secure them with hardware. In

addition, users can enquire about the presence and the security level of secure hardware from SHEMP.

Based on the MyProxy is the GAMA (Grid Account Management Architecture) project (Bhatia et al, 2005) which comprises of the back-end services and the set of portlets. The back-end services include the CACL, MyProxy, and CAS. Unlike our own system which is intended to have two of these components (MyProxy and Certificate Authority); this project implement the CACL for providing a development of a certificate authority that grants user and server certificates and these are used to provide mutual authentication as required by any grid system. The GAMA project also revealed the CAS for defining a set of functions and a set of admission rights for each function and lastly, the authors also use the MyProxy for centralized certificate storage with characteristics like certificate renewal. The set of portlets developed as the second component can be used to grant an interface for users to request an account, login and access web-based applications and to enable administrators efficiently determine policies and accomplish user management activities. This project is almost similar to our work but differs in that ours only bundles up on only the certificate and the MyProxy while GAMA bundles three components as mentioned previously.

Crampton et al, 2007 cited that GSI users are entailed to posse and handle long term credentials which are usually renewed yearly. Schulze et al, 2007 aligns with that statement and declares that; in many Public Key Infrastructures, the entities have the responsibility of securing their own credentials. This then enables the credentials to be kept either in the hardware based cryptographic devices or software-based repositories which is basically available in most web

browsers. The guarantee that the corresponding private key is accessed only by its owner during the lifetime of the certificate can only be conferred when the key is appropriately protected. Though it is a good idea that entities store their own credentials because they cannot deny if they have signed digital certificates (non-repudiation), it still comes with vulnerabilities:

i. Web browsers are not capable of issuing proxy certificates which are utilized in delegation and single sign on. For that reason, digital certificates used in GSI must be kept in a credential server.
ii. Support for mobile users- this is another constraint preventing the use of the software based cryptographic repositories within the browsers. In such manner, an entity would have to carry their credentials with them, which includes moving around with the credentials in different browsers, which can be weighty for most entities.

Crampton et al, 2007 states the management of credentials by equivalent grid users may result into some machines within the grid infrastructure lacking security in the vulnerability patches and virus exposition mode. And this may result into those systems being controlled by attackers who can thus take advantage of the vulnerabilities and hence gain access to long-term user credential. Credential storages are then assumed to be more secure for certificate users.

Moss et al, 2008 introduced a unified authentication framework for accessing heterogeneous web services, though in our case we are dealing with the grid services. But the main purpose or interest in his work is that the author proposes a credential storage and retrieval mechanism to store authentication data and pass that to the corresponding web services clients. That unified framework was implemented as a stand alone web service. It has a

common log in and authentication shared with SAVIOR (Liu et al, 2007) [their system which is the service oriented architecture for virtual organization infrastructure and resources].

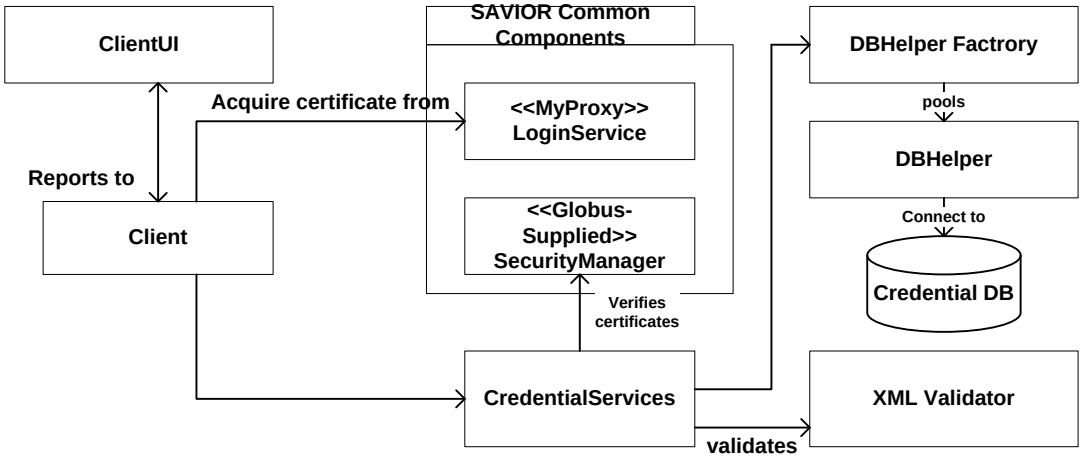


Figure 3.2 Simplified Class diagram for the core entities of an authentication framework (Moss et al, 2008)

The risk of attackers gaining unauthorized access to the database the storage and transmission of credentials are provided by a well-defined web service interface. These credentials are stored and transmitted as XML documents. The inspiration here is that the users log into the system once, receiving token that both identifies ad authenticate them. Then, when the user requires access to a secured resource, the token is presented to the system's credential service, which will then returns any access credentials that this user is authorized to use for that resource.

There are many other available commercial PKI credential storages, usually as an add-on to certification authority products for supporting credential mobility (Lorch et al, 2011). The nCiphernetHSM (www.ncipher.com/netsm) develops a network-attached hardware security module for storing private keys.

## 3.5 Hybrid based Authentication Technology

There have been a numerous number of works which have combined traditional PKI and ID based systems. (Chen et al, 2002) were among the first to propose a hybrid system which combines a conventional PKI with identity-based encryption system. Chen considered the issue of interoperation between entities in conventional PKI and entities in ID-based infrastructure. These schemes assumed the existence of hybrid scheme, but they have not discussed more in depth implementation issues like key escrow problem in ID-based system.

Certificate based cryptography and ID-based cryptography has been designed under different theoretical backgrounds and they have their own advantages and drawbacks, but as have been mentioned, there are works which try to provide them together in an efficient way. ID-based cryptography, issuing private keys to users in escrow-free way had been an important issue. (Lee et al, 2004) proposed a unique private key issuing protocol in the single-authority-multiple-observer (SAMO) model which can reduce the user authentication load a lot, but these schemes are subject to several attacks due to the lack of verifiable authentication of protocol messages Kwon et al, 2004.

In 2005, a hybrid mechanism combining identity-based methods at the user level and traditional PKI for supporting key administration above the user level was proposed (Lim and Robshaw, 2005). In this hybrid structure, a fixed parameter set through a standard X.509 certificate is issued by each user; this parameter set then allows users to act as their own Trusted Authorities with an aim to of allowing delegation and single sign-on. This framework solves the two issues of key escrow and distribution of private keys in IBC, but has a disadvantage of partially losing the original dynamic and lightweight qualities that IBC offers. This is because there is thus a

need for other parties' parameter sets to be authenticated and verified by the users before they can be used.

Wanga and Wanga, 2007 have developed a grid authentication mechanism, which was based on combined public key utilizing elliptic curve cryptography. Property analysis of the mechanism in comparison to the globus security infrastructure (GSI) authentications, revealed that CPK-based grid authentication, might be used as an optimized method towards competent and efficient grid authentication.

Chen et al, (2010) introduced a new concept called unified public key infrastructure (UPKI) in which both certificate-based and ID-based cryptography are provided to users in a highly unified manner. Chen et al assumes the existence of a trusted authority called *Key Generation* and *Certification Authority* (KGCA) who has the role of both CA and KG. After checking the identification information of a user, a certificate for a user-chosen public key X is then issued. In the proposed private key issuing protocol by Chen et-al, user is authenticated with certificate and user's certified public key X is used to blind the protocol messages such that only the legitimate user can retrieve the ID-based private key. Chen also shows that if interactions between end users are mainly executed using ID-based cryptography, then end users don't need to manage other end users certificates, which is a great efficiency gain than traditional PKI.

## 3.6 Summary

Security is a very critical issue and focuses on the provision of confidentiality of communication, the integrity of information and resources, and the privacy of the user data for large scale grid. (Jiancheng et al, 2007) states that "this large scale grid thus offer coordinated and controlled resource sharing with the capability of problem solving in dynamic, interdomain with diverse and heterogeneous computing environment". What mostly increases the apprehension about the security predicament is the synthesis of the grid and web service technology. This is certainly because of their intricate scenery and the mechanisms used for their implementation. This therefore means that the grid security concern is evolving into increasingly more important issue than before.

This chapter has presented some of the work that has previously been done with an attempt to resolve the research challenges brought about in this research work and the concerns that bring fourth the foundation for the solution to these challenges. We started by representing issues around existing technologies. We then presented PKI Technology in specific, analyzing the existing works around it. We discussed research issues around credential storages as part of this work, specifically outlining the frameworks that have used the credential storages before and we emphasize the problem of scalability which has not yet been overcome by the Grid Security Infrastructure we are adopting for this research work.

# CHAPTER FOUR

# DESIGN AND DEVELOPMENT OF A CERTIFICATE-BASED AUTHENTICATION FRAMEWORK

## 4.1 Introduction

GUISET has a group of distributed members and resources, therefore it can be considered as a virtual organization. A pool of resources or/and services that may be shared in the GUISET infrastructure include software applications, hardware components, computational power, and storage. With the increasing ease of sharing and collaboration, there is need for securing these shared resources. This chapter presents the certificate-based authentication model for securing the GUISET resources, thus enabling scalability, considering the expected growth in the use of GUISET. We initially explored the existing authentication frameworks with the aim of enhancing the one that mostly satisfies the GUISET infrastructure requirements.

Section 4.2 addresses the use case scenario which assists in demonstrating the GUISET design requirements. Section 4.3 outlines the GUISET Security requirements (Design Requirements), motivated by the use case scenario. Section 4.4 presents the certificate-based authentication framework and Section 4.5 concludes the chapter.

## 4.2 GUISET Use Case Scenario

Suppose we have a business process of Arts and Crafts that have been incubated in a rural-based environment like in Nongoma, operating their micro business-based on the GUISET Infrastructure. Basically in a GUISET, there would be many business processes grouped according to their associated goals and who are operating under a common domain. Prominently,

security threats would substantially exist at distribution, processing and sharing of resources. Before resources can be shared between business processes, there should be verification of the authenticity of the business process/resource requestor, this therefore, allows access to available resources only by an authorized business process. This is to say that if for example, the Nongoma Arts and Crafts business process (User A) wants to access a certain resource within the GUISET infrastructure, to avoid over-utilization, aggressive usage and loss of resources, User A would have to produce certain documents (credentials) to prove his identity.

When a certain business process sends a resource request to the GUISET administrator, the administrator would have to verify the signed request and would have to check if that business process (User A) has rights of access to the resources that the GUISET provides. What should be noted is that User A may also require that the administrator also proves his identity for the protection of the business process's accounts purpose or for the SMMEs information. What this may mean is that, when dealing with the distribution and access of resources among many business processes or organizations, the combination of security mechanisms and policies becomes a necessity.

In certain circumstances, User A may also have to delegate his credentials to the administrator so that the administrator can present the credentials on behalf of the User whenever the user requires access to other GUISET resources. The security of users' credentials is vital. Storage of the credentials is required and this storage can thus enable the administrator to easily verify the registered user which requires access to different resources. To be specific, data confidentiality and integrity are also very important requirements for safeguarding the business process's information. With the exchange of resources within the GUISET Infrastructure, the enforcement of trust relationships seems necessary within the cooperating business processes. Some business

processes may have long-term trust relationships with each other and some might contain short-term relationships, typically in hours or days.

Based on the description sent by the business process, the administrator may query a local replica to determine suitable resources for User A. When the job is completed, the administrator would inform User A and the results would be sent back to User A.

From the resource sharing aspects, trust establishment between these business processes plays a crucial part in grid security. Policy enforcement can rather be difficult, as articulating and substituting policies within the resource sharing business processes with different security mechanisms and access benefits. All these security objections must be taken into consideration when designing a robust GUISET security architecture.

The user population of the GUISET infrastructure is bound to increase with time e.g. business processes in need for resources may join the GUISET infrastructure. The population can thus be large and dynamic. Participants in such a virtual organization characterized by distributed collaboration will include individuals from different business processes. Given the possibility of increases in the number of credential requestors, scalability certainly becomes an important aspect when designing a security infrastructure.

With the rough idea of what security requirements might be anticipated in a GUISET environment, we would thus be able to bring together a more proper list of the GUISET security requirements regarded to be fundamental for sustaining a dynamic and distributed virtual organization. Below is the use case diagram for the scenario which thus provides an abstraction of the security requirements that may be expected of the GUISET security infrastructure.

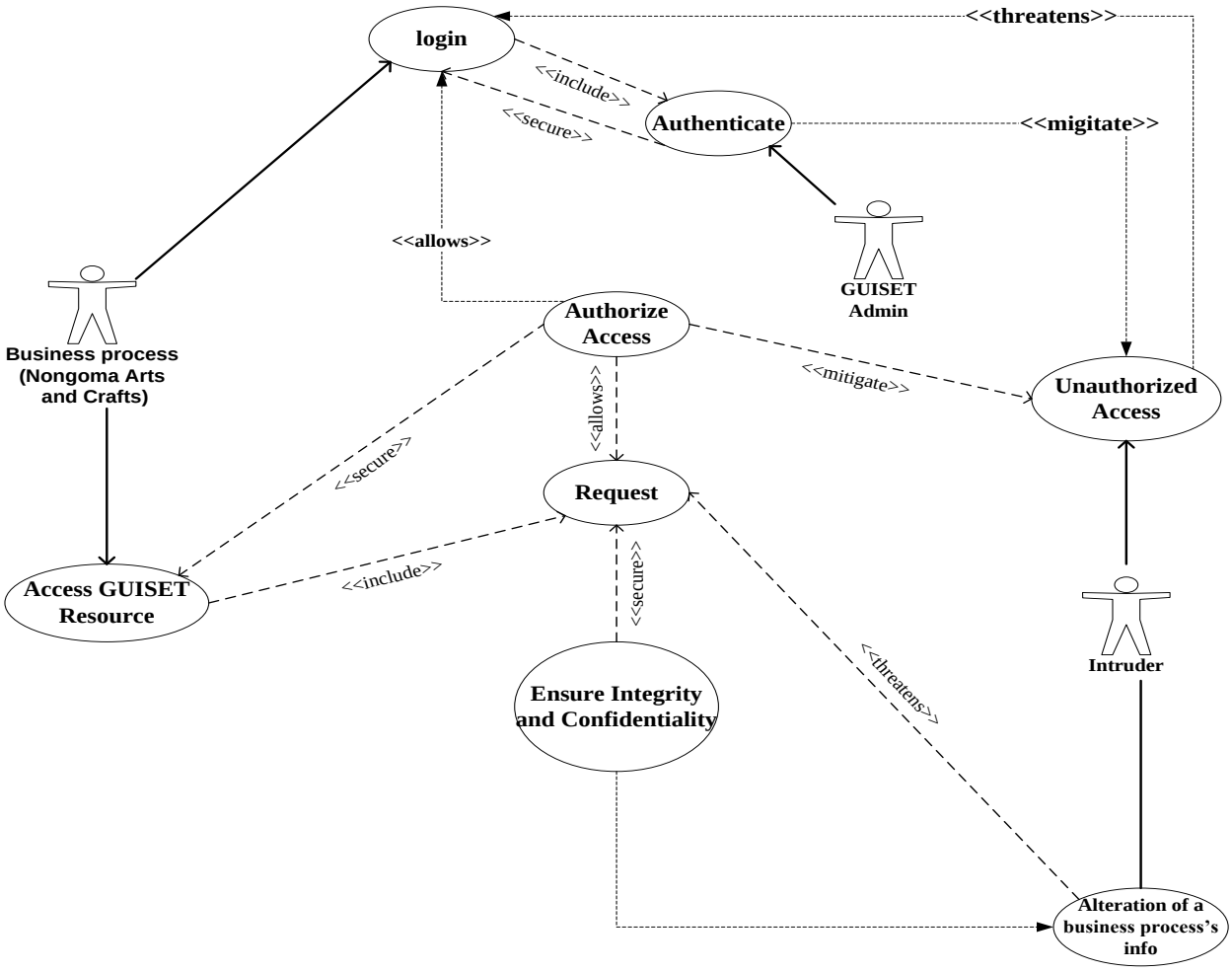


*Figure 4.1 The Use Case Diagram for GUISET security requirements*

The "authenticate" use case forms the basis of the authentication service for the application and is in control of securing the "login" use case, thus mitigating the unauthorized access use case which threatens the "login" use case. The "authorized access" use case forms the basis of the authorization service and is in charge of the "Access GUISET Resource" use case, for mitigating the unauthorized access use case and thus allowing the execution of "login" and "Request" use case. The "alteration of information" use case threatens the change of the information exchanged when a request is made to the GUISET system. This attack is avoided by the "ensure confidentiality and integrity" use case. The "Access GUISET Resource" use case

enables the business process to access the resource and is secured by the "Authorize Access" use case and includes the "Request" use case. The "Request" use case is also protected by the "ensure confidentiality and integrity" use case which mitigates the alteration of information that threatens the "request" use case.

From the use case scenario, we identify many issues that need to be considered. In this case, we refer to the individual's identities that may need to be protected, leaking of information when access to resources is granted etc. To clearly outline these threats, we classify them according to the actors and security vulnerabilities model during computation and outline the existing mitigating aspects which help us construct our own authentication model. Specifically, we utilize the threats classification by (Jiancheng et al, 2007) as the foundation of our appropriate design criteria and development of our own model.

Table 4.1 Threat analysis (Jiancheng et al, 2007)

| Classification | Threats | Mitigating Aspects |
|---|---|---|
| Resource Consumer Threats | • Credential theft<br>• Credential compromise<br>• Insufficient authentication credentials verification | • Lock-out<br>• Logging in credentials<br>• SSL/TLS client credentials<br>• MyProxy |
| Mediator Threats | • Network eavesdropping<br>• Man-in-the middle<br>• Brute force<br>• Replay<br>• Underlying transport protocol<br>• Masquerading | • Lockout<br>• Time-stamp<br>• Encryption<br>• Enforce access permissions |
| Resource Provider Threats | • Illegitimate use of physical resources<br>• Improper password/credential management<br>• Aggressive utilization of resources | • Secret keys<br>• X.509 certificates<br>• Passwords and IDs<br>• Time out approach<br>• Public Key Infrastructure (PKI) |

We exploit the above threat model to essentially categorize impeding attacker's objectives and competences. These aspiring intentions mostly involve user's credentials, SOAP messages, tangible and intangible assets. We use this threat model with an objective to illustrate and classify the range of different attacks comprehensively.

(i) **Resource Consumer (requestor) Threats**- essentially, it is recommended that when an end user needs access to a certain resource or service, precise name/password, credentials etc., should be provided to identify if a user is legitimate and do right actions limited by authorized access rights. If a consumer needs to utilize a certain resource within the GUISET infrastructure, there also should be credentials provided by that particular consumer to identify the legitimacy of the consumer and be granted access to those resources. If the credentials are not well protected, threats will be created.

(ii) **Resource provider threats**- Resource provider provides two kinds of resource including physical and software resource. Physical resource includes CPU cycles, network bandwidth, storage etc whilst software resource is comprised of user and system configuration, policy and audit data. Hence, resource provider threats contain such threats that are caused by illegitimately utilizing physical resources or by interfering or destroying the software resources.

(iii) **Mediator threats**- when resource consumers and providers are communicating, the mediator carries the equivalent data through service message in XML format. This basically means that the threats that would result here would be coming from insecure mediation mechanisms.

## 4.3 Design Requirements

The virtual organization (VO) is a fundamental concept in grid environments. It can be recognized as a temporary or permanent affiliation of geographically dispersed individuals or groups that integrates the resources, capabilities and information to achieve common objectives (Johnson et al, 2007). Depending on the context, dynamic collections of resources, services and individuals that has different kinds of VOs can be small or large, short or long lived, single or

multi-institutional. What should be noted is that security related concerns in any grid are determined by the necessity to support scalable dynamic distribution virtual organization (Foster, 2001).

Security has emerged as an essential and a long pervasive issue for all distributed systems, which in this case we concentrate in the GUISET system. GUISET is provisioned on top of a Grid infrastructure. Therefore, we cannot afford to assume that the security mechanism of the underlying infrastructure is sufficient, hence the need for appropriate mechanism to secure GUISET. Incorporating security mechanisms and policies become a requirement. Clearly, security concerns result from distribution, processing and sharing of resources.

From the literature review in Chapter 3 and the usage scenario in Section 4.2, we have identified the design requirements to take into consideration when designing an authentication framework to secure the GUISET resources. These basically deliver the GUISET security requirements and are as follows:

i. **Protect the accounts of each resource requestor-** GUISET wants to ensure that the information for every resource requestor and member account is protected so that one individual cannot access another individual's account. This requirement is mainly to ensure the privacy of everyone's account information. In most cases, companies hold data on behalf of individuals, and those individuals will eventually dictate who is allowed to see their data and for what purpose. Privacy is a growing topic and there are some mechanisms used to protect the privacy of individual's data.

ii. **Access limitation to individuals without credentials-** GUISET requires that all requests made are from authenticated users. Basically, what is needed is a security mechanism that

will ensure that resources are not compromised when GUISET is performing its functions.

iii. **Administrator control of critical functions-**GUISET wants to ensure that certain critical application functions are only controlled by the infrastructure itself.

iv. **Manageability-**GUISET wants to ensure that security is easy to manage in operational use. The security architecture should support a management framework for its components, users, resources, and enabling technology. The security architecture should also support delegated administration of security components.

**v.** **Scalability-** scalability describes the ability of a system to support variations in size of its workload without design changes. It is anticipated that any grid-enabled system allows the development of highly flexible sharing relationships within the user and resource providers because of various unified structures. A grid security system should be able to support a reasonable large numbers of users without degrading the performance of the system. Because the usages of the GUISET infrastructure may continue to grow, the clients want to be certain that the GUISET security mechanism would be able to handle large volumes of users requiring credentials to have access to resources, without any delays and inconveniences.

Following the above design criteria we have been able to provide a trusted authentication mechanism that will shield the GUISET resources accessed, providing higher security to the credentials of the users who has been granted access to those resources and thus enabling high scalability as the number of GUISET users requiring credentials for accessing the resources grow.

## 4.4 The certificate-based authentication framework

We have identified the need to achieve a mutual authentication framework for the GUISET infrastructure by firstly analyzing a possible scenario, thus disclosing the users of the infrastructure. Use case scenarios have become a widely used method for the elicitation of functional requirements (Srivatanakul et al, 2004) when constructing a security system. It is much easier to understand scenarios with only a confined introduction of their notation and best suited to the analysis of the requirements with the system's users. Therefore, by defining the behavior, actions and interaction of the GUISET users, we were able to define the security requirements needed for the GUISET security system designed.

Basically, the existence of security requirements is because of individuals with the negative factors that they generate, thus causing real threats to the systems. The use case scenario used in Section 4.2 has been used to explore the GUISET users, threats affecting those users and most importantly, the existing threats mitigating aspects. From the use case scenario, mitigating aspect and analyzing the requirements in Section 4.3, we came into a conclusion of adopting the PKI-based Grid Security Infrastructure of the Globus Toolkit (GT) proposed by (Foster et al, 1998). The PKI-based GSI uses certificates and proxy certificates for authentication. It is developed on the Generic Security Service Application Program Interface (GSS-API), which includes GSI-enabled OpenSSL (www.openssl.org) to support these proxy certificates. We thus use certificates as the form of identification as it has gained widespread use even on the Internet (Lock and Sommerville, 2002). In the past, many systems have depended on a simple password for authenticating, which is still used in many existing problems and this has led us onto using the certificate based authentication mechanism. Firstly, knowing a password is not full proof mechanism to authenticate an individual. In theory they can be a strong authentication

mechanism, but in practice, passwords get lost or the administration of passwords files can easily become compromised by user or external forces.

As may be expected, some machines within the extent of a grid community may lack up-to-date security in the form of the latest vulnerability patches and virus definitions. This may result into those machines falling under biased or the entire control of attackers who are capable of remotely developing vulnerabilities and hence gain access to long term user credentials. To improve accessibility of user's credentials and to increase protection, GSI uses an online credential repository which we are also going to use and is called MyProxy server. The Myproxy server stores user's long-term credentials which also enhances single sign on.

The PKI-based GSI has been present for a number of years. It is becoming increasingly apparent that the GSI may not be a perfect solution with regard to its poor scalability. With the extensive use of certificates, the Globus Toolkit GSI has been identified to be sufficiently secure for the small scale developments thus far. This may be challenging for grid systems like GUISET because the number of resource users may be expected to grow, and if the authentication mechanism is not scalable, GUISET users may not be able to access resources efficiently. This could specifically be because of the delay in certificate acquiring, thus degrading the performance of the system. Our framework then intends using the Grid Security Infrastructure, enabling scalability by using multiple Certification Authorities (CA) and Registration Authorities (RA). By using multiple CAs, the GUISET users would not be inconvenienced/delayed when acquiring certificates even when the number of certificates requests increases.

Figure 4.2 below is the standard Grid Security Infrastructure (Foster et al, 1998) from which we base the design our GUISET authentication framework.

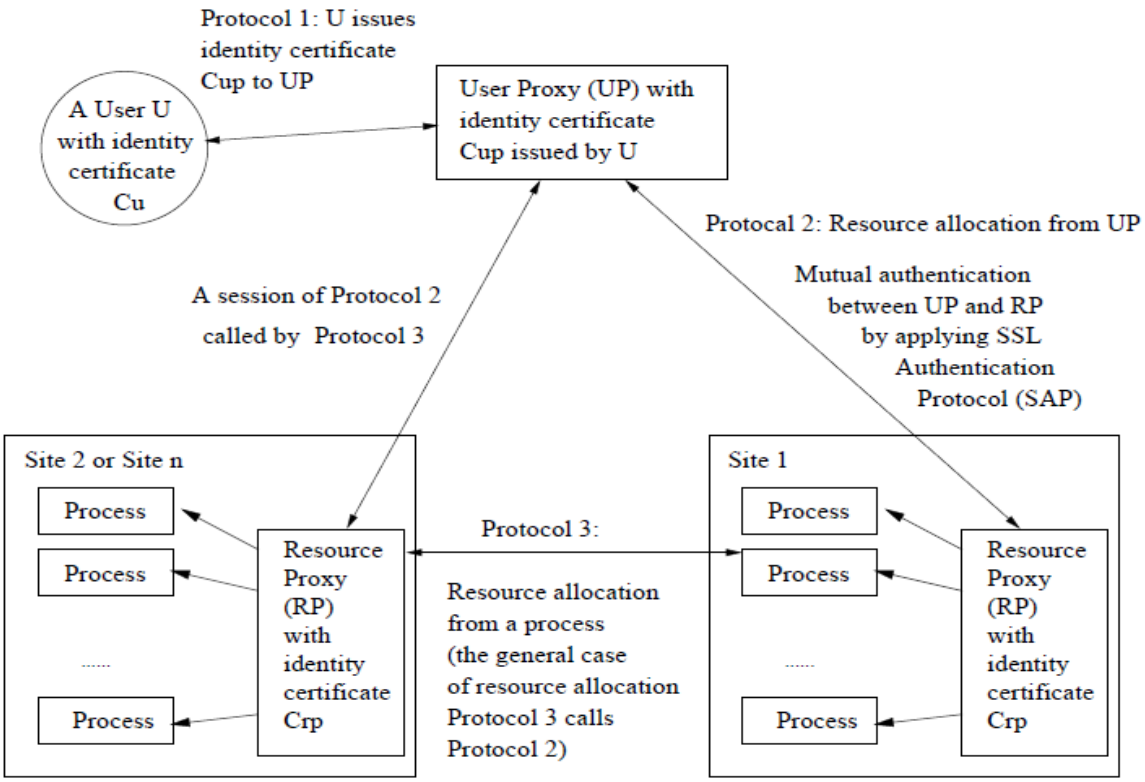


Figure 4.2 Grid Security Infrastructure (Foster et al, 1998)

From figure 4.2, we recognize how authentication is performed when an entity acquires a resource. The execution of authentication is performed by the user proxy who is a client machine of U acting on behalf of U. Entities U and RP have long term cryptographic credentials which are their X.509 identity certificates, denoted by $C_{U \text{ and }} C_{RP,}$ respectively. These certificates are issued by the Grid CA, but this figure does not show the PKI certification structure. In Figure 4.3, we illustrate the Grid Security Infrastructure in action by utilizing the mutual authentication and delegation protocols and also show the PKI certification structure.

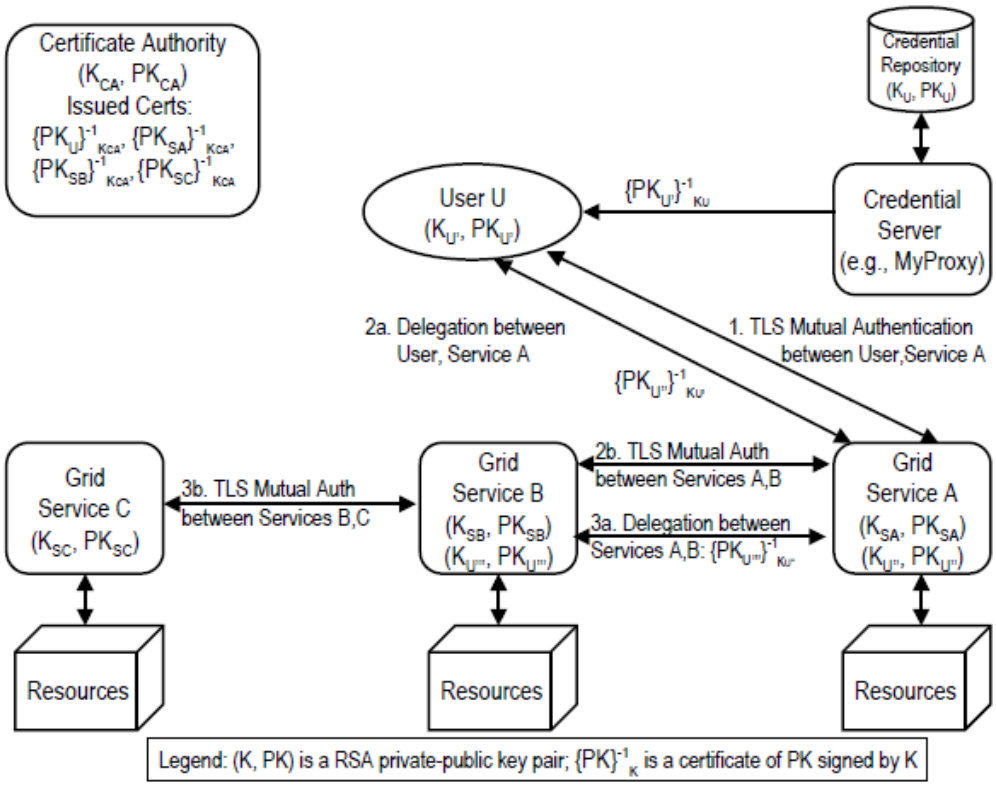


Figure 4.3 Grid Security Infrastructure with a PKI certification structure (Khurana et al, 2007)

In this framework (figure 4.3), the GSI shows delegation achieved through the use of proxy certificates and a proxy delegation protocol. This involves the generation of new key pairs and signing of certificate request with the new private key. Figure 4.3 also shows how mutual authentication is achieved and how the proxy certificates are obtained from the credential repository (MyProxy). They also specify that through the use of proxy certificates assigned by the Certification Authorities, single sign-on capabilities are attained. To be specific, for a user to request a resource, a proxy certificate must be created. This creation of a proxy certificate constitutes generating a new public/private key pair and signing the proxy with his long term private key. The newly created proxy certificate can then be used for repeated authentication

with other grid entities. The user's long-term private key does not need to be accessed again until the proxy certificate expires.

From these frameworks above (Figure 4.2 and 4.3), we acknowledge the vital features (single sign on and unattended use authentication) essential for the grid security solution and which the Grid Security Infrastructure has resolved satisfactorily by using the standard X.509. We examine the GSI authentication framework, and point out the weakness of poor scalability due to the certificate use. We then use multiple certification authorities to enable high performance of the system since the users would not have to wait longer periods for certification acquiring.

In Figure 4.4, we depict our authentication model comprised of several main components which are the building blocks in crafting our authentication model. **Certificate Authorities** which act as trusted third parties responsible for issuing the digital certificates to prove the identity of the resource requester and trusted by both the owner of the certificate and the party relying upon the certificate. The **Registration Authorities** for verifying that a certificate has a valid reason to have a digital certificate. The **GUISET administrator** with the role of intercepting the incoming and outgoing messages with the intention of checking whether the message has relevant security headers needed for authentication purposes (basically responsible for the overall functioning of the GUISET). We have the **Resource Consumer Proxy,** which is essentially a short lived agent produced by the resource consumer to execute security services on the consumer's behalf, and the **Resource Provider Proxy** is formed by the resource provider to assist administer a job submission from the resource consumer/requestor . The **Authentication server** facilitates authentication of the users attempting to access a resource. The **MyProxy credential repository** will be used for the management of the user credentials, helping the administrators to secure the users' private keys by offering an online service.

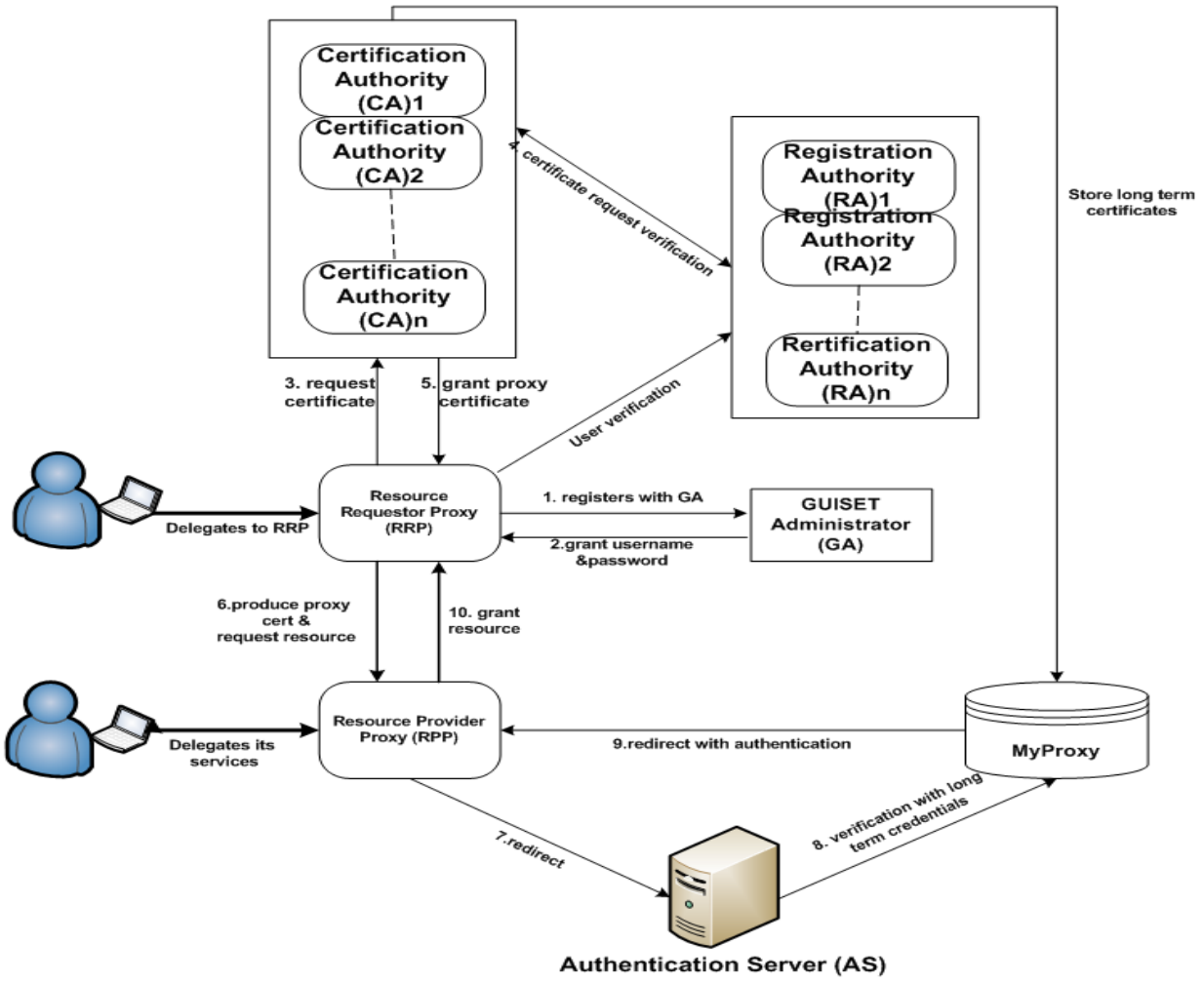


Figure 4.4 A Certificate-based Authentication Model for GUISET

Through the use of public key certificates, our model delivers some benefits and these include support for single sign-on, mutual authentication and delegation. The certificates are issued by multiple trusted Certification Authorities so as to access resources in GUISET where a user is authorized in particular, thus enhancing the performance by not inconveniencing users with delays in acquiring certificates. Our model sets some policies to control the access of resources. We integrated our model with the MyProxy credential storage. These thus facilitate the provision of a resilient authentication infrastructure. Our system then allows a user to access their

credentials from anywhere on the GUISET infrastructure and allows them to delegate credentials through the resource requestor/provider proxy

### 4.4.1 Model component interaction

An entity requesting a resource from the resource providers firstly needs to register with the GUISET Administrators (GA) which evaluates every requestor and administers the operation of the whole infrastructure. The resource requestor and the resource provider firstly delegate their responsibilities to the Resource Requestor Proxy (RRP) and Resource Provider Proxy (RPP), respectively. This is done so that the latter can act on behalf of the resource requestor and the provider. After the RRP has been registered with the GA and has been granted permission to request certificate, the RRP (on behalf of the resource requestor) sends a certificate request to one of the Registration Authority which basically provides an interface between the requestor and the Certification Authority. It then verifies that a certificate requestor has a valid reason to have a digital certificate then redirect the request to a corresponding CA. The use of multiple CAs is employed so as to enable scalability. The certificate request is then assigned to the most available CA (CA that does not have a long queue) for a response. In this way, the resource requestor would not have to wait longer periods before it can be granted a certificate and be able to access a resource. This is bound to be more efficient as the number of GUISET users increase.

After the RRP has been granted certificate, it then requests the resource from the RPP (acting on the behalf of the resource provider). The RPP passes the resource requestor's short term certificate and the request to the Authentication Server (AS) which offloads the responsibility from the provider by facilitating the authentication of the individuals attempting to access GUISET resources. It maintains a set of current registered users by checking if the short term

credentials from the requestors are valid. It confirms this kind of information from the MyProxy online credential storage which stores the requestors' long-term credentials. As may be recalled, the MyProxy enhances the single sign-on property and thus the requestor would not have to re-authenticate when he/she needs to access a different resource. After verification has been done on the credential storage, the MyProxy then redirects with authentication to the RPP. This basically means that MyProxy confirms that the user with a short-term proxy certificate claiming to have been registered with the CA is legitimate. The RRP then grants an acquired resource to the resource requestor.

### 4.4.2 Description of Components

#### Certification Authority (CA) component

This is the trusted third party responsible for accepting certificate requests from entities, grants certificates and maintains status data about the certificates. For this work, we assume that it grants digital certificates to requesting entities after a process of verifying their credential information. Certificates are not granted based on certification alone. There are a number of contributing members in the procedure, and there are several events that should occur before a certificate can be granted. The Certification Authority is only part of a greater network known as the public key infrastructure (PKI). The PKI grants security certificates, verifies credentials and assigns public key encryption. As a part of the agreement, the application is supplied by the Certification Authority to the registration authority, who will determine the validity of that information. If the requestor's data is verified, the process continues; and the public key is assigned for encryption purposes. This public key is obligated to the identity of the certificate holder upon the issuance by certification authority as a final measure.

To obtain access to the GUISET resources, the requestor firstly requests and obtain the security credential from the CA, which issues the certificates, with which in this case is considered because it enables for single sign-on, delegation and also enables mutual authentication.

This is a step-by-step process when a user is requesting a certificate from a CA:

(i) Key Generation: an entity requesting certification first generates key pairs of public and private keys.

(ii) Matching of Policy Information: The entity put together some additional information necessary for the CA to issue the certificate such as proof of identity, e-mail address, etc.

(iii) Sending of Public Keys and Information: The applicant sends the public keys and information (often encrypted using the CA's public key) to the CA.

(iv) Verification of Information: The CA applies their policy rules in order to confirm that the entity must be given a certificate.

(v) Certificate Creation: The CA constructs a digital document with the appropriate information (public keys, expiration date etc.) and signs it using the CA's private key.

(vi) Sending the Certificate: The CA may then send the certificate to the entity.

**Registration Authority (RA) Component**

To verify that a certificate requestor has a valid reason to have a digital certificate, we used the Registration Authority component. The RA implements an interface between the resource requestor and the CA. The authenticity of the user is verified and the certificate request is sent to the CA. The quality of this authentication process basically identifies the level of trust that can be placed in the certificates e.g. if an RA needs an e-mail address and a name before it grants a certificate, the level of trust that should be placed in that certificate would be considerably lower

than if more strict registration procedures were needed. What should be noted is that, an RA can offload many tasks from the CA, but it can never be the issuer of the certificate.

**The MyProxy credential repository component**

It has been recognized that, for any security system to be trustworthiness, resource requestors' credentials need to be protected at all times from the attackers who may want to steal them. Credentials can be protected using different approaches. The easier mechanism can be storing the credentials in a file with restricted access or else on a hardware token, normally protected with a pin. These alternatives are highly competent, but may be very expensive because hardware support for using the token may be needed.

We employ the MyProxy as used in the PKI based GSI. MyProxy is a web-based grid portal that has been designed as online credential storage for storing resource requestors' long-term credentials. It is implanted using the Generic Security Services Application Programing Interface (GSSAPI) Library, offered by the Globus Toolkit which employs the OpenSSL Library for Transport Layer Security protocol implementation and X.509 certificate handling.

The core protocol for the MyProxy system between a user and the MyProxy Server is as follows

i. "The user establishes a TCP connection to the server and initiates a server-authenticated TLS handshake protocol. A full TLS handshake is not mandatory as in most cases; a user does not have an existing X.509 credential.
ii. Once the TLS handshake is complete and a secure channel is established, the user sends a request message to the server. The request contains the protocol version, the command (e.g. retrieve, store, or remove a proxy credential), a username, a passphrase (an ASCII password used to protect the stored proxy credential) and a lifetime.

iii. If all checks succeed, the server will return `0' to indicate success or `1' with an error text that suggests otherwise.

(J. Basney, M. Humphrey, and V. Welch, 2005)

**Resource Requestor Proxy**

The resource requestor proxy acts as a session manager process that is given consent to operate on behalf of a user for a certain period of time. We assume that once the requestor proxy has been implemented, the user may be disconnected in order to abolish the necessity to have the requestor credentials accessible for every security operation. This lessens the possibility of the credentials being compromised during operations.

**Resource Provider Proxy**

The resource provider proxy is an agent that translates between inter-domain security operations and intra-domain mechanisms. It is allocated by user proxy, and is used for scheduling the access to a resource and for mapping a computation on that resource.

**Resource Requestor**

In our work, a resource requestor is either a service or simply an application that needs to use or gain access to the GUISET resources for the purpose of executing, completing and achieving a certain job. As mentioned in the chapter's introduction, we assume our resource consumers are the Small, Micro and Medium Enterprises clusters which in this case we need to authenticate before they can gain access to the resources.

**Resource Provider**

The resource providers register its resources with the resource provider proxy with the aim of enabling the proper mapping of requestor's demand to the matching resources. the registration of the identity by the providers to the proxy also help enable the mutual authentication so that the requestors will know that the resources granted to them are exactly what they requested, and that the personal information they provide when registering with the resource requestor proxy is secured.

**Authentication Server (AS)**

To facilitate the authentication of the individuals attempting to access GUISET resources, we use the authentication server. It maintains a set of current registered users. The AS offloads some responsibility from the resource provider by making sure that the individuals claiming to have been authenticated are legitimate. In essence, it verifies with the MyProxy repository if the requestor has authenticated with one of the CAs by accepting the request redirected from the resource provider and checking its long term credential on the repository.

**GUISET Administrator (GA)**

Every individual that needs to access a resource within a GUISET environment has to register with the administrator before he or she can even request a certificate from the certificate authorities. The GUISET administrator is a GUISET-aware individual/service with the responsibility for the overall functioning of the GUISET. The GA evaluates every individual and accordingly maps it to a specific set of privileges.

### 4.4.3 Authentication Algorithm and the Functions

The function used to grant users (resource requestors) the certificates is as follows:

Let x(y) = authentication function

y = the client to be authenticated

V = $\{v_i\}$, *i=0, 1, 2, ....n,* are requestor's attributes from the certification request process

P = $\{p_i\}$, *i=0, 1, 2, ....n,* are attributes on user's certificates

$$x(y) = \sum_{i=0}^{n} v_i - p_i \begin{cases} accept\ if\ x(y) = 0 \\ reject, otherwise \end{cases} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

If the requestor attributes e.g. password, username etc., are similar to the attributes of the registration process, then CA accepts the user as an authorized user and issues the certificates to the requestor, otherwise the request would be rejected.

Let f(x) = certificate function

A = $\{a_i\}$ *i=0, 1, 2, ....n,* are attributes of requestor's proxy certificate

B = $\{b_i\}$ *i=0, 1, 2, ....n,* are attributes of a corresponding long term credential

$$f(x) = \sum_{i=0}^{n} b_i - a_i \begin{cases} accept\ if\ f(x) = 0 \\ reject, otherwise \end{cases} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

After receiving certificates from the CA, requestor provides the proxy certificate to the resource provider which redirects it to the authentication server. The server checks with the credential storage if the information on the proxy certificate matches with the information on the

corresponding long term credential. If so, the requestor is granted access to resources and if not, access is denied.

a : requestor information

b : information of the requestor certificate

c : information allowed by Certificate Authority

d : information on the MyProxy credential storage

e : certificate authority (CA)1

f : certificate authority (CA)n

**1**.Request Certificate from CAs

( If a=b, then

Create certificate

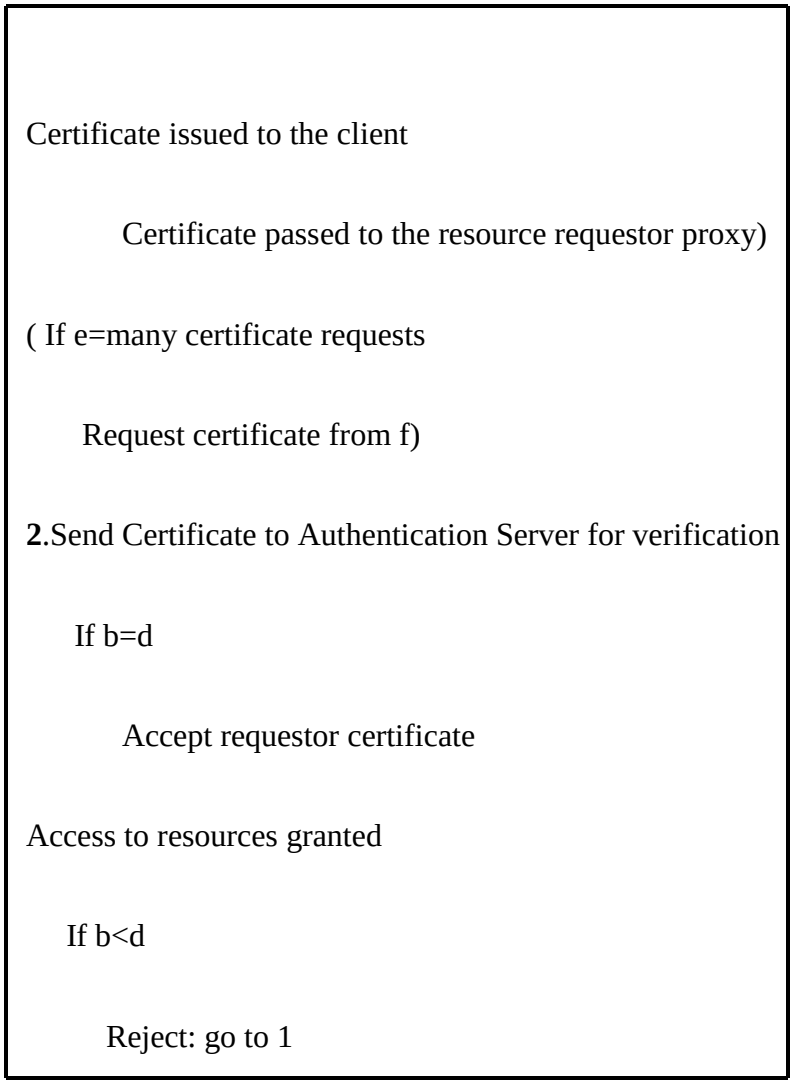


*Figure 4.5 A procedure for certificate-based authentication for GUISET infrastructure*

The above figure illustrates the procedure for the authentication framework. It exemplifies the certificate request from one of the CAs and as may have been mentioned that, if a single CA has many certificate requests, an alternative one is used. With the resource requestor information meeting the CA policies, a requestor is granted a certificate, which is thus verified by an

authentication server by checking with the information on the MyProxy credential storage. And if the information is guaranteed, the resource requestor is granted access to the resource.

## 4.5 Chapter Summary

In this chapter, we have described the design of a certificate-based authentication framework for a grid-based environment, GUISET. We have been able to add on the Grid Security Infrastructure adopted, by ensuring that the GUISET authentication infrastructure enables scalability. As may have been mentioned in the preceding sections, this work aims to develop a framework for authenticating the GUISET users before accessing the GUISET resources, thus allowing scalability as the number of the GUISET users increases concerning the usage of the infrastructure. This has been moderately accomplished in this chapter with a thorough depiction of the model design, algorithm, sequence diagrams and flow of events diagrams.

# CHAPTER FIVE

# IMPLEMENTATION AND PERFORMANCE EVALUATION

## 5.1 Introduction

In the previous chapter, we presented the certificate based authentication framework. This framework allows GUISET users to be authenticated using certificates before they can have access to GUISET resources. In this research work, the main focus is on the scalability of the authentication framework since the users of the infrastructure may increase with time. In order to prove the concept being discussed in the dissertation, this chapter presents the implementation of the model the results obtained.

In demonstrating the performance and the behavior of the model, we present the assumptions considered in Section 5.2. We provide the implementation details in Section 5.3. Experimental setup is presented in Section 5.4. And we present the Experimental results in Section 5.5. This is followed by Section 5.6 which presents the discussion of results. And lastly we conclude the chapter with Section 5.7.

## 5.2 Implementation Assumptions

The following assumptions were made during the implementation experiment:

1. The time spent transmitting data on the wire is considered negligible.
2. There is no delay encountered by users when acquiring certificates, the Certificate Authority is always available. We neglect the aspect that certificate requesters might have to wait until the normal working hours in order for them to receive their certificates.

## 5.3 Design of the Implementation

In this section we present the prototype design of our model. We present the UML models of the certificate based authentication framework developed in our work as a solution to the issue of scalability in existing certificate based authentication frameworks. We begin by presenting the use case model, followed by the sequence diagram, and lastly the activity diagram.

### 5.3.1 Use case Modeling

Based on the design considerations presented in Section 4.3 in the preceding chapter, a use case diagram is selected as the mechanism through which our certificate-based authentication framework prototype can be evaluated. Figure 5.1 represents the use case diagram for the prototype. There are three actors which are certificate requester, web service client and Certification Authority.

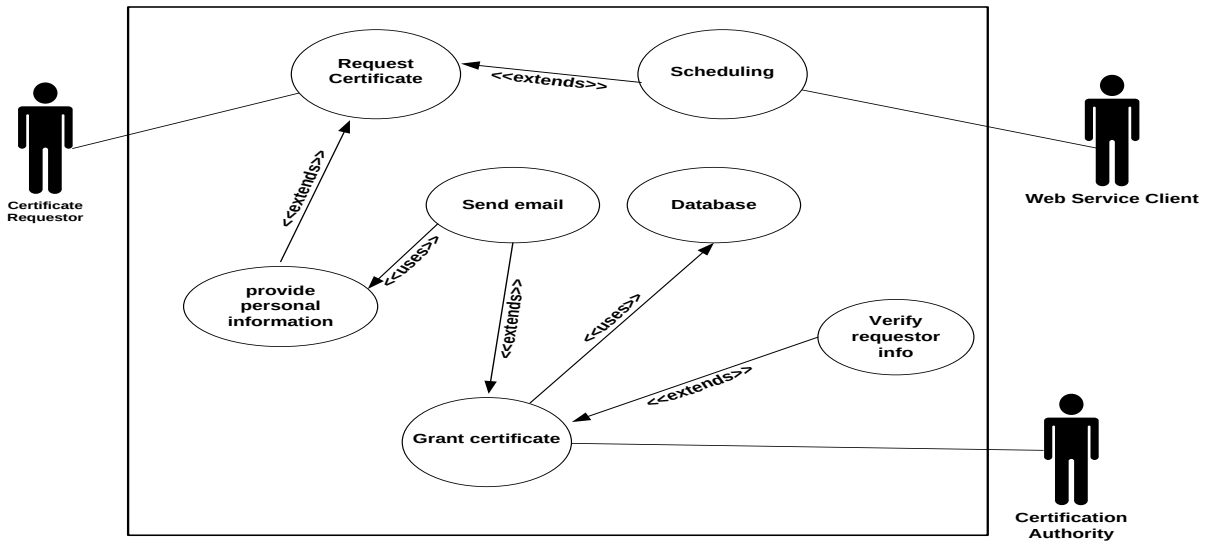


*Figure 5.1 Use Case Diagram*

What happens is that, when the certificate requester requests for a certificate, he or she provides his or her personal information to the web service client. The web service client acts as a broker

(scheduler) and verifies which Certification Authority is available among the 10 deployed Certification Authorities and directs the request to that available Certification Authority. The web service client passes the requester's information through SOAP messages. The Certification Authority verifies the information, confirming if it meets its policies then grants the certificate by sending an email to the requester. The Certification Authority also notifies the web service client that the certificate has been granted to the requester. All the information is then stored in the database.

**Certificate Requester**

The Certificate Requester acts as an entity which requires access to the GUISET resources and has to get a pass phrase from a Certification Authority first. The requester requests the certificate from the Certification Authority through the web service client. It sends its basic information which includes name, address, email etc to the web service client and waits for the response from the Certification Authority.

**Web Service Client**

The Web Service Client also referred to as the WebTester, acts as a scheduler and can also resembles the broker between the certificate requester and the Certification Authority. When the requester needs a certificate, it communicates with the web service client. With more than one CA implemented, the web service client has to check the most available CA to process the certificate request. The web service client then passes all the information of the requester to that available CA.

**Certification Authority**

The Certification Authority acts as a trusted third party that accepts certificate applications from the requesters, grant the certificates, and maintains the status data about the certificates granted. In this case there are 10 CAs implemented and each is denoted by a certain city (e.g. Durban, Cape Town etc.). The Certification Authority here does not get the request directly from the CA but receives it from the web service client described above. The CA processes the request and grants the certificate to the appropriate requester, but only if the requester's information meets the CA's policies/requirements. The certificate is sent via email directly from the CA to the requester, and a notification is sent to the web service client that the requester has obtained the requested certificate.

**Database**

The database component in this model acts as a storage for all the requester's information sent by the web service client to the Certification Authority. And when the Certification Authority grants a short-term certificate to the requester, it uses the database to store the long-term certificate which corresponds to the granted certificate. The long-term certificate contains all the information about the sent certificate. This assists in terms of the certificate holder having to sign-on again when in need to gain access to another resource. The certificate holder then does not sign on again, but that stored information is used.

### 5.3.2 Sequence Diagram Modeling

Figure 5.2 shows the sequence of the flow of messages among the components and actors of our model. The diagram clearly illustrates the role played by each component in the formation, the

issuance of certificates, as well as in the management of the entities' credentials provided by the Certification Authorities; those credentials used when requesting access to resources.

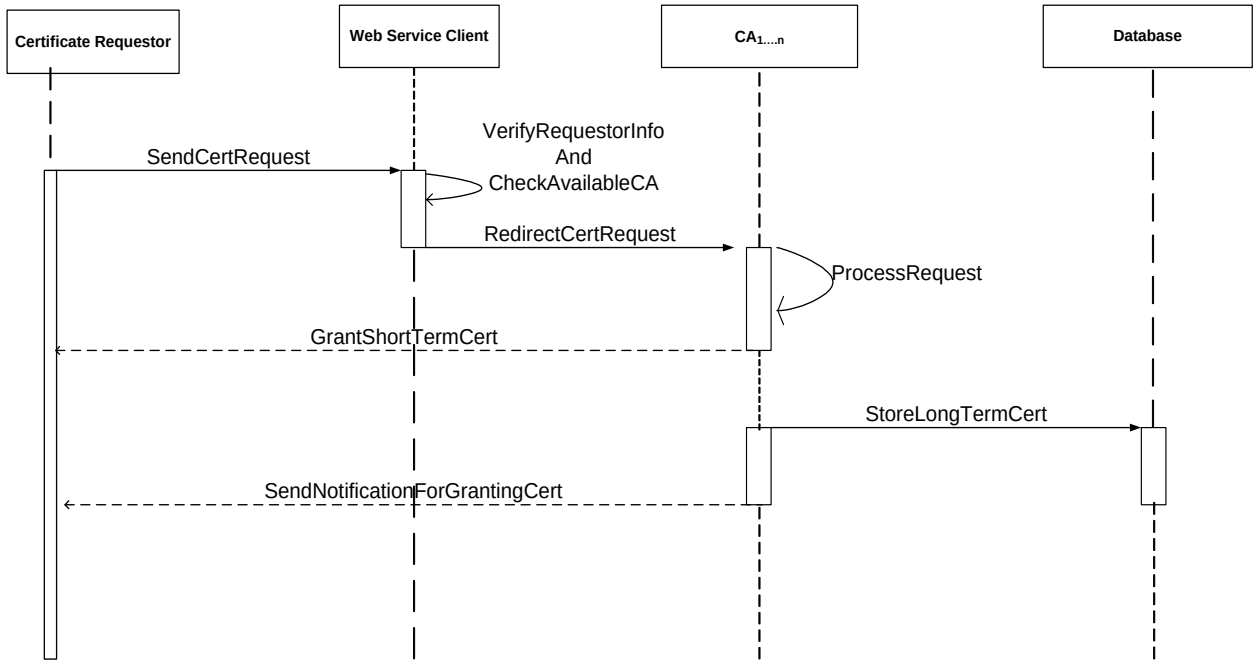


*Figure 5.2 UML sequence diagram for the Certificate-based authentication model*

### 5.3.3 Activity Diagram

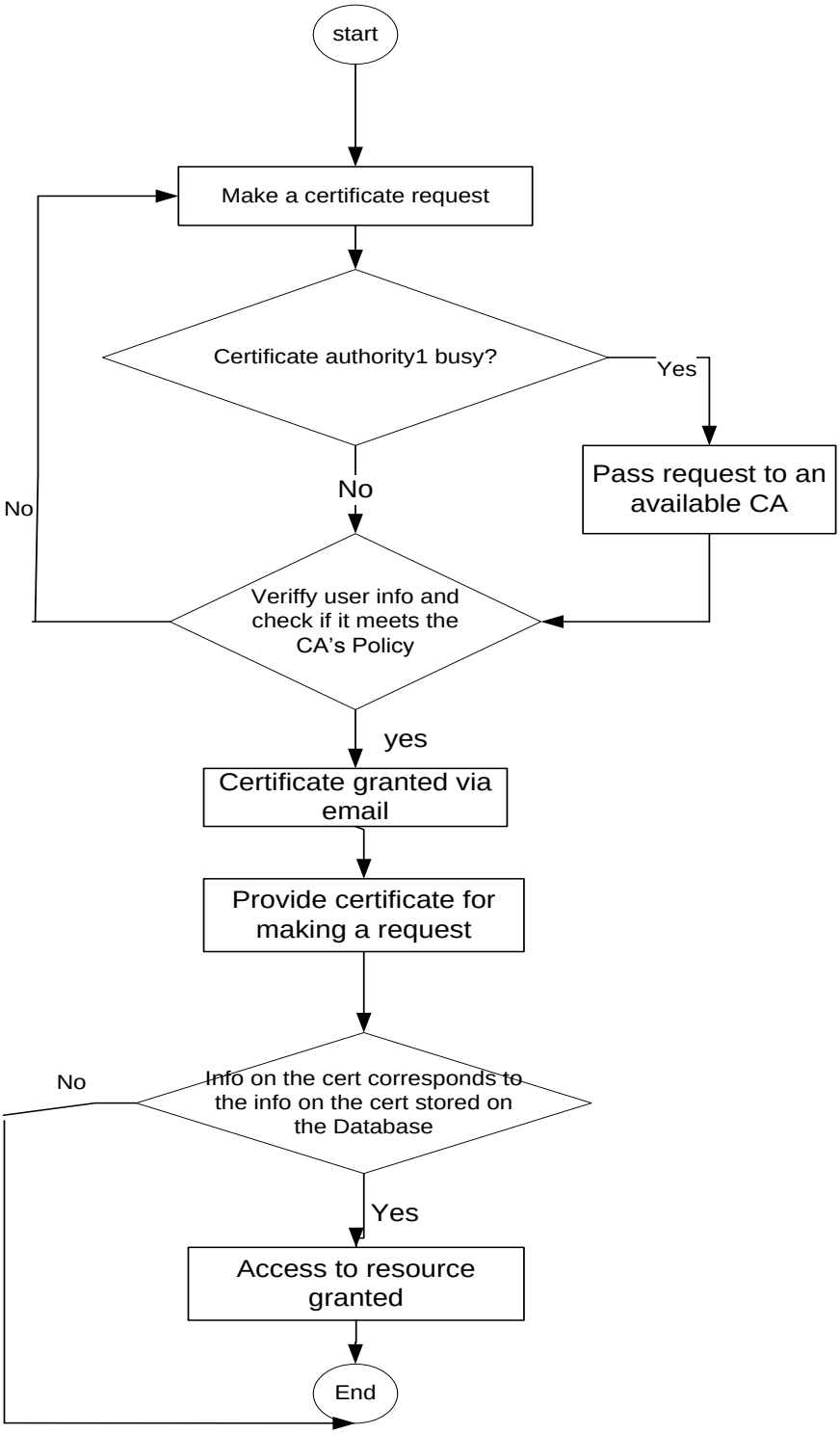


*Figure 5.3 Activity Diagram*

Figure 5.2 (Section 5.2.2) and Figure 5.3 respectively, shows the sequence diagram and activity diagram of our certificate-based authentication framework. The flow is initiated by the certificate requester or the user while sending his personal information to the web service client requesting a certificate. When the requester initiates the flow, the web service client becomes responsible for that request. The web service client verifies the requester's information sent by the certificate requester. As mentioned in previous sections, more than one CA were deployed for processing the user's requests. After verification of the user's personal information, the Web Service Client then acts as a scheduler and checks between these CAs, which one is mostly available.

When the scheduler finds an available CA, it passes the user's information and request to that CA. The CA processes the request and grants the short-term certificate directly to the requestor via an email, while sending a notification to the web service client that a certification has been granted to the requestor. The long term certification corresponding to the sent certificate is thus stored in the database by the CA for future purposes.

## 5.4 Implementation Details

Chapter 4 presented the development of our certificate based authentication model. Section 5.3 presented the design. This section, therefore, presents the details of how our certificate-based authentication infrastructure was implemented. The implementation was carried out in Netbeans 6.5 Development Environment, using Java as a programing language and Glassfish as the application server. For the implementation, we had the certificated requester which acted as a client, the Web Service Client which is the WebTester, the Certification Authorities (10 CAs implemented, each denoted by the name of the city), and we also had a database. The certificate request component was simulated and the other specifications were implemented as prototype.

The following subsections present the implementation details of our infrastructure structured into Web Service of a Certification Authority in Section 5.4.1; Web Service Client Scheduling in 5.4.2; Certificate Granting through email in Section 5.4.3; and Database in Section 5.4.4.

### 5.4.1 Web Service of a Certification Authority

When the client requires a certificate from the Certification Authority, it sends its request to the WebTester. The WebTester in turn requires personal information from the client which may include name, surname, email etc. the client does not directly require the certificate from the CA but from this WebTester which it accesses via the URL, http://10.56.4.73:9692/WebTester/. For the Web Service used to invoke the WebTester, see Figure A1 in Appendix A.

### 5.4.2 Web Service Client Scheduling

After the WebTester has gathered the information from the client, it does the scheduling process. As mentioned earlier, there are 10 implemented CAs, each denoted by cities (e.g. Durban, Cape Town etc.). So the WebTester checks the current available Certification Authorities among these 10 CAs and allocates the request to the most appropriate one. For the algorithm used for the scheduler in See figure AII in Appendix A.

### 5.4.3 Granting a certificate

The currently available CA processes the certificate request following their policies for granting the certificate and sends back the certificate to the requesting client via an email. The acknowledgement is also sent to the Web Service Client to notify it that the client has been granted the certificate. For an algorithm used for sending an email with a certificate from the CA to the requesting client see figure AIII in Appendix A.

### 5.4.4 Database

The database acts as a repository used by the Certification Authorities to store all the users' information. This may also include all the information which is sent by the client when requesting a certificate. When the CA grants a short term certificate to the client, it also stores the long term certificate in the database for future reference. For the Entity-Relationship Diagram depicting how the database is modeled see Figure AV in Appendix A.

## 5.5 Experimental Environment

The application was tested on a desktop machine running Windows & Professional Edition. The machine was an *i*5 processor with a processing speed of 3.2 GHz and 3GB of RAM. To illustrate the execution of results for our framework, an interface is designed which allows for the certificate requests by the users and is depicted below.

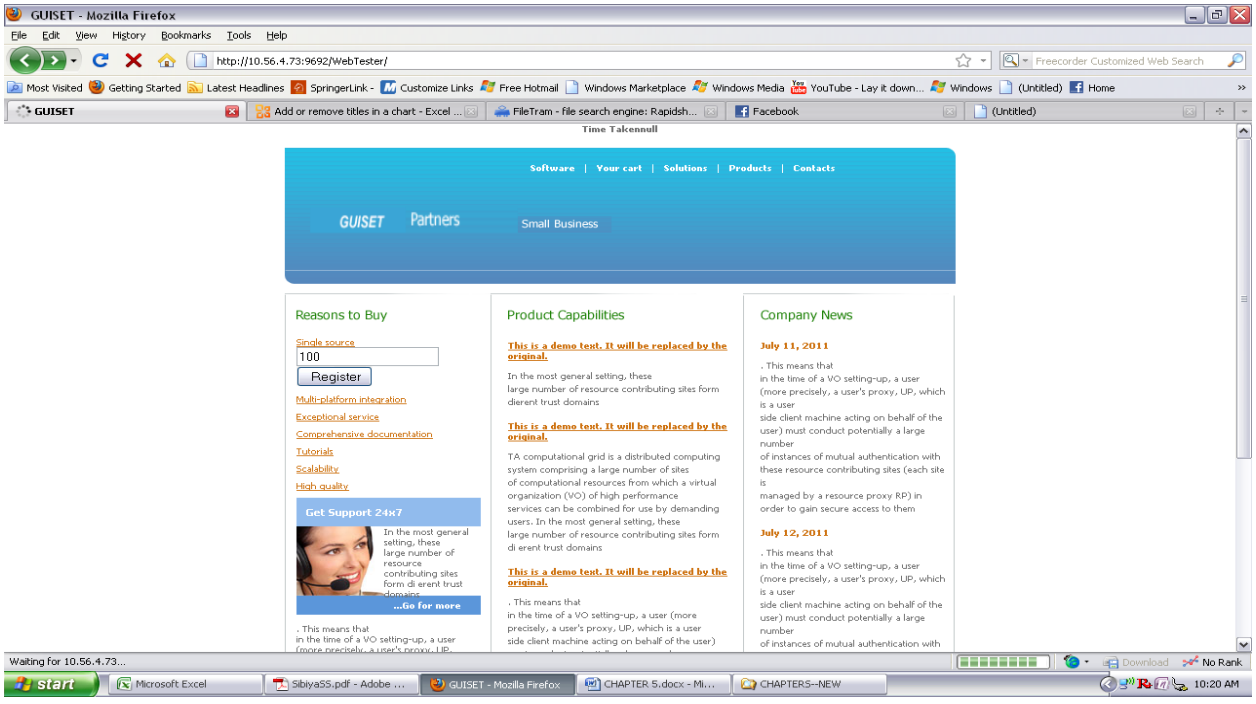


*Figure 5.4 the interface*

In the interface shown in Figure 5.4, a click to the register button in the upper area sends the users' requests for certificates to the CAs, depending on the number of CAs available on that particular moment. It has been mentioned in Chapter Four that we are using multiple Certificate Authorities and to be specific, there is a pool of 10 Certificate Authorities. After the certificates requests have been sent the results displayed back indicate the response time, that is, the time taken by the CA(s) to respond to the certificate (granting the certificate) back to the users. Figure 5.5 illustrates the response times in processing the requests by the CA(s).

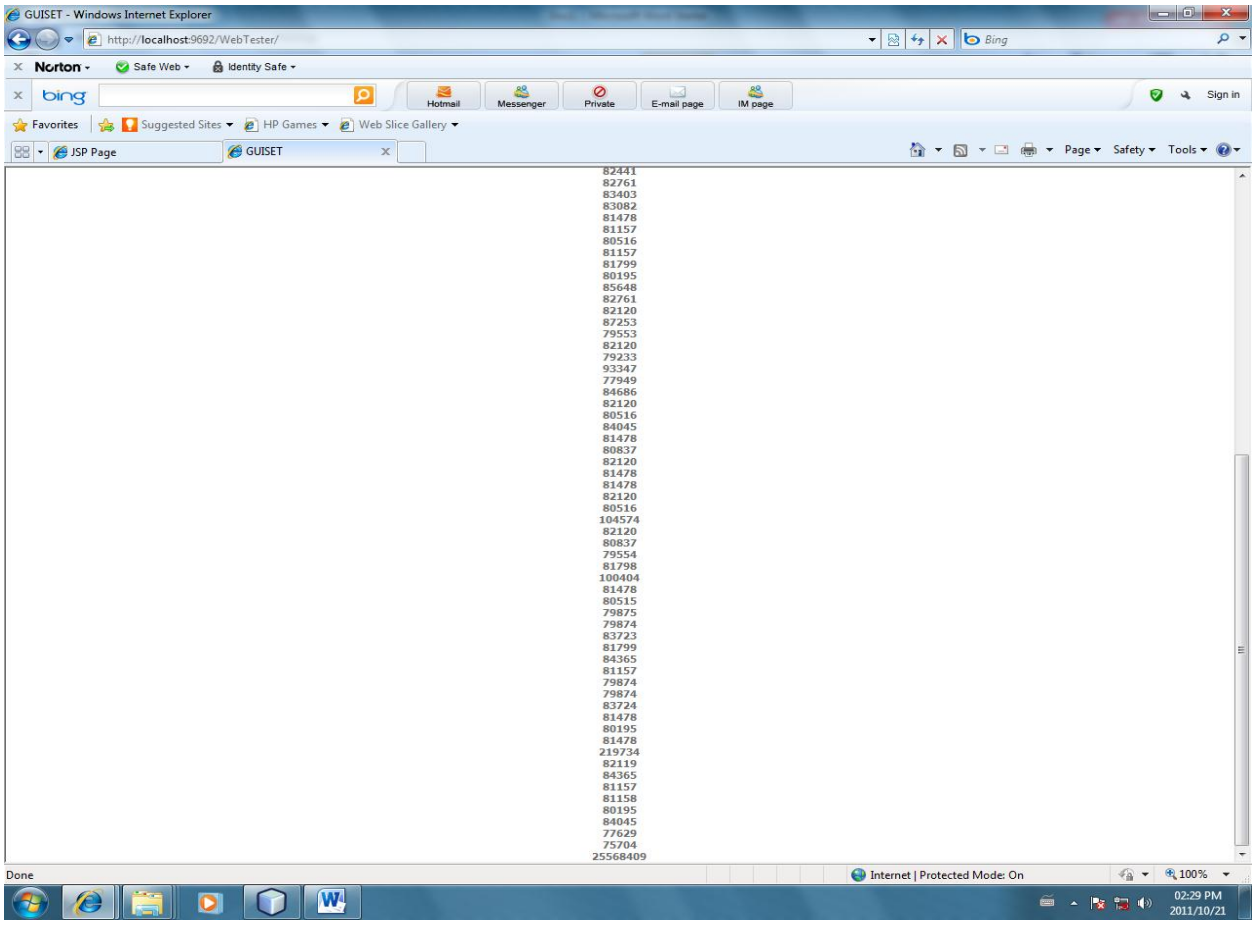


*Figure 5.5 Response time*

In figure 5.5 we recognize the response times for all the requests made, depending on the number of users that have sent the requests. The total response time is displayed as the last number on the figure displayed.

## 5.6 Experiments

This section describes in detail all the experiments carried out and analysis of the data observed, and then discusses the results that were obtained from the analysis. Four (4) experiments were conducted to determine the scalability of our certificate –based authentication framework. For a complete view of the data observed see Table A IV in Appendix A. In the Subsections below, we present the four experiments conducted.

### 5.6.1 Experiment 1

This experiment was conducted to determine the response times as the number of certificate requests increases when there are two Certification Authorities servicing the request.

**a. Experimental design**

In conducting this experiment, we used two Certification Authorities (CAs) for processing certificate requests. We started by sending 100 certificate requests up to 1000 to a single CA. 10 runs were done for each set of requests (100 to 1000) to get an average for each set. Since in this experiment we used two CAs, the same process was done for the sent requests when there are two CAs, and the average values were taken for all sets of requests. We show the data gathered for this experiment overleaf:

| Certificate Requests | Response Time (nano seconds) | |
|---|---|---|
| | 1CA | 2CAs |
| 100 | 4930309 | 2390606 |
| 200 | 8565052 | 4991159 |
| 300 | 13596019 | 8378840 |
| 400 | 18465699 | 11354800 |
| 500 | 24236515 | 13692574 |
| 600 | 28320539 | 16608459 |
| 700 | 33973567 | 18775510 |
| 800 | 37534563 | 21463795 |
| 900 | 43814581 | 24401651 |
| 1000 | 48581835 | 27101586 |

*Table 5.1 Data gathered*

## b. Experimental Results

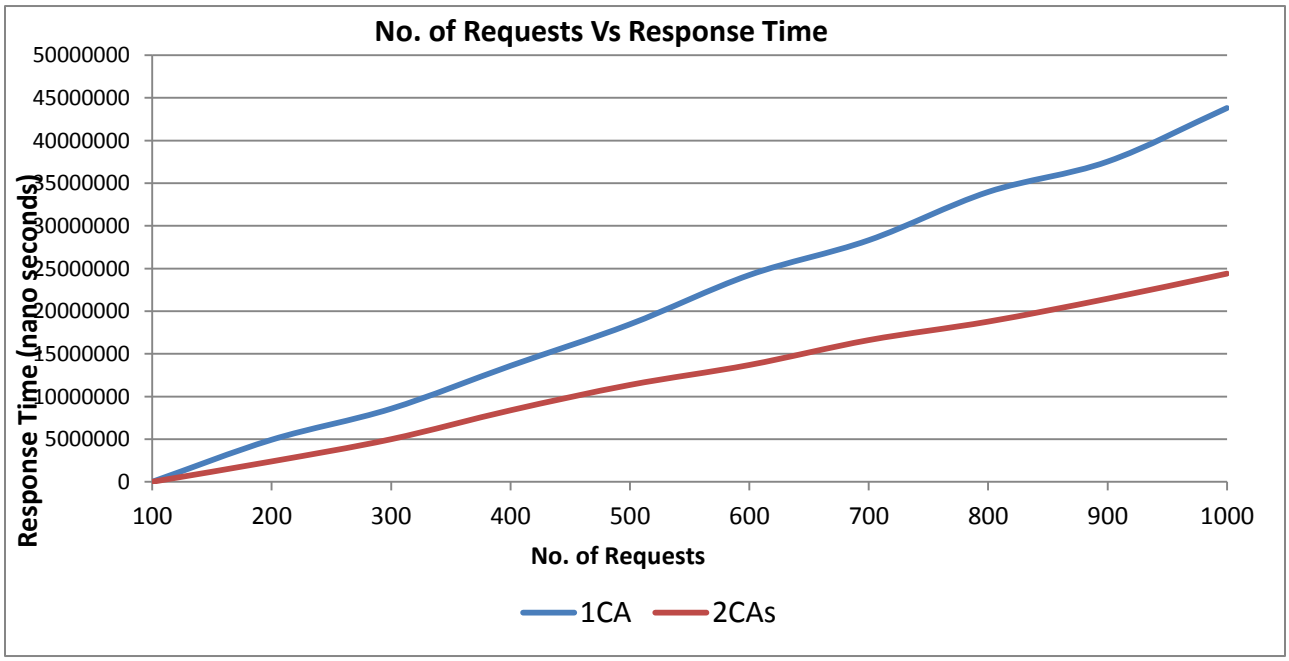


*Figure 5.6 Graph of No. of Requests against Response Times for 2 Certification Authorities*

What can be recognized in Figure 5.6 is that, with a single Certification Authority, much longer time is taken to respond to the certificates requests as the number of requesters keep increasing.

We then implemented another Certification Authority to observe, if there would be a difference. From the graph we can identify the gap between the two lines. What is then recognized from the graph is that, as the number of requests increases, the time the 2 CAs takes to process the request is becoming increasingly lesser than that of a single CA. We can then conclude that using a single CA may not be the best idea because the performance of the system drops (judging from the gap in response times as mentioned). For example: with 700 requests, the response time with one CA is approximately 25000000 nano seconds and with 2 CAs, it's approximately 15000000 nano seconds.

### 5.6.2 Experiment 2:

This experiment was conducted to evaluate the difference in response times when the number of CAs is up to 10 and the number of requests increases steadily from 10 to 1000.

a. **Experimental Design**

   Recognizing that using 2 CAs may essentially enhance the performance of the system by responding faster to certificate requests, 8 more CAs were used to test the performance of the system. We again sent sets of requests from 100 to 1000, continuing from the recordings taken in the first experiment when there were 2 CAs. We continued up to 10 CAs, doing 10 runs, thus taking the average for each set of requests. For the data obtained from this set of experiments see figure AV in Appendix A.

**b. Experimental Results**

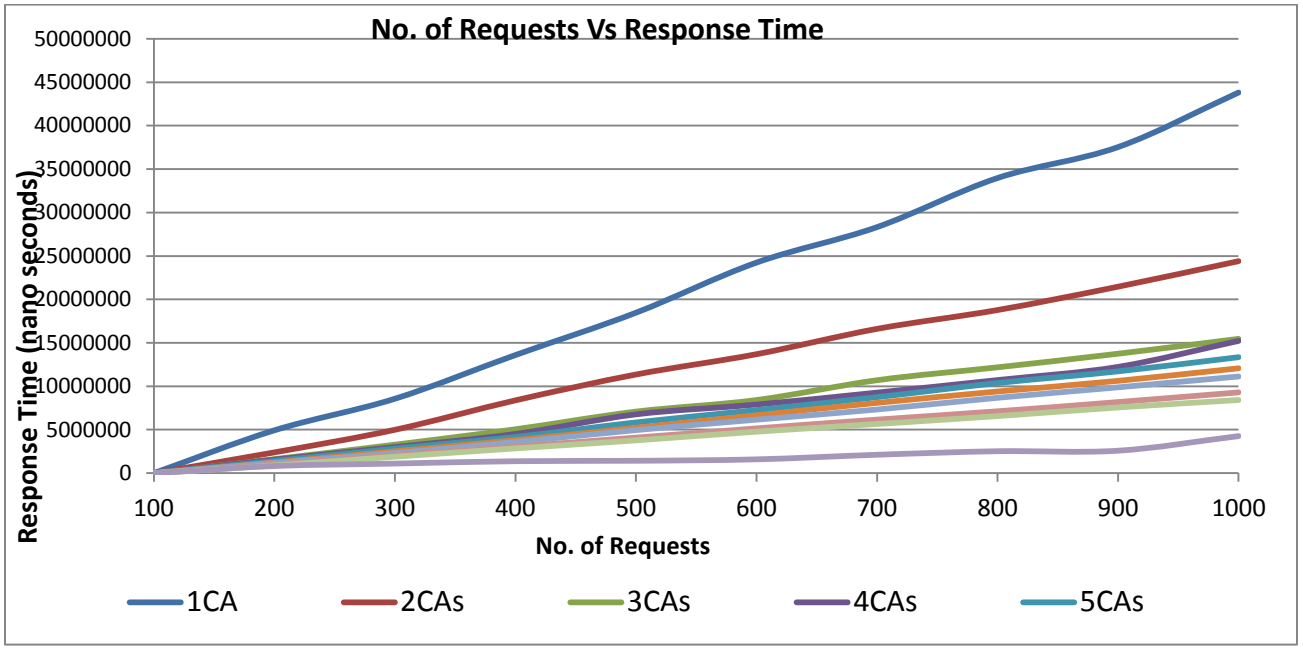


*Figure 5.7 Graph of Response Time against No. of Requests with 10 CAs*

From Figure 5.7 we observe the Response time becomes increasingly less as the number of requests increases and this is caused by the increase in the number of Certification Authorities. We conclude that the increase in the number of CAs has an effect in the response time as the number of requests increases. For instance; with 300 certificate requests, the response time is 13596019 nano-seconds with a single Certification Authority. But as the number of CAs increased to 5, the response time decreased to 4169273 nano-seconds and so on. What can also be established from the graph is that as the number of CAs is increased the response time almost remains constant and the gradient approximately gets to zero.

### 5.6.3 Experiment 3:

This experiment was conducted to establish how the gradient would behave if we had kept the number of requests constant at 1000.

### a. Experimental Design

For this experiment we sent only one set of requests which is the 1000 set. Ten runs were done for only this set and an average response time was used. We kept increasing the number of CAs from a single CA to 10 CAs whilst keeping the number of certificate requests constant. Table 5.2 shows the data gathered for this experiment:

| | Response Time (nano seconds) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Certificate Requests | 1CA | 2CAs | 3CAs | 4CAs | 5CAs | 6CAs | 7CAs | 8CAs | 9CAs | 10CAs |
| 1000 | 48581835 | 27101586 | 16867292 | 15761684 | 14596770 | 13253760 | 11864976 | 10259410 | 9466828 | 4733414 |

*Table 5.2 Data Gathered*

### b. Experimental Results

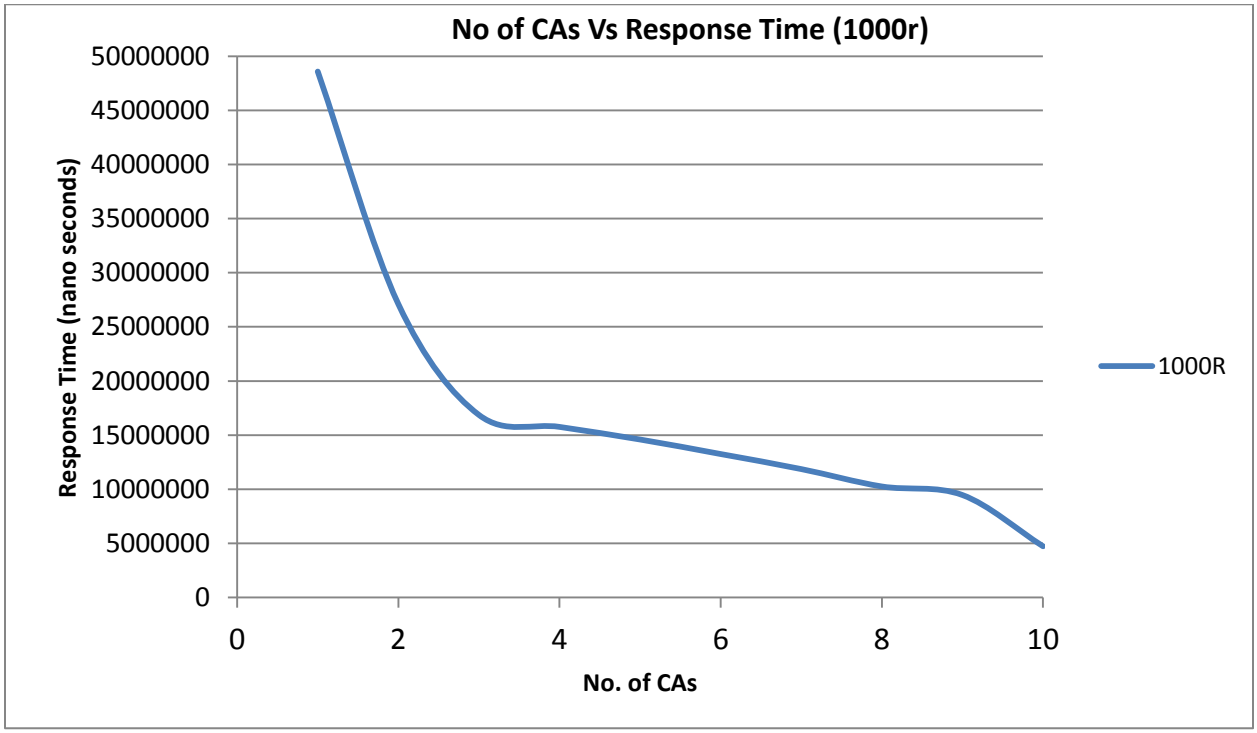


*Figure 5.8 Graph of No. of CAs against Response Time with a constant number of requests*

We can recognize the exponential decrease in response time as we increase the number of certification authorities while keeping the number of request constant. This may still mean that increasing the number of CAs has an effect in the response time when users are requesting certificates.

### 5.6.4 Experiment 4

The graph in figure 5.8 raised some questions about the trend it followed. So we conducted this experiment to essentially see the pattern it would follow if we used all the requests from 100 to 1000.

**a. Experimental Design**

In conducting this experiment we used ten CAs for processing the certificate requests, only this time, we did not keep the number of requests constant to 1000. We kept decreasing the number of requests from 1000 down to 100, doing ten runs for each set of requests and thus taking the average for each set from a single CA to ten CAs. For the data gathered see Figure AV in Appendix A.

**b. Experimental Results**

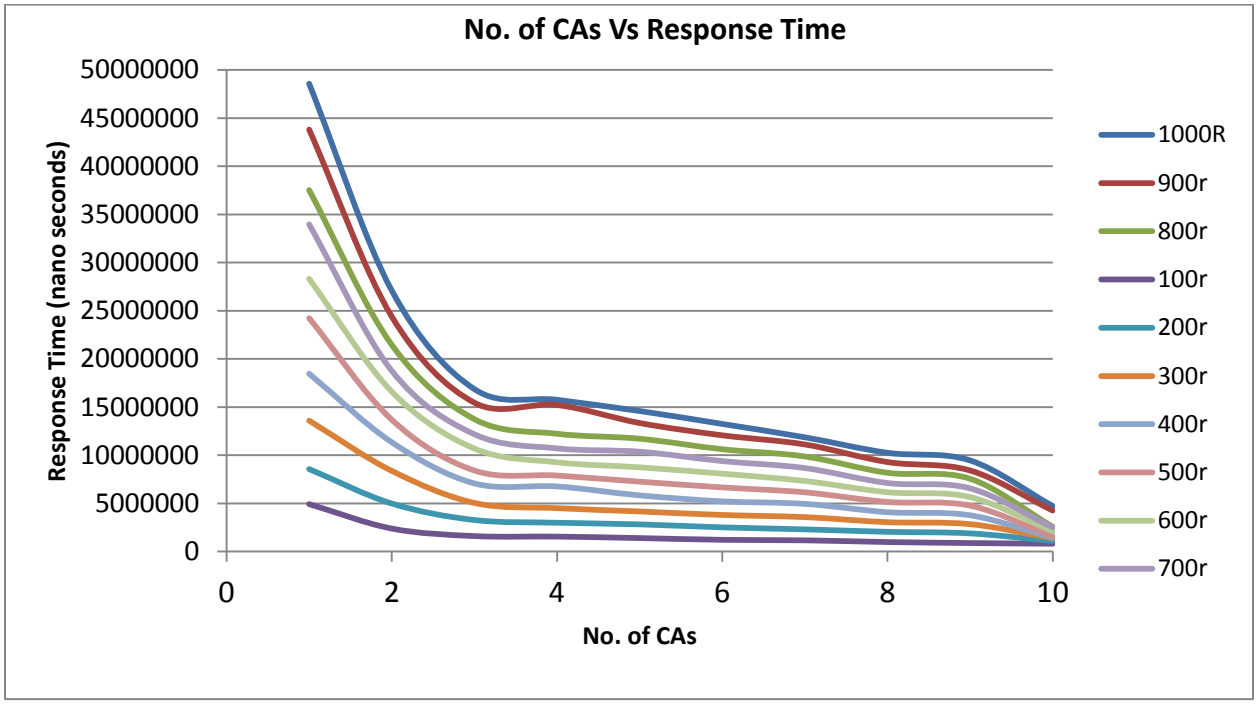


*Figure 5.9 Graph of No. of CAs Vs Response time.*

From the graph we identify an unanticipated trend in the slope of the graph. The graph shows a uniform pattern from a single CA to 3CAs with the time decreasing as the number of requests drops from a 1000 to 100. What is also noticeable is the pattern the graph took after the 3CAs, in which the response time becomes constant for almost all the number of certificate requests. This pattern goes on until it reaches the point where there are 9 CAs then changes the pattern again into much less response time.

## 5.7 Discussion of Results

From the foregoing experiments, the main basic conclusion that can be made is that our certificate-based authentication infrastructure can enable scalability even on a large scale. The literature reviewed states that, though certificate based PKI mechanisms are widely adopted, they still suffer from the uncertainty of poor scalability on a large scale because of its extensive use of

certificates. This became a concern because the GUISET infrastructure discussed in previous sections of this work would eventually; with time, involve a large number of resource contributing sites, and SMMEs as well. This thus requires an authentication mechanism that is flexible (enable scalability) in case there is that growth in the infrastructure.

Literature has proven that some grid system choose to use certificate free systems to authenticate their users. They believe certificate free mechanisms authentication protocol is more lightweight than certificate based mechanisms and this thus contributes to better scalability. Our research shows that, though some certificate based authentication infrastructures have poor scalability e.g. GSI, certificate-based authentication can enable scalability; and that though ID-based authentication prove to be complimentary, they cannot be used to replace certificate-based PKI.

Our results show that using more than a single Certification Authority will definitely enable the scalability of a certificate based authentication framework. This is delivered from the fact that, as we kept increasing the number of CAs (*refer to figure 5.6 and 5.7*), the time it took for the CAs to respond to the certificate requests is much lesser than when there was only one CA. Figure 5.9 may have revealed that, if the number of CAs becomes too much, there might not be that much impact (the gradient almost got to zero).

A firm point that we can base our conclusion on is on the human element factor. That is, in most cases certificates are granted manually. With this point we are saying; with a single Certification Authority, the users' requests may take time to be responded on because of the number of requests that may be on the queue waiting to be processed. But as the number of the Certification Authorities increases, more requests may be responded on within shorter periods, and users may be granted their certificates without waiting longer periods. From this, we can with assurance,

argue that the certificate-based authentication framework we have developed for the security of GUSIET resources provides more scalability than the previously developed grid security infrastructure.

## 5.8 Summary

In this chapter we have presented the implementation results of a certificate-based authentication framework for securing GUISET resources. Experiments on scalability were conducted. The main goal was to achieve an authentication framework that will enable scalability in contrast to other existing certificate based authentication framework. This was promoted by the proven concern that most certificate based systems for authenticating grid users provide poor scalability due to the use of certificates when proving users' identities. This goal was achieved by developing 10 Certification Authorities, thus enabling the certificate requesters to acquire their certificates as faster as possible, without need to wait for a single CA which mostly results in performance degradation of a system. With the users of GUISET expected to grow with time, the security system developed should be able to maintain its integrity; that is not keeping users for longer periods in getting their prove of identities when in need of accessing the GUISET resources. As mentioned in Section 5.7, certificates are at most granted manually, so with many CAs, the chances are that the users will be granted access much faster.

# CHAPTER 6

# CONCLUSION AND FUTURE WORK

## 6.1 Introduction

In this dissertation, we have tried to address the challenges encountered by the GUISET infrastructure in relation to the usage of its resources. These challenges brought the need for a security infrastructure to secure these resources. An authentication infrastructure becomes of high necessity in authenticating the individuals requiring access to the GUISET resources. This work specifically focused on the scalability of that authentication infrastructure developed. This is based on the concern that GUISET as an evolving infrastructure that needs to make its mark of effectiveness, the number of individuals who may require access to its resources may increase from time to time. It is thus of common knowledge that if there is an increase or a decrease in the number of grid system users, its security system should be able to adapt to that change without degrading the performance of the system. The overall goal of this research work was to conduct a study of a certificate-based framework that would be suitable for securing GUISET resources, without failing in scalability.

This chapter concludes the dissertation and summarizes it in Section 6.2. In Section 6.3 we conclude the chapter by suggesting limitations and the future works of our work.

## 6.2 Conclusion

From the existing literature, we identified that in grid environments (GUISET in this case) SMMEs pool their resources and expertise together for collaboration and among themselves and their external parties. Provisioning of appropriate security seems to be more challenging in such kinds of environments than in most conventional distributed systems. We discovered that the Public Key Infrastructure GSI is the presently deployed technology in many grid implementations as it is recognized as a continuous and sophisticated technology. The motivation behind this technology includes its provision of delegation, single sign-on, and that it also limits the exposure of long-term credentials.

Though the PKI-GSI technology is highly recommended, it still has its downsides like any other system. Its use of certificates limits scalability. We argue that the scalability problem resulting from this technology may limit the usage of GUISET resources thus degrading the performance of the infrastructure. We, therefore, have established that a more lightweight authentication mechanism than a traditional PKI-GSI is required for GUISET. We propose a certificate-based authentication framework for securing GUISET resources which would enable scalability as the usage of GUISET resources increases.

As we have mentioned above, the use of certificates appears to be the issue that causes scalability. We believe that this comes from the fact that when the users of a certain grid environment need access to resources, they are required to be authenticated. This becomes much more of a problem when there are many users because the Certification Authority granting certificates to appropriate users becomes overloaded with certificate requests, which thus causes the scalability problem. We implement 10 CAs so that users can get the certificates as fast as

possible. We use this implementation to conduct performance evaluation experiments. The method we used to check our metric, which is scalability, was by checking the effect on the response time if we kept increasing the number of users acquiring certificates, and also the response time as we also implemented more and more CAs.

Analysis of the results obtained from the implementation revealed that our certificate based authentication framework provides better scalability due to the use of many Certification Authorities. In essence, the evaluation concluded that our certificate-based authentication framework as proposed in this research work provides the required security for GUISET resources and enables scalability as there may be an increase in the usage of the system. This thus means there would not be any performance deprivation for infrastructure because of the security system implemented.

## 6.3 Limitations

Although our certificate based authentication infrastructure has been proven to enable scalability, it has some limitations which could be recommended for future enhancements. For example, when testing our system the time spent by the data on the wire was considered insignificant. In practice, network factors have a direct factor when testing for scalability, such as response time. As these factors were not considered, it would be interesting to see how these network-metrics influence the behavior of our certificate-based authentication framework. The other issue could be that we used the theory that in most cases, CAs are a human factor, which basically assisted us onto delivering our proposed model. In the future, it would be motivating to consider automated CAs and also see the effect it would have on the authentication system.

## 6.4 Future Work

In future, we plan on deploying our model on a definite grid infrastructure (if possible GUISET) and together with existing SMMEs as case studies and expect to report results in consequent peer-reviewed publications of this work.

# BIBLIOGRAPHY

Adigun, M., Emuoyibofarhe, O., Migira, S. (2006). “Challenges to Access and Opportunity to use SMME enabling Technologies in Africa” 1st All Africa Technology Diffusion Conference. Johannesburg South Africa, June 12-14, 2006.

Al-Riyami, S., Paterson, K. (2003). Certificate less public key cryptography. In C.S. Laih, editor, Advances in Cryptology - Proceedings of ASIACRYPT 2003, pages 452-473. Springer-Verlag LNCS 2894, 2003.

Anderson, J. (1991). "Guide to Understanding Identification and Authentication in Trusted Systems (Light Blue Book)" .National Computer Security Center NCSC-TG-017, September 1991.

Avizienis, A., Laprie, J., Randel, B., Landwehr, C. (2004). “Basic Concepts and Taxanomy of Dependable and Secure Computing”. In IEEE Transactions on Dependable and Secure Computing, Vol 1, Jan-March 2004.

Baker, R., Lorch, M., Ramakhrishnan, L. (2004). “Conceptual Grid Authorization Framework and Classification”. Global Grid Forum, 23 November 2004, www.gridforum.org/documents/GDF.38.pdf.

Basney, J., Humphrey, M., Welch, V. (2005). “The MyProxy Online Credential Repository”. Software: Practice and Experience, Volume 35, Issue 9, July 2005

Beckles, B., Welch, V., Basney, J. (2005). "*Mechanisms for increasing the usability for grid security*". International Journal of Human-Computer Studies, 63(1-2):74-101, 2005.

Becko, S. (2006). "The Role of Workflow in Next Generation Business Oriented Grids: Two Different Approaches Leading to a Unified Vision". Second IEEE International Conf. on e-Science and Grid Computing, 2006.

Bhatia, K., Chandra, S., Mueller, K. "GAMA: Grid Account Management Architecture". Proceedings of the First International Conference on e-Science and Grid Computing, Dec 2005.

Boneh, D., Franklin, M. (2001). "Identity-based encryption from the Weil pairing". Advances in Cryptology- Proceedings of CRYPTO 2001, pp. 213–229. Springer-Verlag LNCS 2139, August 2001.

Burgess, S. (2002). "Managing Information Technology in Small Business Challenges and Solutions". Victoria University, Australia, 2004.

Burrows, M., Lampson, B., Abadi, M., Wobber, E., "Authentication in Distributed Systems: Theory and Practice," *ACM Transactions ComputerSystems*, vol. 10, pp. 265-310, 1992.

Chakrabarti, A. (2007). "Grid Computing Security". Springer-Verlag, 2007.

Chadwick, D., Otenko, O. (2002). "The PERMIS X.509 Role-Based Privilege Management Infrastructure". In Proceedings of the 7th ACM Symposium on Access Control Models and Technologies (SACMAT 2002), June 2002.

Chen. L., Lim. H., Mao, W. (2005). “User-friendly grid security architecture and protocols”. In Proceedings of the 13th International Workshop on Security Protocols, 2005.

Clercq, J. (2002). “Infrastructure Security”. International Conference, InfraSec 2002 Bristol, UK, October 2002 Proceedings.

Conklin, A., Dietrich, G., Walz, D. (2004). “Password-Based Authentication: A System Perspective”. Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04), Vol. 7, 2004.

Crampton, J., Lim, H., Paterson, K., Price, G. (2007) “A Certificate-Free Grid Security Infrastructure Supporting Password-Based User Authentication”. In Proceedings of the 6th Annual PKI R&D Workshop, 2007.

Curry, I. (1996). “Version 3 X.509 Certificates”. Entrust Technologies White Paper.

Farrel, S. (2004). “Securerly available credentials protocols”. RFC 3767. IETF, June 2004.

Fellestein, C. and Joseph, J. (2004). “Grid Computing”. Prentice Hall Professional, pg 10, 2004.

Foster, I. (1998). “A Security architecture for computational grids”. In ACM conference on computers and security, 1998.

Foster, I., Kesselman, C., Tuecke, S. (2001). “The Anatomy of the Grid: Enabling Scalable Virtual Organizations”. International Journal of High Performance Computing Applications, 15(3), pg 200–222, 2001. Also available from http://www.globus.org/ research/papers/anatomy.pdf

Foster, I. (2001). " The globus toolkit for grid computing". In Proceedings of the 1st International Symposium on Cluster Computing and the Grid, IEEE Computer Society, 2001.

Foster, I. (2002). "What is the Grid? A three point checklist." GRIDToday, July 2002.

Foster, I. Kesselman, C. (2003). "The Grid in a Nutshel"l. In: Grid Resource Management – State of the Art and Future Trends. Nabrzyski, J., Schopf J. and Weglarz, J. Kluwer Academic Publishers, 2003.

Foster, I., Kesselman, C. (1999). "The Grid: Blueprint for a New Computing Infrastructure". Morgan Kaufmann Publishers. ISBN 1-55860-475-8. http://www.mkp.com/grids/.

Foster, I., Kesselman, C., Nick, J. (2002). "Grid Services for Distributed System Integration", IEEEComputer 35(6), pp. 37-46, June 2002.

Gadelha, L., Schulze, B. (2007). "On the Management of Grid Credentials". Proceedings of the 5th international workshop on Middleware for grid computing, MGC'07.

Gasser, M., McDermott, E. (1990). "An Architecture for Practical Delegation in a Distributed System". In Proc. 1990 IEEE Symposium on Research in Security and Privacy, IEEE Press, pp. 20-30, 1990.

Guo. Y., Ma, J., Wang. Y. (2005). "An Intrusion-ResilientAuthorization and Authentication Framework for Grid Computing Infrastructure". In proceedings of the Workshop on Grid Computing Security and Resource Management, Springer Berlin / Heidelberg,Vol.3516, pp.229-236, 2005.

Haidar, A., Abdallah, A. (2009). “Formal Modelling of PKI Based Authentication”. Proceedings of the 4th International Workshop on Automated Specification and Verification of Web Systems, Vol. 235, pp 55-70, 2009.

Halevi, S., Krawczyk, H. (1999). “Public-Key Cryptography and Password Protocols,” ACM Trans. *Information System Security*, (ACM Press), vol. 2, no. 3, 1999, pp. 230-268.

Hongweia, L., Shixina, S., Haomiaoa, Y. (2008). "Identity-Based Authentication Protocol for Grid", Journal of Systems Engineering and Electronics, Vol. 19, no. 4, pp.860-865, August 2008.

Housley, R., Polk, W., Ford, W., Solo, D. (2002). “Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile”. The Internet Engineering Task Force (IETF*)*, RFC 3280, April 2002.

Huang, L., Wu, Z. (2004). “A PKI-based Scalable Security Infrastructure for Scalable Grid”. Grid Cooperative Computing, pt 2, pp. 1051-1054, 2004.

Huang, X., Chen, L., Huang, L., Li, M. (2005). “An Identity-Based Grid Security Infrastructure Model”. ISPEC’05 Proceedings of the 1st International Conference on Information Security Practice and Experience, pg 314-325, 2005.

Hurley, J. (2003). “Overview of Grid Computing”. Available at: www.educause.edu/ir/library/pdf/DEC0306.pdf, accessed: 14 April 2006.

Igbaria**,** M., Zinatelli, N., Cragg, P. and Cavaye, A. L. M. (1997). “Personal computing acceptance factors in small firms: A structural equation model”. *MIS Quarterl*y,September, 21(3), 279-305. Minneapolis.

.

Jiancheng, N., Zhishu, L., Zhonghe, G., Jirong, S. (2007). “Threats Analysis and Prevention for Grid and Web Service Security”. Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007.

Johnson, I. (2007). “Security Requirements for a Grid-based OS”. Building and Promoting a Linux-Based Operating System to support virtual organization for next generation grids. Project no. FP6- 033576. XtreemOS.

Kabanda, S., Johnson, I., Adigun, M. (2007). “Knowledge Resource Providers in a Grid-enabled Infrastructure- The Case of Deep Rural Small and Medium Enterprises”. Proceedings of the 4th International Conference on Intellectual Capital, Knowledge Management and Organizational Learning, 2007.

Kent, R., Zhao, S., Aggarwal, A. (2007). “PKI-Based Authentication Mechanisms in Grid Systems”. In proceedings of the International Conference on Networking, Architecture and Storage, pp.83-90, Guilin, July 2007.

Kornievskaia, O., Honeyman, P., Doster, B., Co®man, K. (2001). “Kerberized credential translation: A solution to web access control”. In Proceedings of 10th USENIX Security Symposium, pages 235-250, August 2001.

Kuhn, D., Hu, V., Polk, W., Chang, S. (2001). “Introduction to Public Key Technology and the Federal PKI Infrastructure”. National Institute of Standards and Technology. 2001.

Kwon, S. (2004). “Cryptanalysis for Secure Key Issuing in ID-based Cryptography and Improvement”, Manuscript. 2004.

Laccetti, G. and Schmid, G. (2007). "A framework model for grid security”, Future Generation Computer Systems, vol. 23, no. 5, pp.702-713, June 2007.

Laganier, J. (2005). “HIPernet: A decentralized Security Infrastructure for Large-Scale grid Environments”. The 6th IEEE/ACM International Workshop on Grid Computing, Nov 2005.

Lampson, B., Abadi, M., Burrows, M., Wobber, E. (1992). “Authentication in Distributed Systems: Theory and Practice”. ACM Transactions Computer Systems, vol. 10, pp. 265-310, 1992.

Langella, S., Oster, S., Hastings, S., Sienbenlist, F., Kurc, T., Saltz, J. (2006). “Dorian: Grid Service Infrastructure for Identity Management and Federation”. Proceedings of the 19th IEEE Symposium on Computer-Based Medical Systems (CBMS’06), 2006.

Lee, S. (2004). “Cryptanalysis of User Authentication Scheme Using Hash Functions”. ACM SIGOPS Operating Systems Review, Vol. 38, Issue 1, pg 24-28, Jan 2004.

Lim, H. and Robshaw, M. (2004). “On identity-based cryptography and GRID computing”. In M. Bubak,G. Albada, P. Sloot, and J. Dongarra, editors, Proceedings of the 4th International Conference on Computational Science (ICCS 2004) , pages 474–477, 2004.

Lim, H. and Robshaw, M. (2004). “A dynamic key infrastructure for GRID. In P. Sloot, A. Hoekstra, T. Priol, A. Reinefeld, and M. Bubak, editors, Proceedings of the European Grid Conference (EGC 2005), pages 255–264, 2005.

Lim, H., Paterson, K. (2005). “Identity-Based Cryptography for Grid Security”. Proceedings of the First International Conference on e-Science and Grid Computing (e-Science’05), 2005.

Lim, H. (2006). “On the Application of Identity-Based Cryptography in Grid Security”. Ph.D Thesis, University of London, May 2006.

Lin, C., Shen, J., Hwang, M. (2003). “Security enhancement for Optimal Strong-Password Authentication protocol”. ACM SIGOPS Operating Systems Review, Vol.37, No.2, pg.7-12, April 2003.

Liu, S., Liang, Y., Books, M. (2007). “Eucalyptus”. A Web Service-enabled Infrastructure. Proceedings of CASCON 2007, pp. 1-11, 2007.

Liu, S., Liang, Y., Books, M., Xu, B., Spencer, B., Zhang, L. (2007). “On demand network and application provisioning through web services”. In IEEE International Conference on web Services (ICWS), pp. 1120-1127, July 2007.

Lock, R. (2002). “Grid Security Requirements, Interactions, Mechanisms and Models”. Available at: comp.lancs.ac.uk/computing/**.../Security Requirements** for **Grid**s.pdf. Last Accessed 14 April 2006.

Lock, R., Sommerville, I. (2002). “Grid Security and its use of X.509 Certificates”. DIRC internal Conference submission. Lancaster DIRC, Lancaster University, 2002.

Lorch, M., Basney, J., Kafura, D. (2004). "A Hardware-secured Credential Repository for Grid PKIs". IEEE International Symposium on Cluster Computing and the Grid, 2004.

Macdonald, R., Smith, S., Marchesini, J., Wild, O. (2003). "Bear: An Open-Source Virtual Secure Coprocessor based on TCPA". Computer Science Technical Report TR2003-471, Dartmouth College, August 2003.

Mao, W. (2004). "An Identity-Based Non-Interactive Authentication Framework for Computational Grids". HP Lab, Technical Report HPL-2004-96, June 2004, Available: http://www.hpl.hp.com/techreports/2004/HPL-2004-96.pdf.

Marchesini, J. (2005). "SHEMP: Secure Hardware Enhanced MyProxy". PhD thesis, Dartmouth College, 2005.

Matyas, J., Riha, Z. (2003). "Toward Reliable User Authentication through Biometrics". IEEE Security & Privacy, vol. I, pp. 45-49, 2003.

Moss, A., Liu, S., Richard, R. (2008). "A Unified Authentication Framework for Accessing Heterogeneous Web Services". 4th International Conference on Next Generation Web Services Practices, 2008.

McDonagh, P., Prothero, A. (2000). "*Euroclicking and the Irish SME: Prepared for ecommerce and the single currency*?" Irish Marketing Review, 13(1), 21-33. Dublin.

Needham, R., Schroeder, M. (1978). "Using encryption for authentication in large networks of computers". Communications of the ACM, 1978.

Neuman, C. and Ts'o, T. (1994). "Kerberos: An Authentication Service for Computer Network", IEEE Communications, vol. 32, no. 9, Sep 1994,pp. 33-38.

Novotny, J., Tuecke, S., Welch, V. (2001). "An Online Credential Repository for the Grid: MyProxy". Proceedings of the Tenth International Symposium on High Performance Distributed Computing (HPDC-10), IEEE Press, pp. 104-111, Aug 2001.

OASIS Security Services (SAML) TC, http://oasisopen.org/committees/tchome.php?wgabbrev=security

Raghunathan, S., Mikler, R., Cozzolino, C. (2005). **"**Secure agent computation: X.509 Proxy Certificates in a multi-lingual agent framework**".** Journal of Systems and Software, Vol. 75, Issues 1-2, Pages 125-137, Feb 2005.

Ramakrishnan, L. (2004). "Securing Next Generation Grids". IT Professional, Vol 6, No 2, pg 34-39, March-April 2004.

Rigney, C., Willens, S., Rubbens, A., Simpson, W. (2000). "Remote Authentication Dial in User Service (RADIUS)". IETF RFC2865, June 2000.

Sakai, R., Ohgishi, K., Kasahara, M. (2000). Cryptosystems based on pairing. In Proceedings of the 2000 Symposium on Cryptography and Information Security (SCIS 2000), January 2000.

Sarbari, G. (2002). "Security characteristics of cryptographic mobility solutions". Proceedings of the 1st Annual PKI Research Workshop. Gaithersburg, MD, April 2002.

Schopf, J. (2002). "Grid Computing and the Globus Toolkit'. Available at: www.sztaki.hu/~vajda/**Grid** technika/**GridGlobus**(**Schopf**).ppt, accessed: 14 April 2006.

Schulze, B., Gadelha Jr., L. (2007). “On Management of Grid Credentials”.MGC’07 Proccedings of the 5th International Workshop on Middleware for Grid Computing, 2007.

Shibboleth Project (2008) http://www.shibboleth,internet2.edu

Sotomayor, B. (2006). “Globus Toolkit 4: Programing Java Services”. Morgan Kaufmann, London, 2006.

Srivatanakul, T., Clark, J., Polack, F. (2004). “Effective security requirements analysis: HAZOP and use cases. In Information Security: 7th International Conference, volume 3225 of LNCS.

Tardo, J., Alagappan, K. (1991). “SPX: Global Authentication Using Public Key Certificates”. Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy, pp. 232-244, 1991.

The Globus Project (2003). GT3 Overview. Available at: www.nsfgrid.marits.edu/docs/introductionGT3.pdf. Last accesses 14 April 2006.

Thompson, M., Essian, A., Mudumbai, S. (2003). “Certificate-Based Authorization Policy in a PKI Environment”. ACM Transactions on Information and System Security (TISSEC), Vol 6, Issue 4, pg 566-588, Nov 2003.

Vivas, L., Fernandez-Cago, C., Lopez, J., Benjumea, A. (2009). “A Security Framework for a Workflow-Based Grid Development Platrform”. Computer Standards and Interfaces 32, April 2009.

Volker, T. (2001). “Collaborative Commerce- Trends and Technology Potentials”. First DEEDS Policy Group Meeting, Brussels.

Wang, W., Wang, R. (2007). “CPK-based grid authentication: a step forward,” The Journal of China Universities of Posts and Telecommunications, vol.14, pp.26-31, 2007.

Welch, V., Siebenlist, F., Foster, I., Bresnahan, J., Czajkowski, K., Gawor, J., Kesselman, C., Metier, S., Pearlman, L., Tuecke, S. (2003). “Security for Grid Services”. In Twelfth International Symposium on High Performance Distributed Computing, pp. 48-57, 2003.

# APPENDIX A

## I. WEB SERVICE OF A CA

```
public class CapeTownWebService {

    @WebMethod(operationName = "CapaTownRegistation")
    public String registation(@WebParam(name = "name")
    String name, @WebParam(name = "email")
    String email, @WebParam(name = "password")
    String password) {
        //TODO write your implementation code here:

         new Regisration(name ,email,password);

        return "Your Have been registed check your email to activated your
account";
    }
}
```

## II. WEB SERVICE CLIENT SCHEDULING ALGORITHM

```
int noOfTime = (Integer.parseInt(noOfUsers)) / 2;
            long myTime = 0;
            for (int i = 0; i < noOfTime; i++) {
                long time1 = java.lang.System.nanoTime();

                //registation.NewWebService_Service service = new
registation.NewWebService_Service();
                //registation.NewWebService port = service.getNewWebServicePort();

                // TODO initialize WS operation arguments here

                Thread thread1 = new Thread() {

                    public void run() {
                        websevice2.CapeTownWebServiceService service = new
websevice2.CapeTownWebServiceService();
                        websevice2.CapeTownWebService port1 =
service.getCapeTownWebServicePort();
                        try {
                            java.util.Scanner inFile = new java.util.Scanner(new
java.io.FileReader("c:\\data1.txt"));
                            java.lang.String name = inFile.next();
                            java.lang.String email = inFile.next();
                            java.lang.String password = inFile.next();
                            // TODO process result here
                            java.lang.String result =
port1.capaTownRegistation(name, email, password);

                            System.out.println("Durban :" + email);
                        } catch (Exception e) {
                        }                    }
                };
                Thread thread2 = new Thread() {
                    public void run() {
                        websevice2.DurbanWebServiceService service = new
websevice2.DurbanWebServiceService();
                        websevice2.DurbanWebService port1 =
service.getDurbanWebServicePort();
                        try {
```

## III. CERTIFICATE GRANTED VIA EMAIL

```
public class SendingEmail
{
    public SendingEmail(String my_email,String key, String password) {
    {
            String to =  my_email;
            String from = "sachie2@gmail.com";

            String host = "smtp.gmail.com";

            Properties props = new Properties();
            props.put("mail.smtp.host", host);
            // To see what is going on behind the scene
            props.put("mail.debug", "true");
            Session session = Session.getInstance(props);

            try {

                Message msg = new MimeMessage(session);
                msg.setFrom(new InternetAddress(from));
                InternetAddress[] address = {new InternetAddress(to)};
                msg.setRecipients(Message.RecipientType.TO, address);
                msg.setSubject("Guiset registration");
                msg.setSentDate(new Date());

                msg.setText("Your password
:"+password+"\n"+"\n"+"\n"+key+"\n"+"\n"+"\n"+"This email is to  a " +
                                "Conformation that you have registered  .\n" +
                                "Thank you by Guiset.");
                  //Send the message
                  Transport.send(msg);
             }
             catch (MessagingException mex) {
                 // Prints all nested (chained) exceptions as well
                 mex.printStackTrace();            }
        }

    }
```

## IV. EXPRETIMENTAL RESULTS (Data Gathered)

| Certificate Requests | Response Time (nano seconds) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **1CA** | **2CAs** | **3CAs** | **4CAs** | **5CAs** | **6CAs** | **7CAs** | **8CAs** | **9CAs** | **10CAs** |
| 100 | 4930309 | 2390606 | 1610808 | 1559257 | 1402450 | 1223037 | 1169121 | 994001 | 888563 | 817880 |
| 200 | 8565052 | 4991159 | 3279225 | 2998927 | 2825366 | 2517192 | 2309181 | 2051708 | 1890428 | 1100280 |
| 300 | 13596019 | 8378840 | 5051964 | 4510004 | 4169273 | 3804065 | 3579326 | 3053248 | 2847435 | 1376160 |
| 400 | 18465699 | 11354800 | 7082706 | 6766897 | 5838547 | 5219410 | 4953839 | 4091099 | 3771409 | 1424270 |
| 500 | 24236515 | 13692574 | 8415042 | 7896242 | 7261527 | 6664813 | 6153937 | 5153395 | 4771699 | 15881450 |
| 600 | 28320539 | 16608459 | 10696959 | 9273101 | 8747958 | 8089589 | 7327415 | 6164172 | 5659777 | 2117150 |
| 700 | 33973567 | 18775510 | 12183399 | 10715659 | 10372254 | 9399430 | 8673214 | 7120896 | 6566781 | 2524540 |
| 800 | 37534563 | 21463795 | 13747912 | 12233278 | 11719601 | 10626796 | 9877939 | 8184059 | 7562036 | 2585490 |
| 900 | 43814581 | 24401651 | 15458422 | 15223674 | 13350189 | 12072586 | 11123012 | 9296396 | 8420760 | 4263170 |
| 1000 | 48581835 | 27101586 | 16867292 | 15761684 | 14596770 | 13253760 | 11864976 | 10259410 | 9466828 | 4733414 |

## V. ENTITY RELATIONSHIP DIAGRAM FOR THE DATABASE