

ENHANCEMENT OF A SIMPLE USER AUTHENTICATION  
SCHEME FOR GRID COMPUTING

VIKNESH A/L RAMAMOORTHY

UNIVERSITI TEKNOLOGI MALAYSIA

ENHANCEMENT OF A SIMPLE USER AUTHENTICATION  
SCHEME FOR GRID COMPUTING

VIKNESH A/L RAMAMOORTHY

A project report submitted in partial fulfillment of the  
requirements for the award of the degree of  
Master of Science (Computer Science)

Faculty of Computer Science and Information Systems  
Universiti Teknologi Malaysia

OCTOBER 2008

To my beloved mother, father, brothers and  
Precious relatives and friends

Thank you for your immense love, your precious prayers, supports and for all that you  
have done to me. May the blessings of God, shower upon you.

## ACKNOWLEDGEMENT

In the name of God, first and foremost, praise is to God, The Cherisher and Sustainer of the Worlds. With the help, care and blessing of god, I finally succeed on doing my master project I from the start till the end.

Uncountable thanks goes to my supervisor, Dr. Norafida Ithnin for her time, guidance and advices throughout the project, her patience, kindness, and for her healthier supports when I needed. I would like to thank all of the FSKSM lectures for giving ideas and some advice regarding to the project.

My pleasure to appreciate my beloved parents; Mr. Ramamoorthy Applasamy and Mrs. Kanaka Latha Ambodai who support me with their love, prayers, moral and financial for my study. Thanks to my siblings, cousins and relatives who never fail to cherish and support me.

These special thanks go to my friends and my mentors, The Late Nurulhaini Anuar and Irda Roslan for always support me. My heart overflows with gratitude for all my friends for being supportive and understanding especially my hostel mates, course mates and my fellow postgraduate colleagues.

I would like to extend my appreciation to those who involved and give a helpful hand in ensuring the success of this project. May God bless you.

## ABSTRACT

Grid computing means a multiple independent computing, because it is composed of resource nodes not located within a single administrative domain. The goal of grid is to only provide secure grid service resources to legal users. Even though grid computing is more than just a technology to abet high performance computing, it is still have some issues to concerns and cares. One of the issues is security issues. Authentication is important part in grid security. Other process in grid are depends on authentication. The aim of this project is to enhance the method of password based authentication scheme and to get better password based authentication scheme in grid computing environment through its time complexity. In this project, the study is done on the existing grid security infrastructure and existing password based authentication scheme. Password Enable Certificate Free Grid Security Infrastructure (PECF-GSI) and A Simple User Authentication Scheme has been selected as the reference for the enhanced authentication scheme. Comparative study and pre-lab testing on A Simple User Authentication Scheme and PECF-GSI has been done in the research methodology. Finally, the enhanced authentication scheme has been designed, developed and tested based on four time complexity notations that are time for modular multiplication, time for multiplication of a number and an elliptic curve point, time for hashing operation and time for inversion. This project has achieved the aim, the scope and the objectives of the project by showing a good performance in terms of time complexity.

## ABSTRAK

Pengkomputeran grid bermaksud pelbagai pengkomputeran yang tidak bersandar, ini kerana grid mempunyai pelbagai nod sumber yang tidak terletak hanya dalam suatu domain pentadbiran. Matlamat grid adalah memberikan perkhidmatan sumber grid yang selamat kepada pengguna yang sah. Walaupun pengkomputeran grid jauh lebih hebat berbanding pengkomputeran keberkesanan tinggi, ia masih mempunyai isu-isu yang perlu diambil kira. Salah satu isu adalah isu keselamatan. Pengesahan merupakan isu yang penting dalam keselamatan pengkomputeran grid. Proses lain di dalam grid bergantung kepada pengesahan. Matlamat kajian ini adalah melakukan penambahan keatas kaedah metod skim pengesahan berasaskan katalaluan dan mendapatkan skim pengesahan berasaskan katalaluan yang lebih baik bagi pengkomputeran grid menerusi masa kerumitannya. Dalam kajian ini, carian dan perbandingan dilakukan terhadap infrastruktur keselamatan grid yang sedia ada dan juga skim pengesahan berasaskan katalaluan yang sedia ada. Infrastruktur keselamatan grid yang membenarkan katalaluan dan tidak perlukan sijil (PECF-GSI) dan Sebuah Skim Pengesahan Pengguna yang Mudah bagi Pengkomputeran Grid dipilih sebagai rujukan bagi skim yang hendak ditambah. Kajian perbandingan dan pra-ujian makmal terhadap Skim Pengesahan Pengguna yang Mudah dan PECF-GSI telah dilaksanakan dalam metodologi kajian. Akhir sekali, model skim penambahan direka, dibangunkan dan diuji menggunakan empat notasi kompleksiti masa. Iaitu, masa untuk modular pendaraban, masa untuk pendaraban suatu nombor dengan lengkungan elliptic, masa untuk operasi hash, dan masa untuk operasi penyongsangan. Projek ini telah mencapai skop dan objektif dengan menunjukkan prestasi yang baik.

## TABLE OF CONTENTS

<b>CHAPTER</b>	<b>TITLE</b>	<b>PAGE</b>
	<b>DECLARATION</b>	ii
	<b>DEDICATION</b>	iii
	<b>ACKNOWLEDGEMENT</b>	iv
	<b>ABSTRACT</b>	v
	<b>ABSTRAK</b>	vi
	<b>TABLE OF CONTENTS</b>	vii
	<b>LIST OF TABLES</b>	xi
	<b>LIST OF FIGURES</b>	xii
	<b>LIST OF ABBREVIATIONS</b>	xiii
	<b>LIST OF APPENDICES</b>	xiv
<b>1</b>	<b>INTRODUCTION</b>	
	1.1 Introduction	1
	1.2 Problem Background	2
	1.3 Problem Statement	3
	1.4 Project Aim	4
	1.5 Project Objectives	4
	1.6 Project Scope	5
	1.7 Significance of Project	6
	1.8 Organization of the Project	6



	Server Public Key	
	2.8.5 Authentication Scheme Based on Elliptic Curve Cryptosystem	43
	2.9 The Summary of Password Based Authentication Scheme	46
	2.10 Conclusion	49
<b>3</b>	<b>METHODOLOGY</b>	
	3.1 Introduction	50
	3.2 Research Approach	51
	3.2.1 Quantitative Method	51
	3.2.2 Qualitative Method	52
	3.3 Research Study	53
	3.3.1 Comparative Study	53
	3.3.2 Lab Experiment	54
	3.4 Operational Framework	55
	3.4.1 Phase I: Literature Review	57
	3.4.2 Phase II: Comparative Study and Pre-lab testing	58
	3.4.3 Phase III: Design. Develop and Testing	61
	3.4.4 Phase IV: Result	61
	3.5 Hardware and Software Requirements	62
	3.5 Conclusion	62
<b>4</b>	<b>COMPARATIVE STUDY AND PRE-LAB TESTING</b>	
	4.1 Introduction	63
	4.2 The Findings for Comparative Studies	64
	4.2.1 Features of Existing Grid Security Infrastructure	64
	4.2.2 Features of Existing Password Based Authentication Schemes	66
	4.2.3 The Features of Security Properties	67

4.3	The Lab Experiment Findings	69
4.3.1	Time Complexity Notations	69
4.3.2	Lab Setup	72
4.3.3	The Findings of Pre-lab Testing	73
4.4	Conclusion	78
<b>5</b>	<b>LAB TESTING AND RESULT ANALYSIS</b>	
5.1	Introduction	80
5.2	The Development of Suitable Grid Security Infrastructure	81
5.3	The Design and Development of Enhanced Password Based Authentication Scheme	83
5.3.1	The Enhancement on Elliptic Curve Cryptosystems	83
5.3.2	The Enhancement on Hash Function	86
5.4	Lab Testing on Enhanced Password Based Authentication Scheme	87
5.5	Conclusion	95
<b>6</b>	<b>DISCUSSION AND CONCLUSION</b>	
6.1	Introduction	96
6.2	Result and Achievements	96
6.3	Limitations of the Project	98
6.4	Future Works	99
6.5	Conclusion	99
	<b>REFERENCES</b>	100
	<b>APPENDIX</b>	102

## LIST OF TABLES

<b>TABLE NO</b>	<b>TITLE</b>	<b>PAGE</b>
2.1	The components of grid environment and its description	9
2.2	Definitions from the major organizations and system vendors	11
2.3	Grid security infrastructures	30
2.4	Existing password based authentication scheme	46
3.1	The example of comparative study table	54
4.1	The variety features of grid security infrastructures	65
4.2	Features of password based authentication schemes	66
4.3	Security properties for Yoon's and Rongxing's scheme	68
4.4	Time complexity notations and its definitions	70
4.5	Estimation of performance aimed at time complexity	71
4.6	Result of the first 10 iterations at user site	73
4.7	Result of the first 10 iterations at server site	74
4.8	Time complexity and rough estimation at user site	75
4.9	Time complexity and rough estimation at server site	76
5.1	Results for the time complexity notations at user site	88
5.2	Results for the time complexity notations at server site	89
5.3	Comparison of time complexity notations	90
5.4	Time complexity and rough estimation at user site	91
5.5	Time complexity and rough estimation at server site	92

## LIST OF FIGURES

<b>FIGURES NO</b>	<b>TITLE</b>	<b>PAGE</b>
2.1	Abstract evolution of the grid computing technology	12
2.2	Grid computing concern areas	14
2.3	Grid security issues	17
2.4	Typical grid scenario	24
2.5	A conceptual view of PECF-GSI	28
2.6	Steps for logging into the grid	34
2.7	Example of mutual authentication	35
3.1	Flow chart of operational framework	56
3.2	Comparative studies on the grid security infrastructure	58
3.3	Comparative study on the password based authentication schemes	59
3.4	Pre-lab testing on the existing password based authentication scheme	60
4.1	Host setup for pre-lab and lab testing	72
4.2	Time complexity notations for user site	77
4.3	Time complexity notations for server site	78
5.1	Part of algorithm for Trust Authority	81
5.2	The hierarchical relationships between user host and TA	82
5.3	Algorithm for elliptic curve point using supersingular curve	85
5.4	Algorithm for hash extension generation and verification	86
5.5	Time complexity at user site	93
5.6	Time complexity at server site	94

**LIST OF ABBREVIATIONS**

CA	-	Certificate Authority
CPU	-	Central Processing Unit
DoS	-	Denial of Server
ECC	-	Elliptic Curve Cryptosystem
GGF	-	Global Grid Forum
GridSim	-	Grid Simulator Toolkit
GSI	-	Globus Security Infrastructure
GT4	-	Globus Toolkit 4.0
KDC	-	Key Distribution Center
OGSA	-	Open Grid Standards Architecture
OTP	-	One Time Password
P2P	-	Peer-to-Peer
PBGSI	-	Password Based Grid Security Infrastructure
PECF-GSI	-	Certificate Free Grid Security Infrastructure
PIN	-	Personel Identity Number
PKI	-	Public Key Infrastructure
QoS	-	Quality-of-Service
SOAP	-	Simple Object Access Protocol
SSO	-	Single Sign-on
TA	-	Trusted Authority
TLS	-	Transport Layer Security
VO	-	Virtual Organization

**LIST OF APPENDICES**

<b>APPENDIX</b>	<b>TITLE</b>	<b>PAGE</b>
A1	Protocol of Password Authentication Schemes without Server Public Key	102
A2	Protocol of Efficient Password Authentication Schemes without Server Public key	104
A3	Protocol of Authentication Scheme Based on Elliptic Curve Cryptosystem	106
A4	Result for Selected Password Based Authentication Scheme	108
A5	Result for Enhanced Password Based Authentication Scheme	110

# CHAPTER 1

## INTRODUCTION

### 1.1 Introduction

Grid computing in general is a special type of parallel computing which relies on complete computers with onboard Central Processing Unit (CPU), storage, power supply, network interface and more connected to a network like private, public or the Internet by a conventional network interface, such as Ethernet. The analogy of grid can be described as below.

When a user plugs an appliance or other object requiring electrical power into a receptacle, the user expects that there is power of the correct voltage available, but the actual source of that power is not known. Any local utility company provides the interface into a complex network of generators and power sources and provides the public with an acceptable quality of service for public energy demands. Rather than each house or neighborhood having to obtain and maintain its own generator of electricity, the grid infrastructure provides a virtual generator and it is a highly reliable generator.

Grid technology also enables complex interactions among computational and data resources. The sharing of the resources in grid computing will increase the range of computer applications. This sharing may involve not only file exchange but also direct

access to computers, software, data, and other resources, as is required by a range of collaborative problem solving and resource-brokering strategies emerging in industry, science and engineering. When a user wants to request some computing and data resources, the grid can seamlessly, transparently and dynamically supply the resources over Internet.

## 1.2 Problem Background

The sharing process in grid is necessarily and highly controlled with resource providers and users or consumers defining clearly and carefully just what is shared, who is allowed to share, and the conditions under which sharing occurs. The controlling process is important because the goal of grid computing is to only provide secure grid service resources to legal users (Lu. R. *et al*, 2007). This statement clears out that, the security issues becomes an important concern of grid computing.

To prevent the illegal users from visiting the grid resources, it should be guaranteed that strong mutual authentication needed for users and servers. Authentication is believed, as a cornerstone service, since other services depend on the authentication of communication entities. As mentioned above, the grid systems should be guaranteed that stronger authentication needed for users and servers and ensure that resources and data not provided by an attacker. Generally, the authentication in the grid computing aims to:-

1. Allow a user, the processes that comprise a user's computation, and the resources used by those processes, to verify each other's identity.
2. Guarantee the safety of the information in grid.

Beside these aims, two main issues of authentication process must be considered. First is the security of the authentication and second is the time complexity of the authentication. The security of the authentication can be found by analyze the security properties which consist of some security attacks. The authentication process must resist some security attack to fulfill the grid's aims.

Authentication process must control the time complexity, whereby the time complexity is important to have efficient authentication process. The time consuming of authentication can be found by analyze the time of complexity and the types of computation.

### **1.3 Problem Statement**

Grid computing, as a distributed computing model, stands for the new kind of systems that combine heterogeneous computational resources, such as computers, storage space, sensors, application software, and experiment data, connected by the Internet and make them easy access to a wide user community.

From the problem background for flexibility in grid computing the authentication framework should strike a proper balance between the entirely demands of the security and the access speed. Fluent choosing different security mechanisms based on different demands is a powerful solution. The authentication framework also should be designed to have the interoperability with local security solutions, and apply the local access control mechanism without change to get lightweight grid environment.

Beside that, the authentication should not be bundled with any concrete and fixed mechanism of security, because of the dynamic nature of the grid (Chen.J. 2006), even though certificate based authentication more appropriate and secure than password based authentication (Chakrabarti.A. 2007).

#### **1.4 Project Aim**

This project aimed to enhance the method of password based authentication scheme and to get better password based authentication scheme in grid computing environment through its time complexity.

#### **1.5 Project Objectives**

To achieve the aim of the project, there are three major objectives have to be fulfilled:

1. To analyze the characteristics of grid computing environments, the security challenges in grid computing, the existing password based authentication scheme, the existing grid security infrastructure.

2. To design and develop the enhanced password based authentication scheme that will secure the grid computing environment using selected existing authentication scheme.
3. To test and implement the enhanced authentication scheme using lab testing.

## **1.6 Project Scope**

1. The study focused on the existing authentication scheme and especially on password based authentication scheme.
2. The features selection of password based authentication scheme done by comparative studies and lab experiment.
3. The grid security infrastructure involve in this study should support password based authentication.
4. The development is using C++ programming language.

## **1.7 Significance of Project**

This study first identified the overview of grid computing including the evolution of grid computing and the concern areas of the grid. The security issues of grid computing also will identified. The next step is to find the existing authentication schemes and existing grid security infrastructures that used for grid computing authentication process. By develop and perform lab testing, a new way of securing the grid computing through authentication is then enhanced and compared to the selected existing scheme of password based authentication system to see whether this approach can give a better solution. The enhanced scheme must give better result rather than the existing scheme.

## **1.8 Organization of the Report**

This report consists of six chapters. First chapter presents introduction to the project which includes the problem background, problem statements, aim of project, the main objectives and scope of the project. Chapter 2 is about literature review on grid computing, which focused on identifying the security challenges in grid computing, the existing schemes and its security infrastructure of password based authentication schemes. The project methodology is covered in Chapter 3 where comparative study and pre-lab testing have been used as the research strategy. In Chapter 4, the implementation of the methodology where the findings of comparative study and pre-lab testing take place and the result and findings of the lab testing is explained in Chapter 5. The overall project will concluded in Chapter 6.

## **CHAPTER 2**

### **LITERATURE REVIEW**

#### **2.1 Introduction**

As covered in previous chapter, the aims of this project is to improve the method of authentication scheme and get better authentication scheme in grid computing environment. In this chapter, some definitions and overview of grid computing is explained to understand the literature of grid computing. Beside that, the taxonomy of grid security issues are identifies. From that, authentication is one of the main security problems. So, the definitions of authentication and different types of authentication schemes in grid computing are included. Description about Grid Security Infrastructure also included in this chapter. Followed after that, authentication in grid security infrastructure is explained. After that, the password based authentication schemes was describe briefly because this project was concern on this part.

## 2.2 Overview of Grid Computing

Grid computing is a phrase in distributed computing which formally means a multiple independent computing clusters which acts like a 'grid' because they are composed of resource nodes not located within a single administrative domain. Grid computing can be used in a variety of ways to address various kinds of application requirements.

There are three types of grids available. First is computational grid where it is focused on setting aside resources specifically for computing power. High performance servers were used in computational grid. Next is scavenging grid, this type of grid commonly used with large numbers of desktop machines, and machines are scavenged for available CPU cycles and other resources. The last one is data grid which responsible for housing and providing access to data across multiple organizations.

Before go further, the understanding of primary components of grid environment is important. Table 2.1 shows the components of grid environment and its description.(Jacob.B. 2006).

**Table 2.1:** The components of grid environment and its description

Components	Description
Portal / user interface	A grid portal provides the interface for a user to launch applications that will use the resources and services provided by the grid. From here, user sees the grid as a virtual computing resource.
Security	Grid environment must be mechanisms to provide security, including authentication, authorization, and data encryption. Grid Security Infrastructure component of the Globus Toolkit provides robust security mechanism.
Broker	This component helps to identify the available and appropriate resources to use within the grid after the authentication process.
Scheduler	Scheduler is used to execute the jobs, no matter the jobs are executed stand-alone or concurrently.
Data management	Needs to move data because grid computing needs to be a secure and reliable method for moving files and data to various nodes within grid.
Job and resource management	Help to launch a job on a particular resource, check its status, and retrieve its results when it is complete.

There are nine main components of grid environment; portal/user interface, security, broker, scheduler, data management, job and resource management. The grid portal can provide the interface for a user to launch applications that will use the resources and services which provided by the grid. Grid environment must be providing security components, including authentication, authorization, and data encryption. Grid must have grid security infrastructure to provide robust security mechanism.

Broker helps to identify the available and appropriate resources to use within the grid after the authentication process. Scheduler is used to execute the jobs, no matter the jobs are executed stand-alone or concurrently. Data management component needs to move data because grid computing needs to be a secure and reliable method for moving files and data to various nodes within grid. Job and resource management help to launch a job on a particular resource, check its status and retrieve its results when it is complete.

The term grid computing originated by Ian Foster and he is one of the earliest proponents of grid technology. Foster defined that grid as “A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive and inexpensive access to high end computational capabilities” (Foster. I.1998).

The definition was changed by include some element of social and policy issues, stating that grid computing is concerned with “coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations” (Foster. I. *et al.* 2000) Grid technology was defined as the technology that enables resource virtualization, on-demand provisioning, and service or resource sharing between organizations (Plaszczak *et al.* 2005).

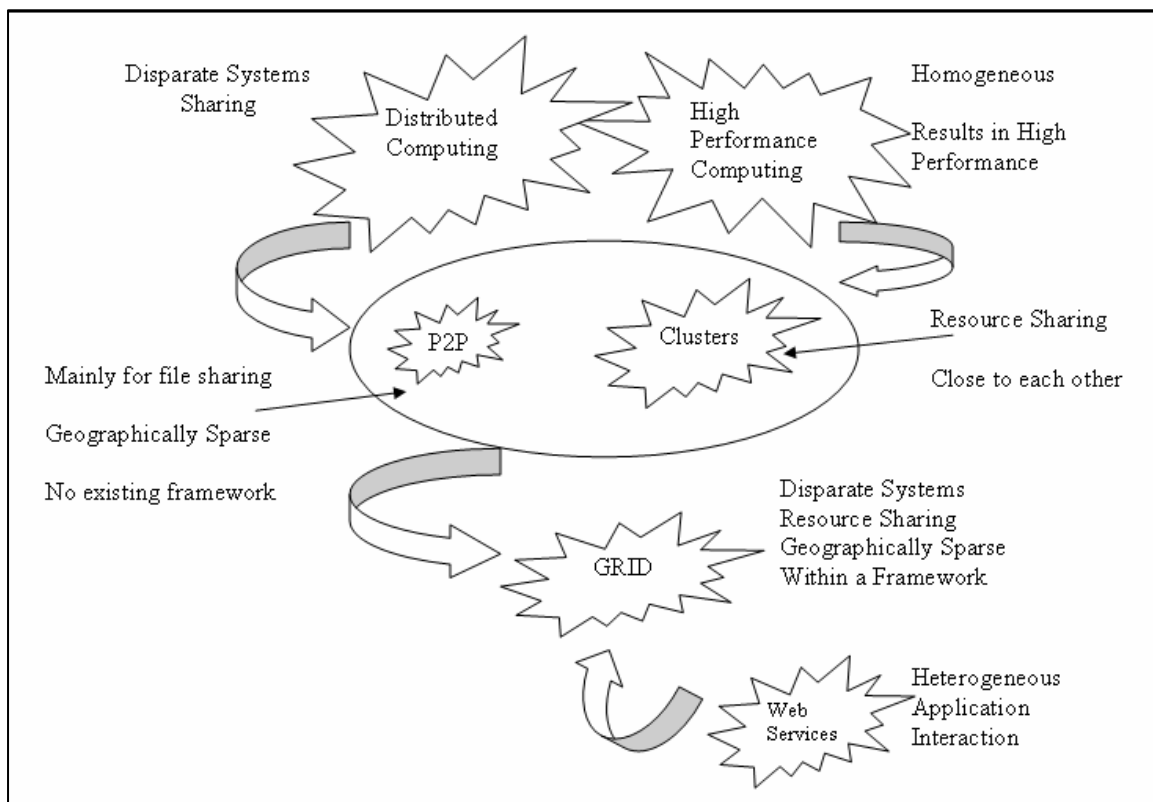
Grid also was defined as a type of parallel and distributed system that enables the sharing, selection, and aggregation of geographically distributed autonomous resources dynamically at runtime depending on resources availability, capability, performance, cost and users' quality of service requirements. Beside the academicians, some system vendors who have grid strategy also have their own definitions (Buyya. 2006). Table 2.2 shows the definitions from the major organizations and system vendors.

**Table 2.2:** Definitions from the major organizations and system vendors

Organizations	Definitions
Oracle	An adaptive software infrastructure which is able to balance resources efficiently through the usage of low cost servers and storage.
Sun Microsystems	Divide the grid into 3 <ul style="list-style-type: none"> <li>• Cluster grids – the resources within a local area network are shared.</li> <li>• Enterprise grids – the resources within an enterprise are shared</li> <li>• Global grids – a grid across enterprise are shared</li> </ul>
HP	Grid is a Type of parallel and distributed system that enables the sharing, selection, and aggregation of resources distributed across 'multiple' administrative domains based on resources availability and capacity. Another term is Utility Computing.
IBM	The ability, using a set of open standards and protocols, to gain access to applications and data, processing power, storage capacity and a vast array of other computing resources over the Internet. In other word it is called autonomic computing.
CERN	A service for sharing computer power and data storage capacity over the Internet.

There are some consistent themes from the academics' and vendors' definitions. The key of grid definitions are some sort of network of computing resources with limited amount of user intervention and done quickly, reliably and cheaply. From the various definitions, Grid computing defined as a hardware and software infrastructure that allows services oriented, flexible, and seamless sharing of heterogeneous network of resources for compute and data intensive tasks and provides faster throughput and scalability at lower costs (Chakrabarti.A, 2007).

According to grid computing definitions, grid computing is not a scratch based technology, but it is conglomeration of different existing technologies like cluster computing, peer-to-peer (P2P), and Web services technologies. Figure 2.1 shows an abstract evolution of the grid computing technology from the P2P and clusters and the possible marriage of the grid with the web services technologies.



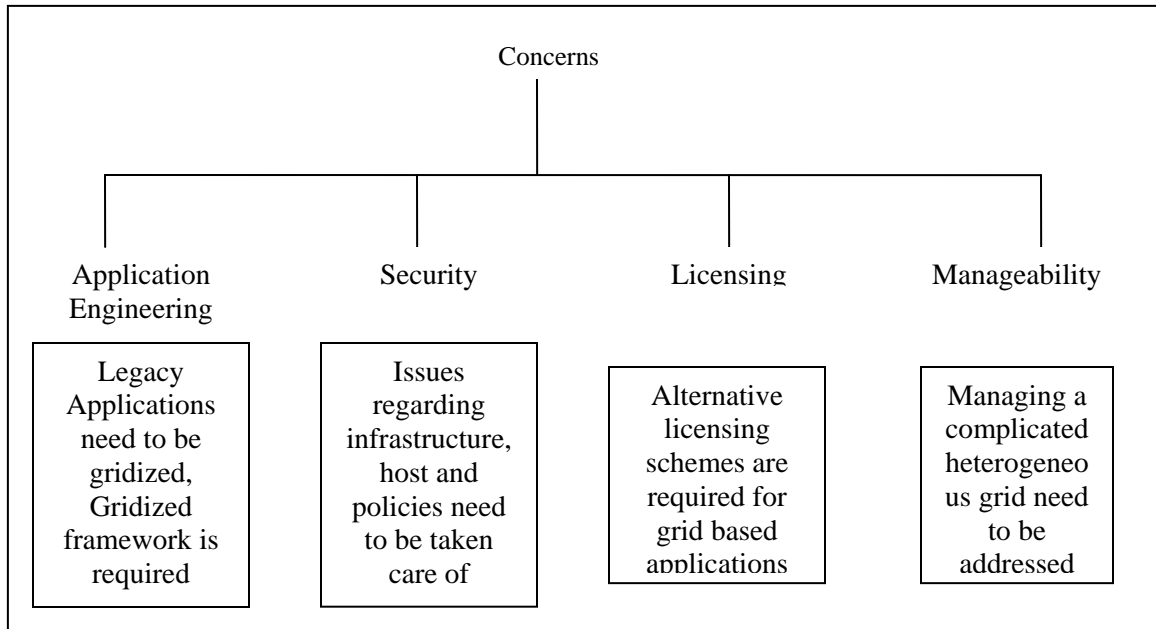
**Figure 2.1:** Abstract evolution of the grid computing technology

The evolution of grid technology starts when the distributed computing and the high performance computing were merged. In distributed computing, the disparate sharing system was used. After merge of these two computing, peer to peer (P2P) and clusters computing was found. P2P mainly used for file sharing and it is geographically sparse with no existing framework whereby the clusters computing is used for resource sharing and close to each other host. Web services provide heterogeneous system and also some application interaction. From the combination of P2P, clusters and Web services the Grid was found. Grid provides disparate systems, used for resource sharing. It is also geographically sparse within a framework (Chakrabarti.A, 2007).

The benefits of using grid was categorized into four types, there are performance and scalability, resource utilization, management and reliability, and virtualization. In terms of performance and scalability, grid computing solutions of having a shared infrastructure provide more computational capabilities and increase scalability of the IT infrastructure. Another pertinent grid imperative is the need to utilize the IT resources more efficiently. Grid computing offers a mechanism to utilize the resources more efficiently through the resource sharing process. The attractive part in grid computing is its ability to share resources across geography.

As the IT infrastructure grows, the systems become more complex and heterogeneous. Therefore, the issue of management becomes critical. Grid computing provides a single interface for managing the heterogeneous resources and it's reduced an integrated management environment. Grid computing allow IT infrastructure to create more robust and resilient infrastructure to respond to minor or major disasters. The grid provides virtualization of heterogeneous resources resulting in better management of the resources.

Even though grid computing is more than just a technology to abet high performance computing and have a lot of benefits but its still have some issues and concerns to cares. Figure 2.2 shows the concern areas of grid computing.



**Figure 2.2:** Grid computing concern areas

Most of the grid users are from areas where they have to handle huge amounts of data and computational involved like life sciences, finance, automotive and aerospace, and more. There are no tools, frameworks, or platforms to help these users to gridize their applications in order to get performance benefits without putting too much investment. Gridization has two aspects:

1. Data can be manipulated, striped across the grid for enhanced performance called data engineering.

2. Manipulating the applications themselves so that they are able to extract maximum benefit out of the grid computing infrastructure and it's called application engineering. Most of the tools and techniques in data engineering and application engineering are insufficient for enterprise needs.

IT systems were besotted with problems like scheduling, management, security and other challenges. To solve these kinds of problems, substantial work has been carried out at different levels. It is becoming clear that this evolutionary growth of the technology occur to complexity and manageability problem. Manageability problem is related to the integration because grids bring together software components, frameworks, and middleware and hardware elements. So to integrating they together will become a problem.

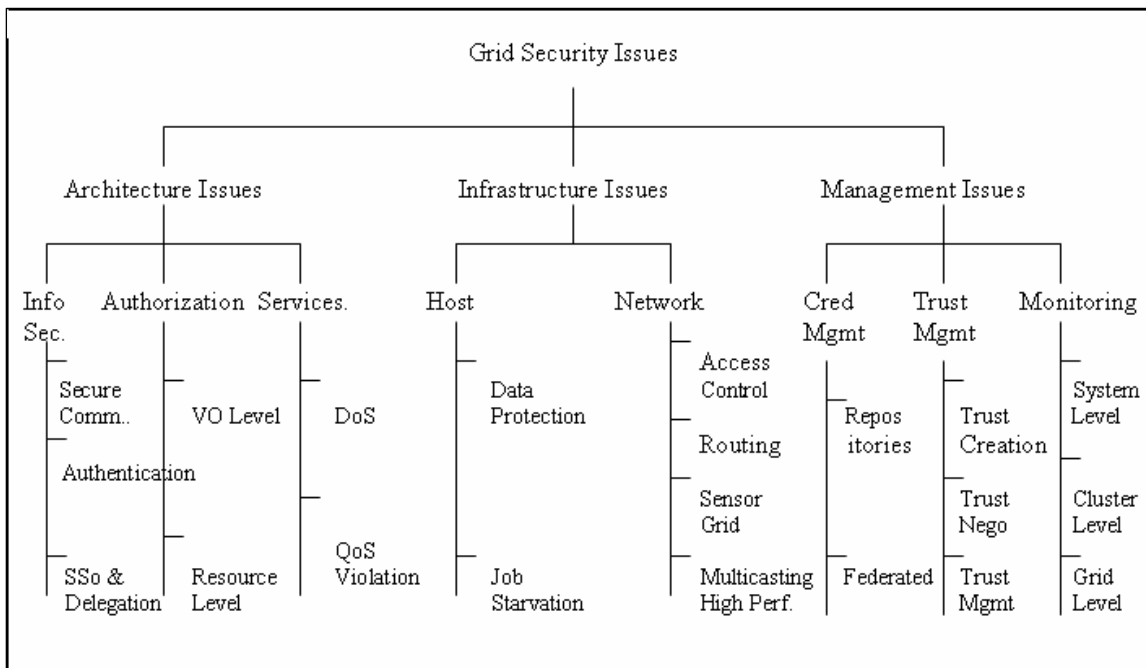
The grid allows the sharing resources across different systems. Even this sharing is good but by licensing needs of the applications sharing resources is bad. Grid must develop suitable pricing and licensing infrastructure. Lastly the security issue also included in one of the concern areas of grid computing. Beside common security challenges like authentication, confidentiality and integrity; grid has other unique security challenges such as policy integration, authorization, and credential management. The following section explained the security issues in grid.

## **2.3 Security in Grid Computing**

In traditional systems the security are to protect a system from its users and protect data of one user from compromise. Anyhow in Grid Systems the security issues are different. First, grid systems must protect applications and data from system where computation executes. Then, grid computing must need strong authentication for users and code. Grid systems also must protect local execution from remote systems, and finally, grid computing needs different admin domains and also security policies (Neuman C, Grid Book).

### **2.3.1 Security Issues in Grid Computing**

Grid computing helps to overcome heterogeneity in terms of computing elements, operating systems, policy decisions, and environments. However, its security issues impede it from adopting as a widespread IT virtualization solution. Chakrabarti.A. (2007) was categorized the grid security issues into three main categories. Figure 2.3 shows the categorization of the different security issues in a grid.



**Figure 2.3:** Grid Security Issues

The three categories of grid security issues are architecture related issues, infrastructure related issues and management related issues. Architecture related issues concerns pertaining to the architecture of the grid. In this level users of grid are concerned about the data processed, and the protection of the data confidentiality and integrity and also user authentication. These requirements are categorized under information security. Beside that, resource level authorization and Service of the grid such as Quality-of-Service (QoS) also include in architecture related issues.

Infrastructure related issues relate to the network and host components. Issues that make a host apprehensive about affiliating itself to the grid system are under host level security issues. The sub issues in this level are job starvation and data protection. A host in grid concerned about the jobs that is running locally so, the external jobs should not reduce the priority of the local jobs and lead to job starvation.

The last issues are about management related issues which pertains to the management of grid. As mention above, grid system are using heterogeneous grid infrastructure and applications, therefore managing credentials is absolutely important. Other than credentials management, like any other distributed system, managing trust is also critical requirements in management related issues. In concern of auditing purposes, grids require some amount of resource monitoring. Much of the information obtained from the monitoring systems is fed back to higher level systems like intrusion detection and scheduling systems.

### **2.3.1.1 Architecture Related Issues**

As mention above, architecture level issues concern of the grid system as a whole. Issues like Information security, authorization, and service level security generally destabilize the whole system. This subsection briefly described about the issues.

#### *Information Security*

Grid's information security issues was classify as secure communication, authentication, single sign-on (SSO) and delegation (Chakrabarti *et al.* 2007). Secure communication issues are concerns on security problem when arise during communication between two entities such as resources and users in grid system. These include some of the security goals like confidentiality, authentication and integrity.

There are also issues related to authentication, where the identities of entities involved in the overall process can be accurately asserted. This issue becomes critical in grid computing because of the heterogeneous and distributed nature of the entities involved. Users of grid computing also concerned about the single sign on capability which provide by grid computing infrastructure. In single sign on, the authentication is done once.

Even information security issues become common problem in all fields of computing and communication, in grid computing area researchers and practitioners developed Grid Security Infrastructure to define secure grid systems. The Global Grid Forum (GGF) (now called OGF) released an open standard called Open Grid Standards Architecture (OGSA). Grid security infrastructure is a layer of OGSA which addresses most of the information security challenges mentioned above. Grid security infrastructure adopted as a standard and Globus as an open source implementation of OGSA. More details about GSI were described in section 2.5.

### *Authorization*

Another important security issues is authorization. Grid systems also require resource specific and system specific authorizations like any resource sharing system. Authorizations are important because the resources are shared between multiple departments or organizations, and department wide resource usage patterns are pre-defined.

The authorization systems divided into two levels. First is Virtual Organization or VO level system. VO level systems have a centralized authorization system which provides credentials for the user to access the resources. Furthermore, some VO level

systems are large and long-lived in case of explicit negotiations with resource provider; others will be short-lived which are created to support a single task (Weleh V, 2005). The second level is Resource level system. This systems, allow the users to access the resources based on the credentials presented by the users.

### *Service Security*

One of the important security threats in any infrastructure is the malicious service disruption created by adversaries. For example, in the Internet space servers and networks are brought down by a huge amount of network traffic and users are denied the access to a certain Internet based service.

The service level attacks are currently nonexistent because in grid computing the deployment still in under control level or in generic has not reached the critical level yet. The grid service level security issues can split into two main types, there are QoS Violation Issues and Denial-of-Service related issues. QoS violation was forced by the adversary through congestion, delaying or dropping packets, or through resource hacking. The Denial of Server (DoS) is more dangerous where the access to a certain service is denied.

## **2.4 Authentication in General**

Authentication is the process by which a subject proves its identity to a requestor, typically through the use of credential. In other word, authentication mechanisms are used to ensure that the entity that supposedly sent message to another party is indeed the legitimate entity. Authentication in which both parties, example like requestor and the requester, authenticate themselves to one another simultaneously is referred to as mutual authentication. A credential is a piece of information that is used to prove the identity of a subject. Passwords and certificates are examples of credentials. Authenticating a system or a user generally handled using three different mechanisms. Following subsection explained about these three authentications schemes (Chakrabarti.A. 2007).

### **2.4.1 Shared Secret Based Authentication**

This first scheme is through sharing a secret. It's similar like bank PIN number supposed known only by that individual person when he/she want to get details of their account. The implementation of this kind of system is to share a password between the authenticator and the user. In this type of system, the authenticator asks the user for a password which when disclosed will allow the user to enter the system. This system is perhaps the most prevalent mechanism of authentication that is used. These systems are simple to implement and computationally inexpensive because the password supplied by the user is checked with hash of the password which is stored in database.

The checking process is simple. However this type of system is vulnerable in two ways. Firstly the password is sent unencrypted over the wire which can be easily tapped

by malicious adversary. Therefore, the password has to be encrypted by different mechanisms as mentioned earlier. Secondly, choosing the password itself is a difficult problem as automated tools are available which can guess a password with relative ease and accuracy. Therefore a password based system cannot be used where strong authentication is needed.

Through challenge mechanism the shared secret can be implemented. The authenticator would challenge the user to encrypt a bit of known information by using the shared key. So the user responds by encrypting the required information and is allowed to access the system once the encrypted information has been validated by the authenticator. The shared secret needs to be changed periodically so that the adversary cannot guess the secret. This system is more expensive than the password based system. The challenge based system has another vulnerability that is the man-in-the-middle attack. This vulnerability prevents the challenge based systems to be the sole mechanism of authentication and used in conjunction with the other mechanisms.

#### **2.4.2 Public Key Based Authentication**

In public key based authentication scheme the user has a public and private key pair and the authenticator knows the public key of the user. The user encrypts standard information with his/her private key. The authenticator can verify the authenticity by decrypting the same information with the user's public key. This scheme is very secure and tampers proof. The problem in this mechanism is the scalability of the system. For example, there are millions of users using this kind of schemes and the authenticator face difficulties to maintain the public information of so many users.

### **2.4.3 Third Party Authentication Schemes**

The authenticator in third party authentication scheme does not know the user, however, uses a third party credential for authentication purposes. This kind of mechanism is popular in digital systems. The user gets a digital certificate from Certificate Authority (CA) which is known third party. Information about the user hashed and then signed by the CA's private key. The public key of the CA is widely known therefore the authenticator has no problem in validating the certificate and hence authenticating the user to access the system based on the certificate.

However this kind of scheme mandates that each user has a public key which can be validated by the CA. Means there is a need for Public Key Infrastructure (PKI) to make this scheme work. However it's not feasible if millions of users using this mechanism. Another mechanism of third party authentication scheme is used in Kerberos system. Kerberos system is to have key distribution center (KDC) which authenticates the user using a standard mechanism like using password. The KDC generates a session key for the user to access the system encrypted with the systems public key.

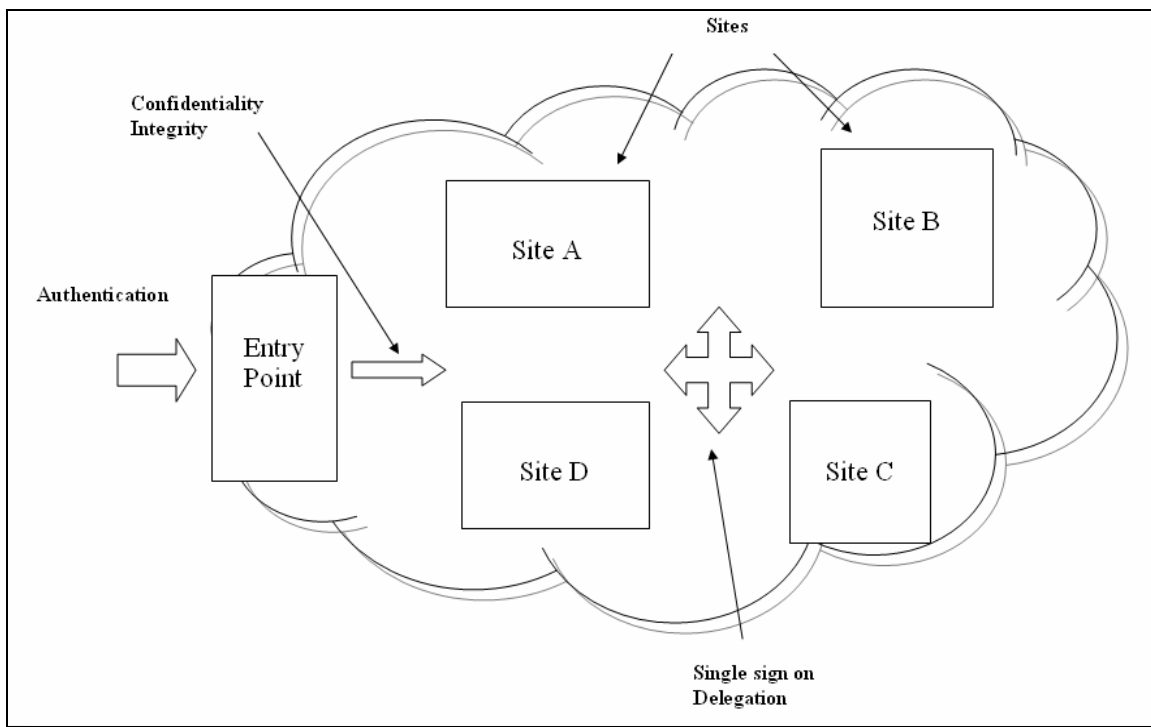
## **2.5 Grid Security Architecture**

The standardization effort of grid security has led to the design of security standards in grid which is defined under Grid Security Infrastructure. The driving force behind the generic standardization efforts in grid computing is the Global Grid Forum (GGF). GGF is a forum of researchers and practitioners for exchanging information and defining standards for grid computing. The open standard has been put forward by the GCF community is called Open Grid Standard Architecture (OGSA). OGSA define

mechanisms based on Web services for different systems to communicate and share the heterogeneous grid resources.

### 2.5.1 Grid Security Infrastructure

As mentioned earlier, a grid defines a concept called the Virtual Organization (VO). In a VO, different individuals, enterprises, organizations come together to share resources and services under a set of rules or policies guiding and governing the extent and conditions of sharing. The main aspect that separates grid systems from all the different systems are the heterogeneity involved and policy complications. Figure 2.4 shows a typical grid scenario consisting of sites which constitute a VO.



**Figure 2.4:** Typical grid scenario

A user submits a job to the grid which arrives at the entry point of the grid system. The mechanism to authenticate the user was implemented at this point. After the job submitted to the grid then grid system need to provide confidentiality and integrity so that no one is able to see the contents of the information and modify the contents. Finally, single sign on and delegation mechanism must be applied. Security requirements in grid security infrastructure are explained below.

First security requirement in grid security infrastructure is authentication. Authentication mechanism applied at entry points. Different authentication mechanism can be applied. In supporting many types of authentication mechanism, the security protocol should be flexible and scalable to handle all the different requirements and provide a seamless interface to the user.

Grid security requirement should protect the confidentiality of the messages and the documents that follow over grid security infrastructure. Similar to authentication mechanisms there may be needed to define, store, and share security contexts across different entities. Integrity also is one part of security requirement in grid security infrastructure. Its means that any changes made to the messages or the documents can be identified by the receiver.

In a grid environment, there may be instances where requests may have to travel through multiple security domains. Therefore, there is need for single sign-on facility in the grid infrastructure. There also may be need for services to perform actions on the user's behalf. A computational job may require accessing database many times. In this case there needed to delegate the authority to some service which will perform the action on the user's behalf.

## 2.5.2 Types of Grid Security Infrastructure

There are several types of grid security infrastructure, but this project will focused on three grid security infrastructure. There are Globus Security Infrastructure (GSI), Password Based Grid Security Infrastructure (PBGSI) and Certificate Free Grid Security Infrastructure (PECF-GSI).

### 2.5.2.1 Globus Security Infrastructure

The Globus Toolkit's Grid Security Infrastructure (GSI) is a set of libraries and tools that allow user and applications to access resources securely. GSI focused on authentication and message protection. GSI is based on Public Key Infrastructure (PKI) and uses authentication credentials composed of X.509 certificates and private keys (Pearlman.L. *et al.* 2005).

In brief, GSI user generates a public private key pair and obtains an X.509 certificate from a trusted entity called a Certificate Authority (CA). GSI uses temporary credentials called proxy credentials. GSI implements its authentication process via the TLS protocol in a GSS-API library. GSI also allows the user to delegate a proxy credential to a process on a remote host.

GSI provides mechanisms for translating a user's GSI identity to a local identity. The GSI identity is the name from the user's certificate and the local identity is Kerberos principle or local UNIX account.

### 2.5.2.2 A Password Based Grid Security Infrastructure

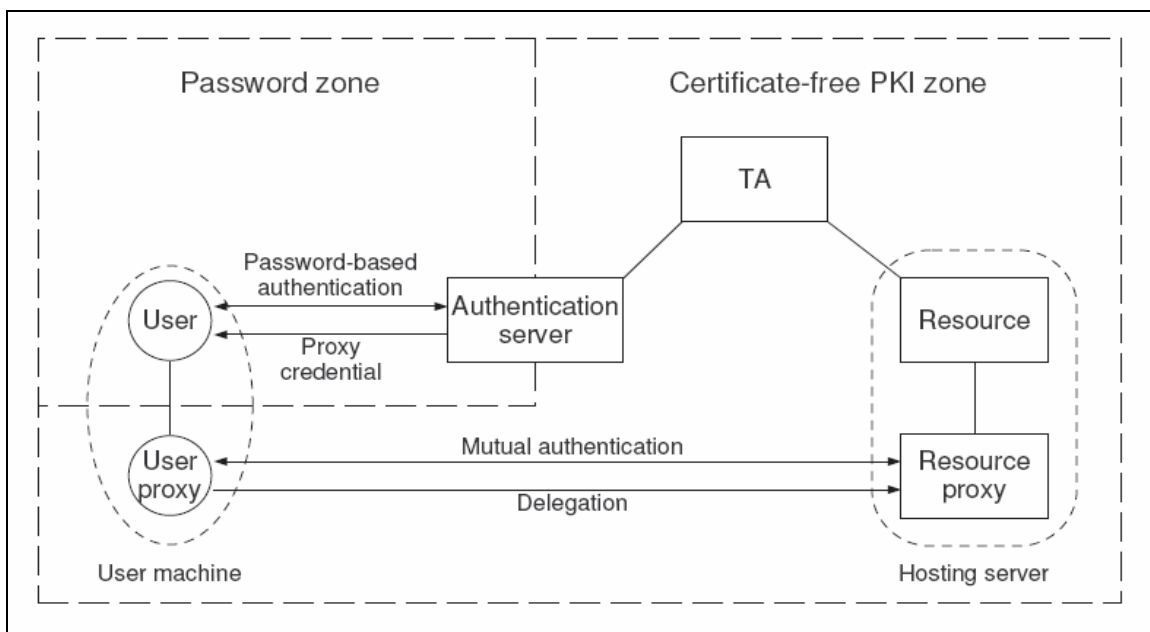
A password based security infrastructure (PBGSI) was proposed by Zhun Cai in year 2008. The authorization in PBGSI is based on the user's password. The server keeps a copy which is transformed from the password. An access control list is used to map the user's grid identity to an account of local system. The user may delegate to the server the right for that server to act on the user's behalf.

PBGSI provide key exchange with mutual authentication using weak secrets like passwords instead of public key certificates. The authentication and key exchange protocol in PBGSI is based on Authenticated Key Exchange (AKE). Subsequently, grid authorizes computing resources to authenticated users by mapping grid user ID to accounts of local system. At the same time, the user may delegate a server the right for that server to act on the user's behalf. Chaffing and Winnowing schemes are used for handle the issue of secure data transfer.

In PBGSI, server generates a token which contains the user's grid ID and the authorization information after processes of mutual authentication. The user uses the token to access grid resources. The PBGSI also allows the user to delegate a short-term password to processes on remote hosts. PBGSI used secret passwords to protect the authentication process against Man-in-the-middle Attack and uses time stamp mechanism to defend DoS attack and Replay Attack.

### 2.5.2.3 Password Enable Certificate Free Grid Security Infrastructure

Crampton *et al.* (2008) proposed Password Enable Certificate Free Grid Security Infrastructure (PECF-GSI). This security infrastructure allows the user only authenticated using passwords, with the authentication taking place between users and a centralized authentication server. This server plays a similar, but not identical, role to the MyProxy server in the PKI- based GSI or GSI. The benefit of PECF-GSI is neither the client nor server certificates are required during user authentication. PECF-GSI also completely removes the need for long-term user public keys, and hence the need for a revocation mechanism for these public keys too. Figure 2.5 shows the conceptual view of PECF-GSI.



**Figure 2.5:** A conceptual view of PECF-GSI

The PECF-GSI has two zones. One is a user-friendly zone, where only passwords are involved, called password zone. Another zone is a certificate free PKI zone, which is hidden from the users' view, and makes use of full-strength public key techniques. PECF-GSI employs a Trusted Authority (TA), instead of a CA, as the root of trust within

a grid environment. The TA's roles include acting as the Private Key Generator (PKG) in the identity based setting and providing a key management service.

User's long-term credential is simply a password, which shares with an authentication server. The authentication server is assumed to be accredited by the TA and hosting servers or resource providers within the grid environment. The authentication server and hosting servers must obtain the TA's authenticated parameter set through out-of-band mechanisms. A user proxy is a short-lived agent created by the user to perform security services on the user's behalf. Similarly, the resource proxy is created by provider to help manage a job submission from a user.

The light-weight and user-friendly grid security infrastructure can be found in PECF-GSI. Mutual authentication of a user and a server in PECF-GSI is based only on a provably secure password based authentication protocol. PECF-GSI still supports various grid security services, such as single sign-on, mutual authentication and delegation.

This grid security infrastructure is used to support essential security services for grid applications. Key agreement and delegation in PECF-GSI require much less bandwidth. In addition, the delegation technique in PECF-GSI requires only a single verification. The computational effort required by the key generation algorithms that are employed in PECF-GSI is lower than other security infrastructures.

## 2.6 The Summary of Grid Security Infrastructure

The summary of grid security infrastructure can be seen in Table 2.3 shows the three grid security infrastructures that involves in this study. These infrastructures are compared to see what approaches are been used, the strengths and the weaknesses of each of them.

**Table 2.3:** Grid security infrastructures

<b>Grid Security Infrastructure</b>	<b>Approaches</b>	<b>Strengths</b>	<b>Weaknesses</b>
Globus (GSI)	PKI based authentication; Use X.509 certificate; PKI based	Secure; Provide translating mechanism	Can't support password based authentication only, must include CA
PBGS	Secret based authentication; access control list is used to map the user's grid identity; authentication and key exchange protocols based on AKE	User delegate server to act on the user's behalf; Secure data transfer	Required timestamp
PECF-GSI	Authentication using password only;	Remove long-term user public keys; Support grid	Employs TA

	Used authentication server; Consists two zones, password zone and certificate free zone	security services; Required less bandwidth	
--	--	---	--

The approaches, the strengths and the weaknesses of three grid security infrastructures were clearly figure out in Table 2.3. All these can be summarize as below. In Globus Security Infrastructure, it is used third party authentication scheme like Certificate Authority (CA) technique. The authentication in GSI is PKI based authentication and used X.509 certificate. This infrastructure is more secure than other grid infrastructures. GSI also provide translating mechanism. The weaknesses of GSI is it is cannot support password based authentication scheme only.

The PBGSI is secret based authentication because it support password based authentication scheme. PBGSI also provide access control list to map the user's grid identity. The authentication and key exchange protocols are based on AKE. The strengths of PBGSI are user can delegate server to act on the user's behalf. PBGSI also provide secure data transfer. The major weakness of PBGSI is it is required timestamp.

The authentication in Password Enable Certificate Free Grid Security Infrastructure (PECF-GSI) is using password. PECF-GSI also used authentication server and provide two zones; password zone and certificate free zone. PBGSI totally remove long-term user public keys. The grid security services also were supported by PECF-GSI. It is also required less bandwidth among other infrastructures. PECF-GSI employs Trust Authority (TA) than CA.

The strengths elements among these existing grid security infrastructures that can guide in designing and developing a best practice grid security infrastructure would be the ability to be applied. The PECAF-GSI is chosen as a selected grid security infrastructure for complete this project.

## **2.7 Authentication in Grid Security Infrastructure**

In high level view of grid security infrastructure, there are three types of authentication. Using certificates, passwords, and using Kerberos are the types of authentication. For confidentiality mainly key based encryption algorithms are used. For single sign on/delegation proxy certificates are generally used.

This project is weights to authentication in GSI. In GSI authentication the most prevalent mechanism is the certificate based authentication mechanism but any how Kerberos and password based mechanisms also been implemented. Below is the descriptions about certificate based authentication, password based authentication and integration with Kerberos.

## 2.7.1 Certificate Based Authentication

Certificate based authentication mechanism has been implemented in all of Globus toolkit. It assumes that each user within the grid system possesses a public private key pair, and trusted third party or CA exists to sign and certify the users. Following information is included in the grid security infrastructure certificate.

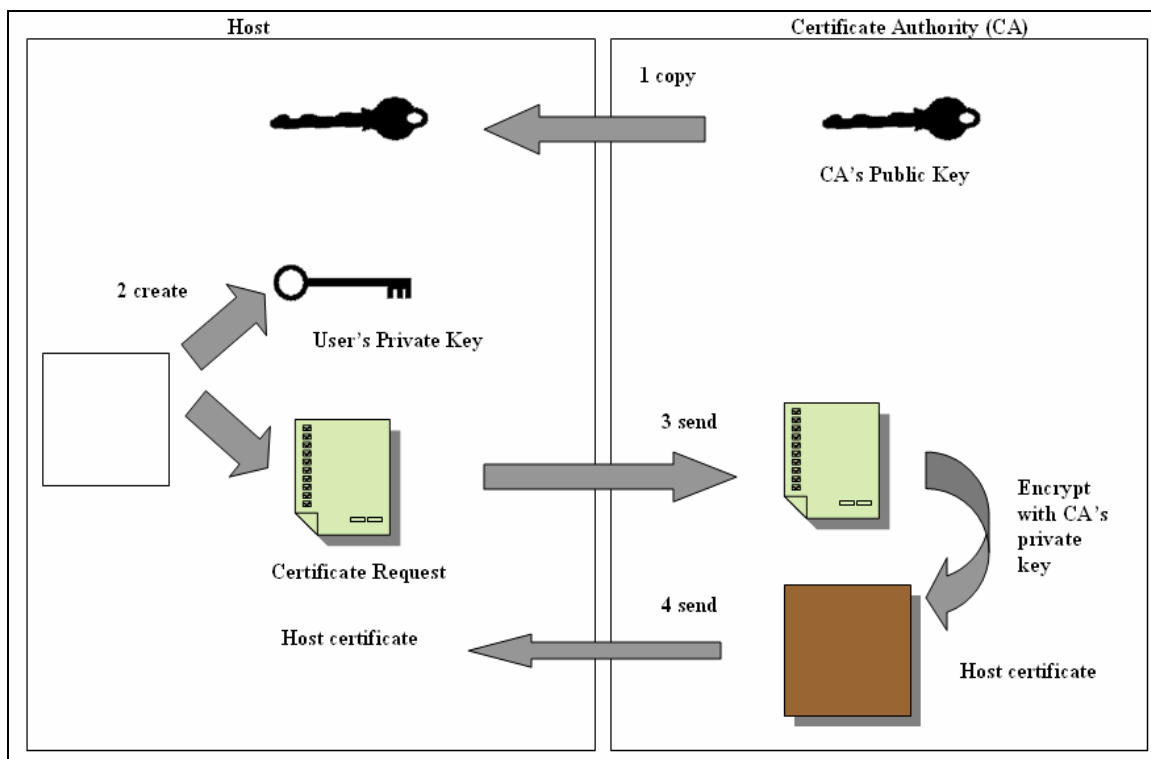
- A subject name, which identifies the person or object that the certificate represents.
- The public key belonging to the subject.
- The identity of a CA that has signed the certificate to certify that the public key and the identity both belong to the subject.
- The digital signature of the named CA.

### 2.7.1.1 Logging into the Grid System

There are four steps to logging into the grid system. Figure 2.6 shows the four steps of allowing a user to access a grid system using certificate based authentication in grid security infrastructure.

1. The first step is to know the public key of the CA. This information is used to verify the validity of the certificate obtained from CA. The certificate is stored in the local host.

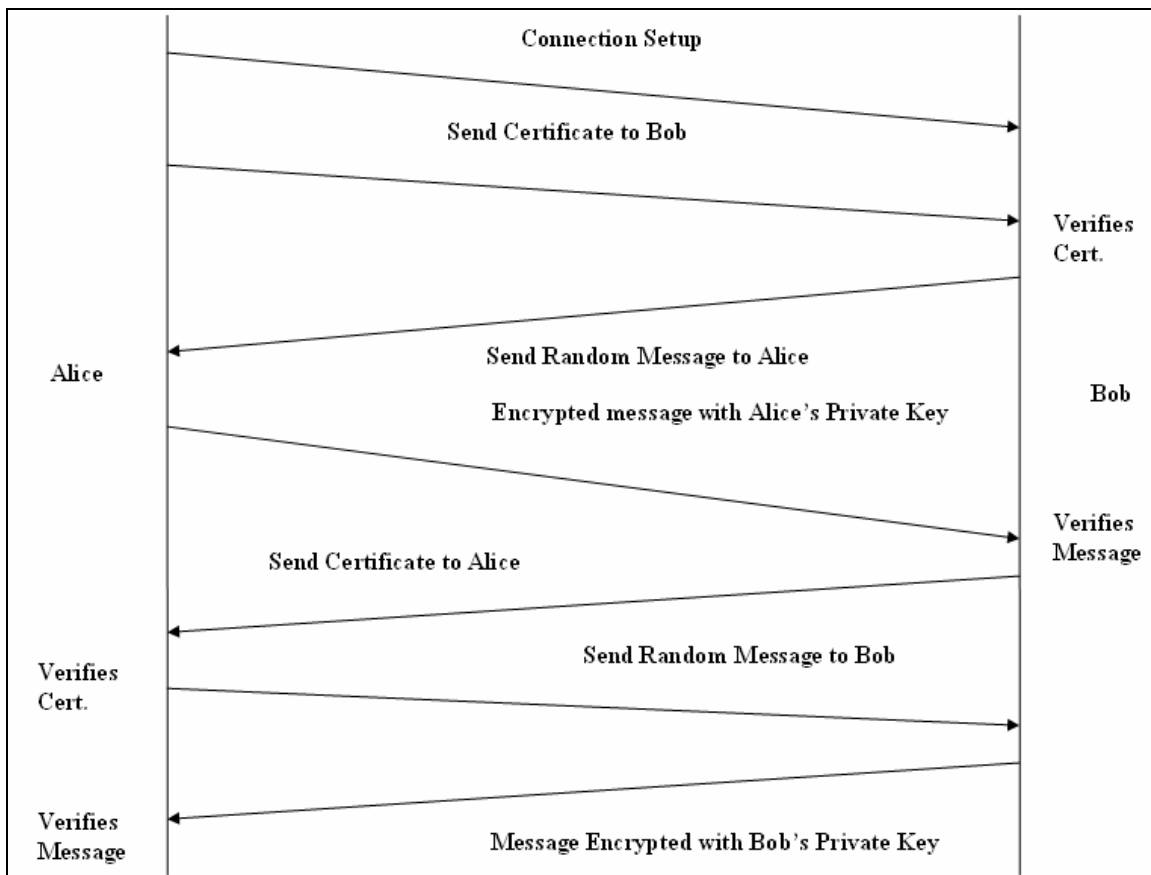
2. The second step is to create public private key pair using any common protocol. The private key thus obtained is also stored in a secure place in the local host. A different credential service like MyProxy can also be used for this purpose. In this step, the user also generates the certificate request, which is its public key signed with the authenticity of the information.
3. In third step, the CA first verifies the information obtained from the user and then signs the request with its public key. The certificate is then sent to the user.
4. The last step is to store the certificate which would be used for all subsequent authentication purposes. The public key of the CA, the user's public key and the signed certificate is stored at the local host.



**Figure 2.6:** Steps for logging into the grid

### 2.7.1.2 Mutual Authentication

Mutual authentication is an important aspect that needs to be considered where two hosts mutually authenticate each other if both of them trust the third party or the CA. Figure 2.7 shows the example of mutual authentication.



**Figure 2.7:** Example of mutual authentication

Let us assume Alice and Bob are authenticating each other.

1. First Alice sets up a connection with Bob.

2. Alice then sends her certificate over to Bob for authentication. The certificate is a standard certificate and holds the information about the identity of Alice, her public key, and the information about the CA.
3. After receiving the information from Alice, Bob first validates the received certificate to make sure that the certificate has actually been signed by CA and the authenticity of the public key. Bob then creates a random number or a message and sends it to Alice.
4. When Alice receives the random message, she encrypts it with her private key and sends the encrypted message back to Bob.
5. Bob then decrypts the message received from Alice and checks that the decrypted message is really the one that it sent before. This step is to validate that Alice actually possesses the private key corresponding to the public key she has communicated to Bob.
6. Lastly Bob trusts the identity of Alice. If Alice also like to validate Bob's identity, steps 2 – 5 are repeated with Bob sending Alice his certificate and Alice sending the random message to Bob and it is called mutual authentication.

### **2.7.2 Integration with Kerberos**

Kerberos is one of the popular authentication systems in enterprises. GSI in current form does not support Kerberos based interaction. It means Globus security does not accept Kerberos credentials as an authentication mechanism. For Globus accept this

mechanism, there is a need for gateways or translators which accept GSI credentials and convert it to Kerberos credentials.

### **2.7.3 Password Based Authentication**

Even though certificate based authentication systems are more secure, but password based systems are widely used in common. Therefore, the GSI design team allowing passwords as means of authentication in Globus based grid systems. The Globus Toolkit 4.0 (GT4) has the provision of allowing users to authenticate through username and password. The GT4 security allows Simple Object Access Protocol (SOAP) messages to be secured using Transport Layer Security (TLS) or using WS-Security standards. In unauthenticated mode authentication can be done using username and password in the SOAP message. Web services standards allow GSI in GT4, to use usernames and passwords in addition to digital certificates. However, it is to be noted that more advanced security features like confidentiality, integrity, delegation are not present in password authentication based systems.

There are a lot of inadequacies in integrating GSI with a password based system. The first is the lack of confidentiality, there is a possibility of adversaries tapping into the system. Second one is the constant change of policies and lack of trust on the host system; users do not store long-term credentials in the host. The registration process in password based authentication is well much simpler than applying for certificate based authentication (Crampton J. *et al.*, 2008). More explanation about the types of password based authentication is given in section 2.8 by listing down the existing password based authentication schemes.

## **2.8 Existing Password Based Authentication Schemes**

There are several types of existing password based authentication schemes in grid computing. The examples are one time password (OTP), user authentication with anonymity, password authentication scheme without server public key, and one-way hash function and server private key based on elliptic curve cryptosystem. Followed subsections described about these schemes in brief.

### **2.8.1 One Time Passwords (OTP)**

To have more secure system One Time Password (OTP) technologies with Globus was integrated. One Time Passwords is a step to remove some of the inadequacies of the password based systems. In this technology, the passwords change over time, like RSA, SecureID. The OTP protect compromised user's password and allows the grid systems or data centers to securely transfer a short lived credential to the user. The exchange of credentials also needed in OTP. The integration between OTP technology and secure key exchange was developed by researcher from Lawrence Berkeley National Laboratory (LBNL), and it is called OpKeyX.

In OpKeyX, first a one time password is derived, which is a function of the key and a random number which can be the current time. After that a Diffie-Hellman key exchange algorithm is used to decide on a session key. The one time password derived in the previous step is used to encrypt the key exchange mechanism. This protocol has been integrated with Transport level and Message level security of GSI. Earlier, OpKeyX is

used as the key exchange protocol in TLS but latterly it is used as the key exchange protocol in WS-SecureConversation (Chakrabarti.A. 2007).

### 2.8.2 User Authentication with Anonymity

A simple and efficient grid authentication system providing user anonymity was proposed. This scheme based on hash function, and mobile users only do symmetric encryption and decryption. This proposed technique possesses several desirable emerging properties that enable it to provide an improved level of security for grid computing systems (Ronghui Wu et. al, 2006).

User authentication with anonymity assume that the home agent HA and the foreign agent FA share a common secret key  $K_{HF}$ . In initialization phase, mobile user MN submits his/her identity  $ID_{MN}$  and chosen password  $PW_{MN}$  to the home agent HA for registration. According to the submitted values  $ID_{MN}$  and  $PW_{MN}$ , HA uses its private key  $N$  to generate  $r_1$  by computing  $r_1 = h (ID_{MN} \parallel N)$ , where  $h ( )$  is one way hash function with the output sized 512 bits. Then  $r_2 = h (r_1 \parallel h (PW_{MN}))$  was computed and followed by compute the hashing value  $h (N)$ . Anonymity technique are used smart card to delivers the information to MN which are issued in a smart card by HA through a secure channel.

When a user MN wants to roam in a new foreign network, FA needs to authenticate MN through MN's HA. User must insert the smart card into specific device and keys in the password. After that, the smart card generates the shadow SID of the user's identity by compute it. In this computing process timestamp will included which

contains the current date and time. After computation process success, a random integer selected and secret key was generated by computing the secret key.

After receiving the login request, firstly FA verifies the timestamp with current date and time. If the current date minus timestamp is larger, FA terminates the connection. Otherwise, FA uses the common secret key and timestamp to generate values to send to HA a message to verify whether the user MN is legal or not. After HA received the message, first HA verifies the timestamp, and if timestamp still valid, it will continue authentication steps.

In authentication steps, the timestamp was computed then the user's identity was obtained. Format of user identity was verified. If the format is not valid, HA terminates the connection. To check the password is equal or not some values were computed. If the password is not equal system will send FA a message for acknowledging that MN is an illegal user. But if the password is equal the HA's timestamp was generated and it send message to FA for acknowledging that MN is a legal user. Even though this technique simple, secure, and adaptive to the demand of accessing and controlling shared resources it is still required smart card and timestamp.

### **2.8.3 Password Authentication Schemes without Server Public Key**

In 2004, Chang et al. proposed a secure, efficient and practical password authentication scheme without using the server public key. Some notations used in this schemes such as U for client, S for trusted server, T for attacker, *id* for public user identity of client and more. There are two sub schemes in this authentication scheme.

First is protected password transmission scheme. In this scheme, the server stores public user identity of client ( $ID$ ), and the symmetric encryption scheme of secret and possibly weak user password with secret key of server  $E_2(K, pw)$  for each client in the database.

The steps in this scheme are start with the client chooses a random number of session-independent random exponents and computes  $E_1(pw, g^{r_1} \bmod p)$  where  $g$  is generator with order of prime numbers ( $p$ ). Then the computational result with  $id$  and the timestamp  $T$  to  $S$  as a login request.  $S$  uses  $pw$  to retrieve  $g^{r_1} \bmod p$ .  $S$  generates a random number  $r_2$  and computes the session key  $sk$ . Thereupon,  $S$  computes symmetric encryption scheme of  $g^{r_1} \bmod p$  with secret user password and send the result to  $U$ .

$U$  uses the computational result in above steps to authenticates  $S$  by decrypt the result with the session key where the session key is equal to the hash value of the data sent by  $U$  in first step. If it holds,  $U$  computes and sends the value of symmetric encryption scheme of data transmitted in second step by strong one-way hash function with session key to  $S$ . To know the access is granted or denied,  $S$  decrypts the symmetric value in third step and compares the result with the hash value of the transmitted data in second step.  $S$  will grant  $U$  the access right if it holds.

The second sub scheme in Chang et al.'s schemes is protected password change scheme. The steps of protected password change scheme are almost the same as those of the password transmission scheme, except an additional password change request in third step. In third step  $U$  send  $id$ , value of symmetric encryption scheme of data transmitted in second step by strong one-way hash function with session key and value of symmetric encryption scheme of new password and timestamp with session key to  $S$ . To granted the access process  $S$  compares the timestamp with the transmitted one in first data transmitted to authenticate  $U$  besides decrypts the value of symmetric encryption scheme of new password and timestamp with session key. However, this scheme uses the

timestamp like user authentication with anonymity and it requires serious time synchronization tasks. The protocol of this scheme is included in Appendix A1.

#### **2.8.4 Efficient Password Authentication Schemes without Server Public key**

This authentication scheme was proposed by Yoon et al. in year 2005 to avoid the timestamp problem in Chang et al.'s proposed scheme. The steps in this scheme are similarly to Chang et al.'s scheme. In Yoon et al.'s scheme protected password change scheme can simply update user passwords without a complicated process (Yoon *et al.*, 2005).

In protected password transmission scheme, the first step is U chooses a random number  $r_1$  and computes the value of symmetric encryption scheme of generators with password. Then, U sends the computation result with *id* to S as a login request. The main thing to highlight in this scheme is there are no more uses of timestamp. Secondly, after receiving a login request, S uses passwords to retrieve the generators. S generates a random number and computes the session key and sends the result to U.

U computes the session key and authenticates S by checking whether one way hash function holds to both party or not. If holds, U compute the session key and sends it with *id* to S. The last step in protected password transmission scheme is granted for access or denied. S computes the hash value using its own copies of session key and generators. Then it determines whether the hash value holds or not. If it holds, S will grant U access. After mutual authentication between U and S, generates random value numbers used as session key.

The protected password change scheme allows U to change its old password to new password. The steps in protected password change scheme are much the same as the protected password transmission scheme, except steps 1 and 4. First, U chooses a random number and new password, and computes the value of symmetric encryption scheme of generate random numbers with new password. Then U sends the computational result with *id* to S as a login request.

In between the other steps are the same as protected password transmission scheme. In final steps, S computes the hash value using its own copies of session key and random values and determines whether hash value are equal to the new session key hash value. If it is same, S will grant U access and replaces the value of symmetric encryption scheme of old password with new password. After mutual authentication between U and S, The session key is equals to the new session key, which taken from generator of random numbers. Even this scheme was proposed for avoid timestamp uses and not requires time synchronization, but it is requires a symmetric encryption algorithm and verification table maintained at server side. The protocol of this scheme is included in Appendix A2.

### **2.8.5 Authentication Scheme Based on Elliptic Curve Cryptosystem**

Rongxing et al. in 2006 proposed a new password based user authentication scheme based on the elliptic curve cryptosystem. This scheme inherits the advantages of Yoon et al.'s scheme in year 2005. Rongxing et al.'s schemes only requires a one way hash function and server private key.

Before go further about this scheme some problem and definitions must be understood first. A subgroup  $\mathbf{G}$  of the elliptic curve group  $\mathbf{E}(\mathbf{F}_p)$  with order  $q$  is constructed, where  $q$  is a large prime number. Three mathematical related problems was considered, the elliptic curve discrete logarithm problem (ECDLP), the elliptic curve computational Diffie–Hellman problem (ECCDHP) and the elliptic curve decisional Diffie-Hellman problem (ECDDHP).

The relationships between these problems are ECCDHP is no harder than ECDLP, and ECDDHP is also no harder than ECCDHP in  $\mathbf{G}$ . From *Elliptic Curve Cryptosystem* in 1987, there is no polynomial time algorithm to solve ECDDHP, ECCDHP and ECDLP with no negligible probability. A one way hash function is secure if, Hash function can take a message of arbitrary-length input and produce a message digest of a fixed length output. In Rongxing et al.'s scheme there are three phases: the registration phase, the authentication phase and the password change phase.

In the registration phase, user  $U$  submits identity to register to the server. After checking the valid of identity, the server chooses a shelf life and user secret key to compute the hash value. Then,  $S$  generates user's password and return to user. And thus user holds the password and its shelf life. Here the server doesn't need to maintain a verification table in database to store user identity and password, which therefore overcomes the stolen-verifier attack. Nevertheless the secret key of the server must be safeguard.

In authentication phase, when user wants to login into the server, some steps will followed. First, user chooses a random number and computes it and sends it to server with user public identity, random number and its shelf life. The second step is the server first checks the shelf life. If it is valid, continue to computes hash value, random number from the user and generator point from subgroup of the elliptic curve group. Server

chooses another random number and computes it. Finally server sends the server random number and hash value to user. If the shelf life is not valid the process will stop automatically.

Third step is user computes session key, this key must equal to user random number multiply server random number. User also will check the hash value, whether its hold or not. If it does hold, server is authenticated, and then user computes second value of hash function and sends it to server. Finally, server computes the second hash value which gets from user in step three. After computes the value will compared whether its same with the user's 2<sup>nd</sup> hash value or not. If they are equal, user is authenticated and granted to access the resources by server. In addition, after the mutual authentication between user and server, the same session key will be used for further operations.

After a common session key is shared between user and server as in authentication phase, they can establish a secure channel between them. Then, when user wants to change his/her password in password change phase, he/she can securely request a new password. First, user sends identity, old password and the shelf life to server using secure channel. Then server checks whether password equals to hash value multiply secret key or identity or shelf life power of secret key. If it does hold, server chooses a new shelf life and new password. Then it will sends back to user using the secure channel. The protocol of this scheme is included in Appendix A3.

## 2.9 The Summary of Password Based Authentication Scheme

There are five types of existing schemes. Table 2.4 shows the existing password based authentication schemes that involves in this project. These schemes are compared to see the approaches, the strengths and the weaknesses in each of them.

**Table 2.4** Existing password based authentication scheme

Authentication scheme	Approaches	Strengths	Weaknesses
OTP	Password change over time	Remove inadequacies of the password based systems	Only used in Globus Security Infrastructure
User Authentication with Anonymity	Assume home agent and foreign agent share common secret key; Has own access control system which consists of authentication server and terminal proxy system	Low cost functions; Easily implemented; Suitable for wireless environment	Required smart card; Used timestamp
Chang's Scheme	Not using server public key; Server stores ID, secret keys, user password in database;	User can change password;	Required 5 times symmetric encryption & 5 times symmetric decryption; Required timestamp

	Consists 2 schemes, password transmission scheme and password change scheme;		
Yoon's Scheme	No server public key; Server stored id, secret key, password using server's secret key for each user in database;	Overcome server data eavesdropping; Allow user to change its old password to new password; Resist 7 security properties; Required 1 times symmetric encryption and symmetric decryption	Required a symmetric encryption algorithm; Require verification table maintained at server side
Rongxing's scheme	Based on elliptic curve cryptosystem; Required one-way hash function; Consists three phases, registration phase, password change phase, authentication phase; Require server private key;	No verification table at server side; Resist 5 security properties; Reduce total overhead for communication and performance; Simple and easily implemented	Must ensure the system secret key is secure enough.

Five types of existing authentication scheme were compared. Table 2.4 shows the approaches, the strengths and the weaknesses of the existing authentication schemes. All these can be summarize as below. In OTP scheme, the password change over time and it able to remove inadequacies of the password based systems. But the weakness of OTP is it can be implemented in Globus Toolkit only. Authentication with Anonymity assumes home agent and foreign agent share a common secret key. This authentication scheme has its own access control system which consists of authentication server and also terminal proxy system.

This scheme used low cost functions and also suit for wireless environment. Anonymity scheme required smart card and timestamp. The scheme that proposed by Chang et al.'s not using server public. This scheme required 5 time's symmetric encryption and also 5 times symmetric decryption. Timestamp was required in this scheme and tend to serious time synchronization task.

Yoon et al.'s scheme also not using server public key. This scheme is enhancement for Chang et al.'s scheme. This scheme overcomes server data eavesdropping. This authentication scheme allow user to change user's old password to new password and also resist seven security properties. This scheme required one time's symmetric encryption and symmetric decryption. Anyhow, this scheme required symmetric encryption algorithm and required verification table maintained at server side.

Rongxing et al.'s scheme is based on Elliptic Curve Cryptosystem. It consists three phases; registration phase, password change phase and authentication phase. This scheme no needs to provide verification table at server side and resist five security properties. Rongxing et al.'s scheme reduced total overhead for communication and performance and also easy to implement. The problem in this scheme is the system secret

key must be ensured that it is secure enough. For complete this project The Rongxing's *et al.* scheme is selected as reference scheme for pre-lab testing in Chapter 4 and also lab testing in Chapter 5.

## **2.10 Conclusion**

This chapter covers the literature review according to previous works done by several researchers regarding on grid security infrastructure, password based authentication schemes. This literature review is very important as it gives overall views and details explanation on particular subjects. This will indeed gives better understanding on what the study is all about and what are the things that must be included in the proposed scheme. Some topics that had already covered up in this chapter including overview of grid computing, security in grid computing, grid security infrastructure, the existing grid security infrastructure, authentication in general, authentication in grid security infrastructure, existing password based authentication schemes. Finally, some literatures are done on simulator that is suitable to simulate grid environment that can implement grid security infrastructure and password based authentication scheme.

## **CHAPTER 3**

### **METHODOLOGY**

#### **3.1 Introduction**

The overall approaches and perspectives to the research process and concerned with issues like why collecting certain data, what data is collected and where it is collected, how the data is collected and analyzed are refers to methodology (Collis and Hussey, 2003). This chapter will explained about the methodology of the project including research approach, research strategy and the operational framework. The project methodology will covered the research approach of the project and find out which approaches are suitable to be implemented. From the research approach, the research strategy which consists of comparative study and lab testing will be explained. Then, flows of operations involved in the project are shown through operational framework. Lastly, the software and hardware requirement needed for this project are explained.

## **3.2 Research Approach**

Process of enquiry and investigation are called as research. The investigation process must be systematic, methodical and ethical, which can help in solving practical problems and increase knowledge. The purpose of research is to investigate existing problems, provide the solutions, construct or create new procedure or system and also to generate new knowledge. There are several kinds of research approaches, but the quantitative method and the qualitative method approaches are commonly used.

### **3.2.1 Quantitative Method**

Quantitative method is a type of scientific research that can be defined as the collection of numerical and statistical data. It concentrates on measuring phenomena in terms of the scale, range, frequency etc. Quantitative data can be described as data which can be stored, classified, measured in a strictly objective way. It is also involves few variables and many cases, and employs prescribed procedures to ensure validity and reliability. Generally, quantitative methods are designed to provide summaries of data that support generalizations about the phenomenon under study. It is harder to design initially, but quantitative method is usually highly detailed and structured where the results can be easily presented statistically. The benefit of quantitative research lies in the researcher's ability to summarize results in statistically meaningful ways, allowing findings to be generalized to other populations. This method mostly used in final stage of this project.

### 3.2.2 Qualitative Method

Another method in research approach is qualitative method. This method is more subjective in nature than quantitative method. Qualitative method involves examining and reflecting on the less tangible aspects of a research subject like values, attitudes and perceptions. Qualitative methods also a type of scientific research that mainly concerned with the properties, the state and the characteristic of the research subject. Although data collection standards exist, qualitative research is highly reliant upon the researcher carrying out the study. The researcher has total control over the type of data collected and the methods used for analysis.

Qualitative method usually achieves a greater level of depth and detail, however fewer subjects tend to be studied resulting in a study being more difficult to generalize. This method is preferred when researching sensitive subjects. Rather than being constrained by pre-set answers, they allow sensitive subjects to be approached in a sensitive way by allowing the researcher to employ personal skills to help lessen the difficulties of the subject matter. Although this kind of research can be easier to start, it can be often difficult to interpret and present the findings. The findings can also be challenged more easily. For methodology of this project, the qualitative method which is typically more flexible than quantitative method is used. The study is carried out by selecting existing authentication schemes.

### **3.3 Research Study**

After selecting the appropriate method for the methodology, which is using the qualitative research method, it is essential to think and create a research strategy. This research strategy will help researchers to conduct their studies or experiments in a proper way to get a better result within the scope of their research project. Generally, research strategy makes it easier to discuss and handle dilemmas that might occur among gathered information in research. There are two strategies used in conducting this project methodology that are comparative study and lab testing.

#### **3.3.1 Comparative Study**

Comparative study is one of the exploratory studies where scientists try to move from the initial level of case studies to a more advanced level of general theoretical invariance such as causality or evolution. Comparison is one of the methods that can be used for explicating or utilizing tacit knowledge and also tacit attitudes. In comparative study there is no earlier model or theories to start with and it helps to easy design. The design involves cases which are similar in some respects but they differ in some respects. These differences become highlight of examination. The goal of comparative study is to find out why the cases are different in order to reveal the general underlying structure which allows and generates such variation.

In comparative study, two or more cases are observed. As the number of cases is large, the study begins to approach classification. From the bundle of gathered information, there is a need to decide what are the interesting aspects, properties or

attributes that need to be recorded for each of the cases. This information can be simplified in a matrix table as shown in Table 3.1, so the differences among the cases can be clearly defined. During analysis, new aspects can be added and the unwanted aspects can be taken out. Table 3.1 shows the example of comparative study table.

**Table 3.1:** The example of comparative study table

<b>Scheme</b>	<b>Observed state of things</b>	
	<b>Features 1</b>	<b>Features 2</b>
<b>Scheme A</b>	A1	A2
<b>Scheme B</b>	B1	B2
<b>Scheme C</b>	C1	C2

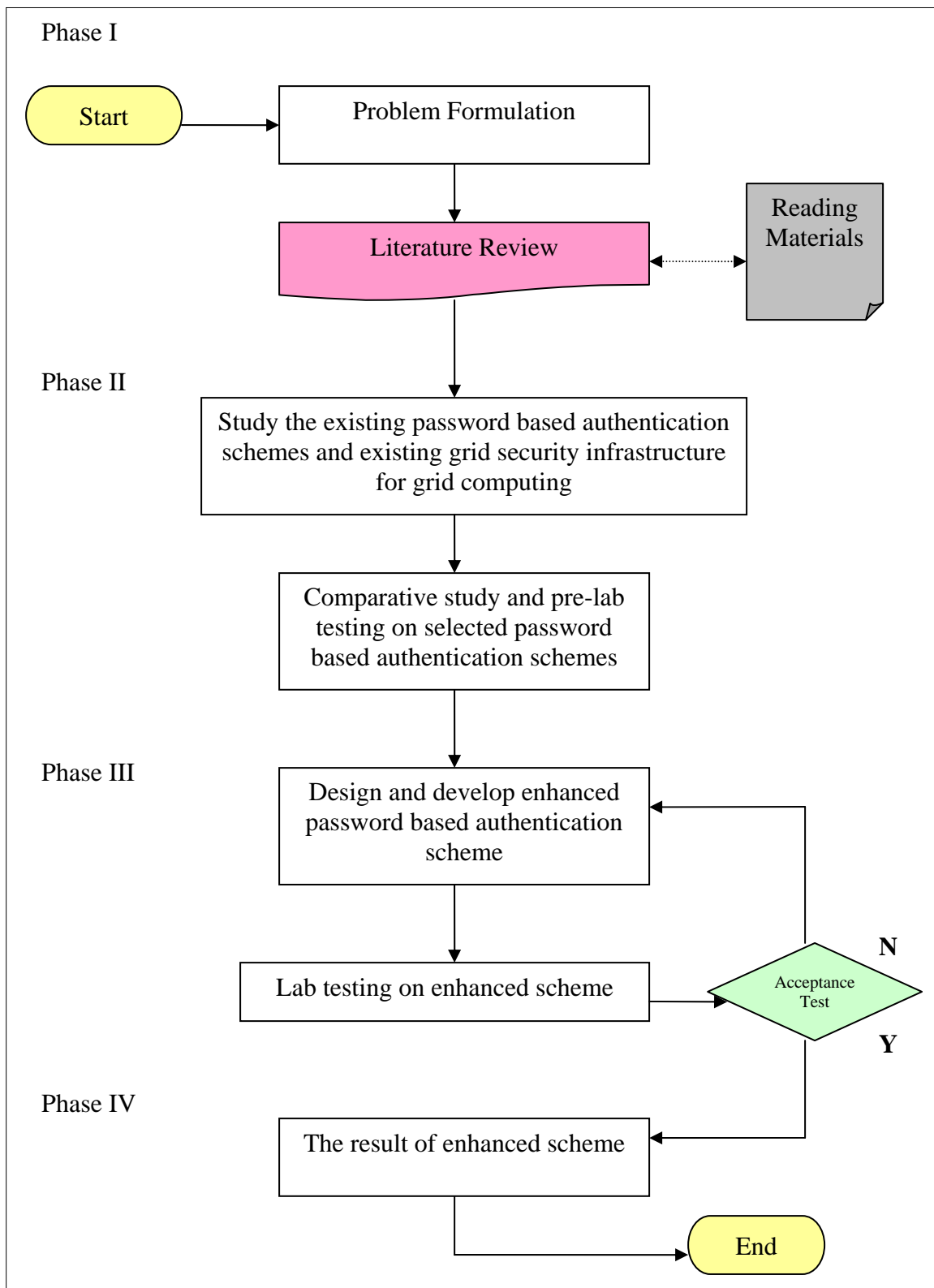
The final goal of research is to reveal the systematic structure and invariance, that is true not only for the cases that were studied, but for the entire group where the cases came from or for all the comparable cases. As conclusion, the goal is to generalize the findings.

### 3.3.2 Lab Experiment

Experimental studies are done carefully controlled and structured environments and enable the causal relationships of phenomena to be identified and analyzed. The variables can be manipulated or controlled to observe the effects on the subjects studied. Studies done in laboratories tend to offer the best opportunities for controlling the variables in a rigorous way, although field of studies can be done in a more real world environment. However, with the former, artificiality of the situation can affect the response of the people studied, and with the latter, the researcher has less control over the variables affecting the situation under observation.

### **3.4 Operational Framework**

This section is about the flows of operations that involves in this project, which act as a guidelines for archive the project's aim and objectives. Figure 3.1 shows the flow chart of the operational framework. Based on figure 3.1 the operational framework can be divided into 4 phases. There are literature review phase, comparative study and pre-lab testing phase, design, develop and testing phase and also result phase.



**Figure 3.1:** Flow chart of operational framework

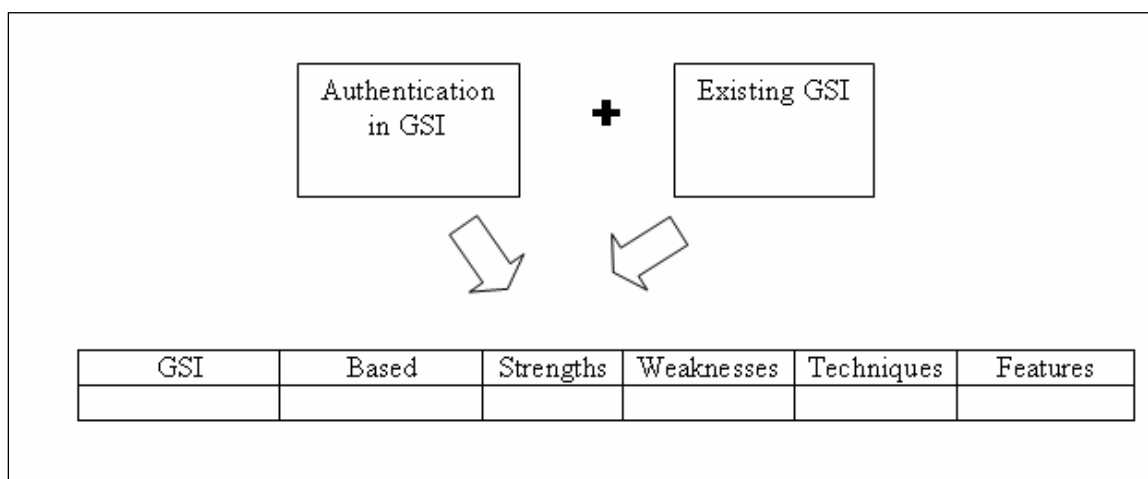
### **3.4.1 Phase I : Literature Review**

In literature review phase, the problem of the security of grid computing which can referred as the security issues in grid computing is first looked into. In order to do so, there is a need to understand the concept of grid computing by the overview of grid computing. During the problem formulation activity, it is figured out that authentication is an essential part in securing grid computing and thus will be the focus of the project. The aim, objectives and the scope of project then listed out. After that, the literature review of the project was written based on the reading materials. The reading materials are mostly taken from general and specific papers such as journals, proceedings, books and others, which mainly about grid computing, security in grid computing and authentication schemes.

Some literature reviews are also done on the selected authentication schemes and selected security infrastructure which based on password based authentication schemes and grid security infrastructure. These include the study of existing password based authentication schemes and also existing grid security infrastructures. Study is also done on a few codes that are capable to run the Grid environment with the designated activities in the enhanced password based authentication scheme.

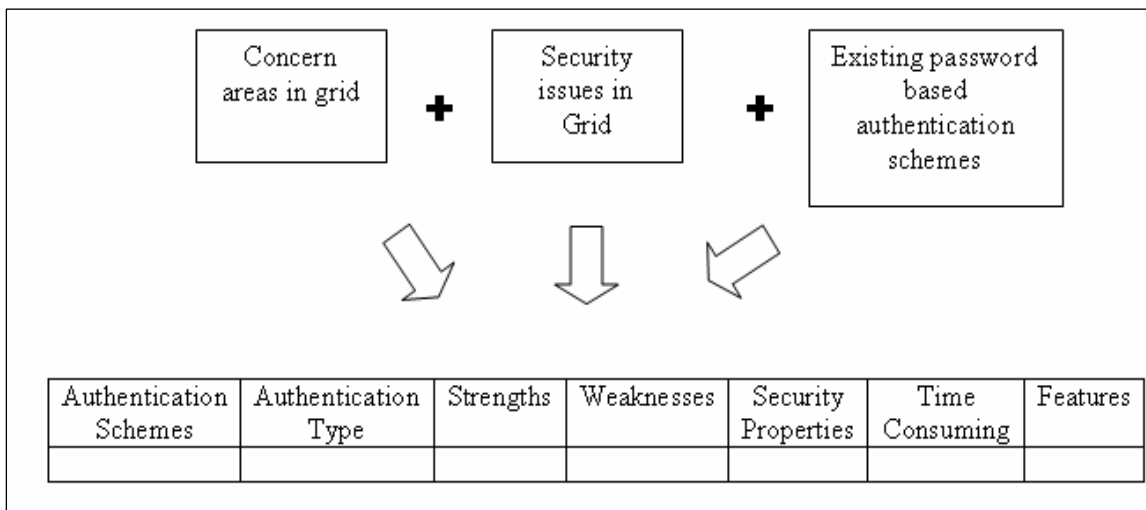
### 3.4.2 Phase II: Comparative Study and Pre-lab testing

Some critical analysis on the existing password based authentication schemes was done before enhanced a password based authentication scheme. In order to enhance a password based authentication scheme, the analysis of the existing grid security infrastructure also was done. This is done in comparative study, where by important information is first need to be derived from the gathered reading materials. The information are about the security issues in grid computing, the existing grid security infrastructure, and also the existing password based authentication schemes. Figure 3.2 show the comparative studies on the grid security infrastructure.



**Figure 3.2** Comparative studies on the grid security infrastructure

The comparative study on existing password based authentication also has been done. Figure 3.3 shows the comparative study on the password based authentication schemes



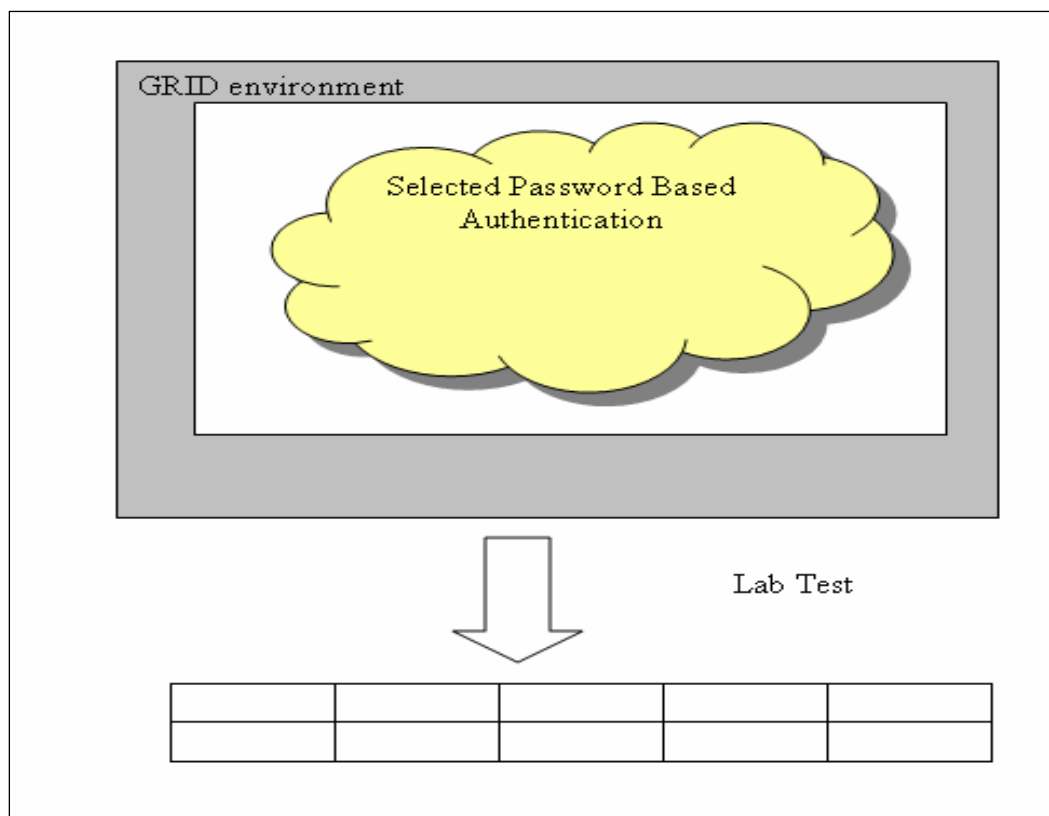
**Figure 3.3:** Comparative study on the password based authentication schemes.

From Figure 3.3, it can be seen that from all the gathered information, the existing authentication schemes need to compare in terms of authentication type, the strengths, the weaknesses, security properties, the time consuming and also the features of each of them. From figure 3.3 also can be seen about the existing grid security infrastructures. These infrastructures were compared in terms of what based security infrastructure, strengths, weaknesses, techniques used and also the features of each of them.

All these findings are combined in particular matrix tables so that clear difference among the password based authentication schemes and also the grid security infrastructures can be seen. These comparative study tables can be found in the literature review of chapter 2. The clear view can be found in comparative study and pre-lab testing which covered in Chapter 4 by carrying out the comparative study on the existing authentication scheme and also the existing grid security infrastructures. Password Enable Certificate Free – Grid Security Infrastructure (PECF-GSI) is selected as the grid security infrastructure for the

selected and enhanced password based authentication scheme based on its flexibility and well-suited for password based authentication schemes.

After performing the comparative study, the suitable grid security infrastructure was chosen and the pre-lab testing is done on the existing password based authentication schemes. Figure 3.4 shows the pre-lab testing on the existing password based authentication schemes.



**Figure 3.4:** Pre-lab testing on the existing password based authentication scheme

How the pre-lab testing will be done, can be shown in Figure 3.4. Real lab grid environment was created and selected password based authentication scheme was run. In order to run the existing password based authentication schemes, the suitable grid security infrastructure must be developed first.

### **3.4.3 Phase III: Design, Develop and Testing**

The result from phase II, will be used for the next step. The next step is to design and develop the enhanced password based authentication scheme. After the enhanced scheme is designed and successfully developed, the lab testing will be done on the enhanced scheme in a real lab environment. If the enhanced scheme works well during test acceptance, the result of that test will be gained. If it is not, the design and development of the enhanced scheme need to be done again. Chapter 5 explained in details about this phase.

For developing and lab testing on the password based authentication scheme in a grid environment, the real lab test approach is used rather than a simulation approach. The reasons are still now there are no any simulators to simulate password based authentication schemes for grid computing. C++ language is chosen as a programming language to develop the enhanced password based authentication scheme. C++ is easy to implement and it takes little memory to compile the program.

### **3.4.4 Phase IV: Result**

In this final phase, the results of the enhanced scheme are taken into consideration. Some analysis and discussion on the result will be done to measure and compare the selected password based authentication scheme and the enhanced password based authentication scheme. This is covered up in Chapter 5

### **3.5 Hardware and Software Requirements**

The following are the minimum software requirements for this project:

1. Borland C++ 5.02 Compiler for Windows XP professional
2. GNU Compiler for Fedora Core
3. Microsoft Office Word 2003.

The following are the minimum hardware requirement for this project:

1. 3 computers with Fedora Core operating system
2. Mouse and other necessary input/output devices.

### **3.6 Conclusion**

This chapter discussed the activities involved in designing the enhanced password based authentication scheme where the comparative study and pre-lab testing involved in the research strategies. The operational framework of this project also was included. This chapter gives some guidelines in designing the enhanced password based authentication scheme. Finally the minimum requirement of hardware and software is included.

## **CHAPTER 4**

### **COMPARATIVE STUDY AND PRE-LAB TESTING**

#### **4.1 Introduction**

This chapter covers the findings of the project. In this project some findings have been discovered while conducting research strategy which is by doing comparative study and lab experiments. The findings are much more related to the grid security infrastructures and the selected existing password based authentication schemes which suitable for grid environment.

The selected password based authentication schemes that are used in this study is the Rongxing et al.'s scheme and the selected grid security infrastructure is the Password Enable Certificate Free Grid Security Infrastructure (PECF-GSI). The lab experiment findings that involve the use of real code also covered in this chapter. The findings of both research strategies are very important and need to be accomplished as they became the basis on developing the enhanced authentication scheme.

## 4.2 The Findings for Comparative Studies

Several comparative studies has been conducted in this chapter include the features of password based authentication schemes and grid security infrastructure, The findings of the features of password based authentication schemes and the grid security infrastructure lead to selecting for the enhanced authentication scheme.

### 4.2.1 Features of Existing Grid Security Infrastructure

Initially, there are several numbers of existing grid security infrastructures found during comparative study phase and among them, three infrastructures have been chosen to be studied. There are Globus Toolkit's Security Infrastructure (GSI), Password Based Grid Security Infrastructure (PBGSI) and lastly Password Enabled Certificate Free Grid Security Infrastructure (PECF-GSI). These three infrastructures then studied in details and compared among each other.

During the analysis, there are a few aspects that have been looked into where some infrastructures are similar in certain aspects but differ in some other aspects. These aspects are the features of existing infrastructures. Table 4.1 shows the variety features of grid security infrastructures. This finding gives some idea to choose which grid security infrastructure is suitable for the enhanced scheme. “√” symbol shows that the infrastructure has a particular feature while “X” symbol shows that the scheme does not have that particular features.

**Table 4.1:** The variety features of grid security infrastructures

Grid Security Infrastructure	Features						
	Authentication Server	Timestamp	CA	TA	Key Exchange	Secure Data Transfer	Proxy
GSI		X	√			√	√
PBGSI	X	√			√	√	
PECF-GSI	√	X	X	√		√	√

The features of the existing grid security infrastructure are listed as below; authentication server, timestamp, Certificate Authority, Trust Authority, key exchange, secure data transfer and proxy. From table 4.1, the analysis on existing grid security infrastructures can be done. The analysis can be described as below. The GSI not required timestamp and it is means that no time synchronization task occurs. Anyhow GSI used CA and has proxy credentials.

PBGSI not required authentication server but it is used timestamp. PBGSI also has key exchange protocols. The secure data transfer channel can be found in PBGSI. PECF-GSI has authentication server and not required for timestamp. It is also not required CA for authentication but it is required TA. From the analysis, the suitable grid security infrastructure is the PECF-GSI because it support password based authentication without CA and not required timestamp.

#### 4.2.2 Features of Existing Password Based Authentication Schemes

There are five numbers of existing password based authentication schemes. There are One Time Password (OTP), Authentication with Anonymity, Chang et al.'s scheme, Yoon et al.'s scheme and Rongxing et al.'s scheme. Same as grid security infrastructures, these existing schemes also were studied in details and compared in terms of many features among each others.

Exactly same as grid security infrastructures, during the analysis process certain features are same and some are differ among these schemes. Table 4.2 shows the features of password based authentication schemes.

**Table 4.2:** Features of password based authentication schemes

Scheme	Features										
	Timestamp	Hash Function	Smart Card	Server private key	Symmetric Encryption	Verification table	Server public key	Modular exponential	Elliptic curve crypto system	CA	Key Exchange
OTP										√	√
Anonymity	√	√	√	√							
Chang's	√	√	X	√			X	√			
Yoon's	X	√	X	√	√	√	X	√			
Rongxing's	X	√	X	√		X	X	√	√		

The features of the existing authentication schemes are list as below; timestamp, hash function, smart card, server private key, symmetric encryption, verification table, server public key, modular exponential, elliptic curve cryptosystem, Certificate Authority, and key exchange. Table 4.2 give a clear view of comparison on features of existing password based authentication scheme.

OTP just required CA and has key exchange protocols. The Anonymity scheme required timestamp, hash function, smart card and also server private key. Chang's scheme also required timestamp but it is not required for any hardware because it is not using token based authentication. The modular exponential is one of the computational types and it is required here.

Yoon's scheme is similar to Chang's scheme but it is not required timestamp. Symmetric Encryption algorithm and verification table are required by this scheme. The other features are similar to the Chang's scheme. Lastly Rongxing's scheme required hash functions with server private key and used elliptic curve cryptosystem technique.

### **4.2.3 The Features of Security Properties**

Besides the features of existing grid security infrastructure and features of existing password based authentication scheme, the analysis of security properties from Yoon et al.'s scheme and Rongxing's et al.'s scheme was done before choose a proposed scheme. Table 4.3 shows the features of security properties from Yoon et al.'s scheme and Rongxing et al.'s scheme.

**Table 4.3:** Security properties for Yoon's and Rongxing's scheme

Scheme	Security properties							
	Replay	Server data eavesdropping	Server spoofing	DoS	Mutual Authentication	Perfect forward secrecy	On-line password guessing	Off-line password guessing
Yoon's	√	√	√	√	√	√	√	
Rongxing's	√		√			√	√	√

The security properties of these two existing scheme are consists of replay attacks, server data eavesdropping, server spoofing attacks, DoS attack mutual authentication, Perfect forward secrecy, on-line password guessing and off-line password guessing. From Table 4.3 the clear view of the security properties of Yoon's scheme and Rongxing's scheme was displayed and the analysis can be done.

Yoon et al.'s scheme can resists Replay Attacks, Server Data Eavesdropping, Server Spoofing, Denial of service (DoS) attacks, Mutual Authentication, Perfect Forward Secrecy and On-line Password Guessing Attacks. Yoon et al.'s considered seven security properties to provide the proof of correctness.

Five security properties were examined of Rongxing et al.'s scheme. There are; Replay Attacks, On-line password guessing attack, Off-line password guessing attack, Server spoofing attack and Perfect forward secrecy.

### 4.3 The Lab Experiment Findings

The lab experiment findings done in this section is very important for the enhancement of selected password based authentication scheme. The findings of time complexity notations is an important component used to validate the selected password based authentication and the enhanced password based authentication scheme. The time complexity notations will calculate according the formula of time complexity which is given by Rongxing's *et. al.*

#### 4.3.1 Time Complexity Notations

Time complexity is the time of a program is the number of elementary instructions that this program executes. This number is computed with respect to the size  $n$  of the input data. Each elementary instruction takes the same amount of time because it is probably one processor instruction, which is true for example for addition and multiplication between integers. The requirement for time complexity is the execution time and memory space. The quantities estimation described that execution time is equals to time complexity (Nicolas Stroppa, 2006). The time complexity notations is defined as aggregated information that satisfies a particular information need when it comes to monitoring process performance. There are several time complexity notations for the password based authentication scheme. Some of the time complexity notations and its definitions for the password based authentication are as shown in Table 4.4.

**Table 4.4** Time complexity notations and its definitions

Notation	Definitions
$T_{Mul}$	The time for the modular multiplication
$T_{Exp}$	The time for the modular exponentiation
$T_{Pmul}$	The time for the multiplication of a number and an elliptic curve point
$T_{Inv}$	The time for the modular inversion
$T_{Ha}$	The time for the hashing operation
$T_{En}$	The time for the symmetric encryption operation
$T_{De}$	The time for the symmetric decryption operation

There are seven time complexity notations shown in Table 4.4. There is time for the modular multiplication, time for the modular exponentiation, time for the multiplication of a number and an elliptic curve point, time for the modular inversion, time for the hashing operation, time for the symmetric encryption operation and also time for the symmetric decryption operation. Even there are seven time complexity notations listed in Table 4.4, four of these notations are used for measure the performance of selected and enhanced password based authentication scheme. There are time for the hashing operation, time for the multiplication of a number and an elliptic curve point, time for the modular inversion and the time for modular multiplication. The other time complexity notations are not involved in this project because the selected scheme and enhanced scheme does not require exponentiation, symmetric encryption and symmetric decryption. Under the conditions assumed in *The state of elliptic curve cryptography* by N. Koblitz *et. al.* (2000), the time complexity associated with the different operations can be roughly combined into multiplication operations. For example:

$$T_{Exp} \approx 240 T_{Mul}$$

$$T_{Pmul} \approx 29 T_{Mul}$$

The above expression defined that one time for the modular exponentiation is equal to 240 of the time for the modular multiplication. Same as that, 1 time for the multiplication of a number and an elliptic curve point is equal to 29 of the time for the modular multiplication. These time complexity notations is important to measure the performance of selected and enhanced password based authentication scheme. There is a formula for measure the time complexity of the password based authentication scheme. The formula for user and server side is different. The estimation of performance aimed at time complexity is shown in Table 4.5.

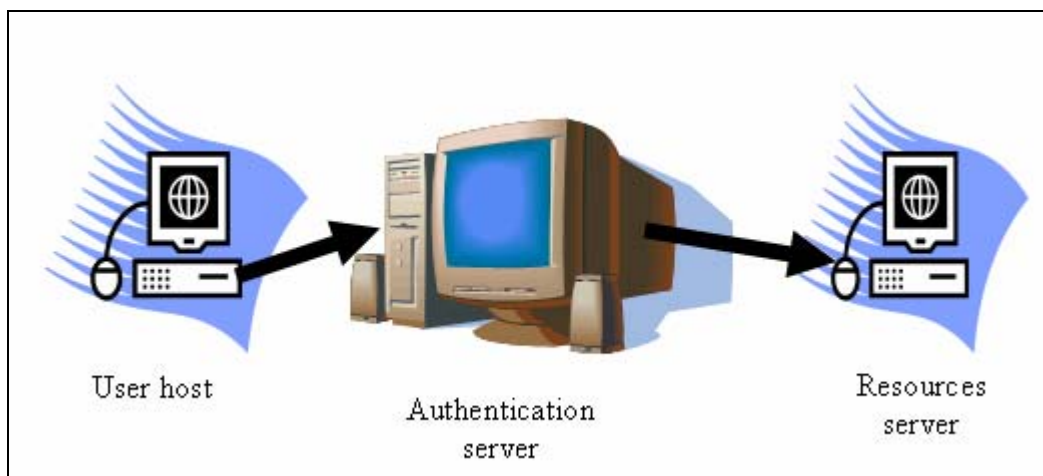
**Table 4.5** Estimation of performance aimed at time complexity

Items	User	Server
Time Complexity	$2 T_{Pmul} + 2 T_{Ha}$	$3 T_{Pmul} + 3 T_{Ha} + T_{Inv}$
Rough Estimation	$58 T_{Mul} + 2 T_{Ha}$	$87 T_{Mul} + 3 T_{Ha} + T_{Inv}$

Table 4.5 shows the formula for time complexity and the estimation of performance aimed which is different for the user site and server site. The time complexity in user site will calculate by multiplying 2 with the time for the multiplication of a number and an elliptic curve point ( $T_{Pmul}$ ), and multiplying 2 with the time for hashing operation ( $T_{Ha}$ ) and finally add both of the answers. For the server site the time for the multiplication of a number and an elliptic curve point ( $T_{Pmul}$ ) will multiply by 3, the time for hashing operation ( $T_{Ha}$ ) also multiply with 3, finally both answers need to add with the time for the modular inversion ( $T_{Inv}$ ). The time complexity formula then changed to the rough estimation formula which considered by the conditions in *The state of elliptic curve cryptography* by N. Koblitz *et. al.*(2000). These notations are tested in the pre-lab testing, which is done on the Rongxing's *et. al.* scheme as a selected password based authentication scheme and its discussed in the next section of this chapter and also in the lab testing for the enhanced password based authentication scheme that is discussed in Chapter 5.

### 4.3.2 Lab Setup

For the lab testing, three hosts were used as a user, resources server and authentication server. The public identity of user (ID) of the user host and resources host was set earlier. Even though, the resources host is called as resources server, it's still a user host for the authentication server. The ID of the user and resources host was set in alphanumerical format. The compiler will convert the alphabet and the numbers to ASCII code when its want to generate a key using the ID. Figure 4.1 shows the hosts setup in the pre-lab and lab testing.



**Figure 4.1** Host setup for pre-lab and lab testing

Referred to Figure 4.1, there are 3 hosts are used in this lab testing. There are user host, authentication server and resources host. For the real lab there can be many user hosts and resources hosts, but for this lab testing just one user host and one resources server was used. The PEFCF-GSI was developed in authentication server. The explanation of the development on PEFCF-GSI is discussed in Chapter 5. The password based authentication scheme was developed in between user host and authentication server in

PECF-GSI. The clear view of the PECF-GSI and the location of the password based authentication scheme were already discussed in Chapter 2.

### 4.3.3 The Findings of the Pre-lab Testing

A real grid environment was build with three hosts which is discussed in previous section. The researcher of the Rongxing's scheme did not specify any accurate result for the time complexity. The password based authentication scheme was run about 20 iteration and the full details of the result as shown in Appendix A4. The results after 10<sup>th</sup> iterations are remaining same. Table 4.6 shows the results of the first 10 iterations at user site.

**Table 4.6** Result of the first 10 iterations at user site

<b>Iterations</b>	<b>T<sub>Pmul</sub></b>	<b>T<sub>Mul</sub></b>	<b>T<sub>Ha</sub></b>
<b>1</b>	0.00023	0.00031	0.00025
<b>2</b>	0.00023	0.00031	0.00025
<b>3</b>	0.00023	0.00031	0.00025
<b>4</b>	0.00021	0.00030	0.00023
<b>5</b>	0.00022	0.00029	0.00023
<b>6</b>	0.00021	0.00029	0.00023
<b>7</b>	0.00021	0.00027	0.00023
<b>8</b>	0.00022	0.00027	0.00021
<b>9</b>	0.00022	0.00027	0.00021
<b>10</b>	0.00020	0.00025	0.00021

Referred to Table 4.6, the time for the multiplication of a number and an elliptic curve point is between 0.00020 milliseconds to 0.00023 milliseconds. At 5<sup>th</sup> iterations the  $T_{P_{mul}}$  is increase 0.0001 milliseconds compare to 4<sup>th</sup> iterations and decrease again 0.0001 milliseconds at 6<sup>th</sup> iterations. The time for the modular multiplication is between 0.00025 milliseconds to 0.00031 milliseconds. The time for hash operation for the first 10 iterations is between 0.00021 milliseconds to 0.00025 milliseconds. The time for modular multiplication and the time for hashing operations are in descending order from 1<sup>st</sup> iteration till the 10<sup>th</sup> iterations. Table 4.7 shows the results of the first 10 iterations at server site.

**Table 4.7** Result of the first 10 iterations at server site

<b>Iterations</b>	<b><math>T_{P_{mul}}</math></b>	<b><math>T_{Mul}</math></b>	<b><math>T_{Ha}</math></b>	<b><math>T_{Inv}</math></b>
<b>1</b>	0.00025	0.00036	0.00027	0.00018
<b>2</b>	0.00025	0.00036	0.00026	0.00018
<b>3</b>	0.00025	0.00034	0.00026	0.00018
<b>4</b>	0.00023	0.00033	0.00026	0.00018
<b>5</b>	0.00023	0.00033	0.00026	0.00016
<b>6</b>	0.00022	0.00033	0.00025	0.00016
<b>7</b>	0.00022	0.00031	0.00025	0.00016
<b>8</b>	0.00020	0.00031	0.00024	0.00015
<b>9</b>	0.00020	0.00028	0.00024	0.00015
<b>10</b>	0.00020	0.00028	0.00023	0.00015

Referred to Table 4.7, the results of the time complexity notations such as time for modular multiplication, time for multiplication of a number and an elliptic curve point, the time for hashing operation and the time for modular inversion are in descending order from 1<sup>st</sup> iteration to the 10<sup>th</sup> iterations at server site. The range of the time for multiplication of a number and an elliptic curve point is 0.00020 milliseconds to 0.00025 milliseconds. The time for modular multiplication is between 0.00028 milliseconds to

0.00036 milliseconds. The time for hash operations is near to the time for hash function at user site, but at server site the range is between 0.00023 milliseconds to 0.00027 milliseconds. The time for modular inversion is between 0.00015 milliseconds to 0.00018 milliseconds.

From the Table 4.6 and Table 4.7, the time complexity notations give better results at lower iterations when compared to upper iterations. The time complexity and the rough estimation of performance of the selected password based authentication are calculated by using the formula in Table 4.5. The time complexity and rough estimation at user site as shown in Table 4.8.

**Table 4.8** Time complexity and rough estimation at user site

<b>Iterations</b>	<b>T<sub>Pmul</sub></b>	<b>T<sub>Mul</sub></b>	<b>T<sub>Ha</sub></b>	<b>Time complexity (milliseconds)</b>	<b>Rough estimation (milliseconds)</b>
<b>1</b>	0.00023	0.00031	0.00025	0.00096	0.01848
<b>2</b>	0.00023	0.00031	0.00025	0.00096	0.01848
<b>3</b>	0.00023	0.00031	0.00025	0.00096	0.01848
<b>4</b>	0.00021	0.00030	0.00023	0.00088	0.01786
<b>5</b>	0.00022	0.00029	0.00023	0.00090	0.01728
<b>6</b>	0.00021	0.00029	0.00023	0.00088	0.01728
<b>7</b>	0.00021	0.00027	0.00023	0.00088	0.01612
<b>8</b>	0.00022	0.00027	0.00021	0.00086	0.01608
<b>9</b>	0.00022	0.00027	0.00021	0.00086	0.01608
<b>10</b>	0.00020	0.00025	0.00021	0.00082	0.01492

Table 4.8 shows the time complexity and rough estimation at user site for the first 10 iteration. The time complexity at user site is between 0.00082 milliseconds to 0.00096 milliseconds. This time complexity results also in descending order except at 5<sup>th</sup> iteration.

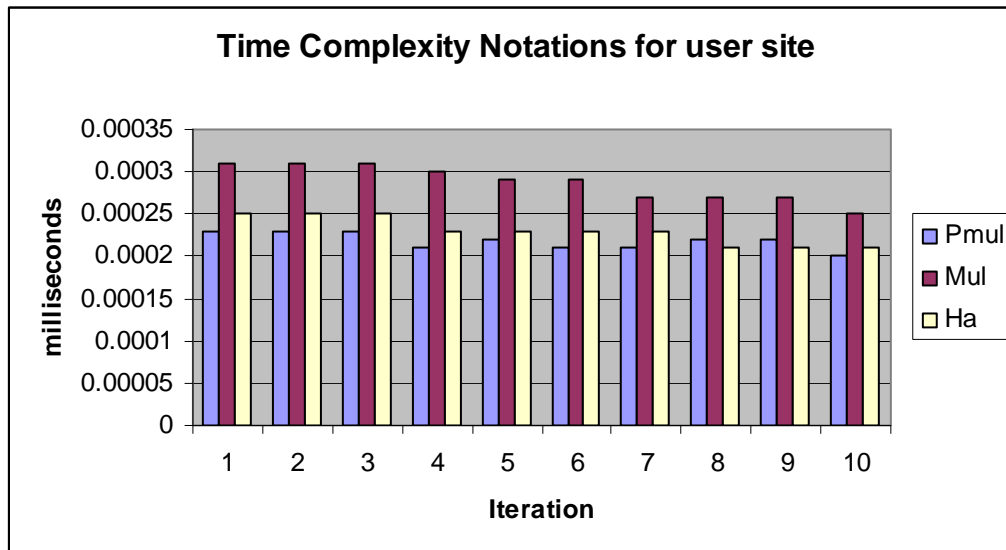
It is happened because the time for multiplication of a number and an elliptic curve point in 5<sup>th</sup> iterations is increased compared to 4<sup>th</sup> iterations. The rough estimation results for the first 10 iterations are between 0.01492 milliseconds to 0.01848 milliseconds. The results of time complexity and rough estimation for the first 10 iterations at server site as shown in Table 4.9.

**Table 4.9** Time complexity and rough estimation at server site

<b>Iterations</b>	<b>T<sub>Pmul</sub></b>	<b>T<sub>Mul</sub></b>	<b>T<sub>Ha</sub></b>	<b>T<sub>Inv</sub></b>	<b>Time complexity (milliseconds)</b>	<b>Rough estimation (milliseconds)</b>
<b>1</b>	0.00025	0.00036	0.00027	0.00018	0.00174	0.03231
<b>2</b>	0.00025	0.00036	0.00026	0.00018	0.00171	0.03228
<b>3</b>	0.00025	0.00034	0.00026	0.00018	0.00171	0.03054
<b>4</b>	0.00023	0.00033	0.00026	0.00018	0.00165	0.02967
<b>5</b>	0.00023	0.00033	0.00026	0.00016	0.00163	0.02965
<b>6</b>	0.00022	0.00033	0.00025	0.00016	0.00157	0.02962
<b>7</b>	0.00022	0.00031	0.00025	0.00016	0.00157	0.02788
<b>8</b>	0.00020	0.00031	0.00024	0.00015	0.00147	0.02784
<b>9</b>	0.00020	0.00028	0.00024	0.00015	0.00147	0.02523
<b>10</b>	0.00020	0.00028	0.00023	0.00015	0.00144	0.02520

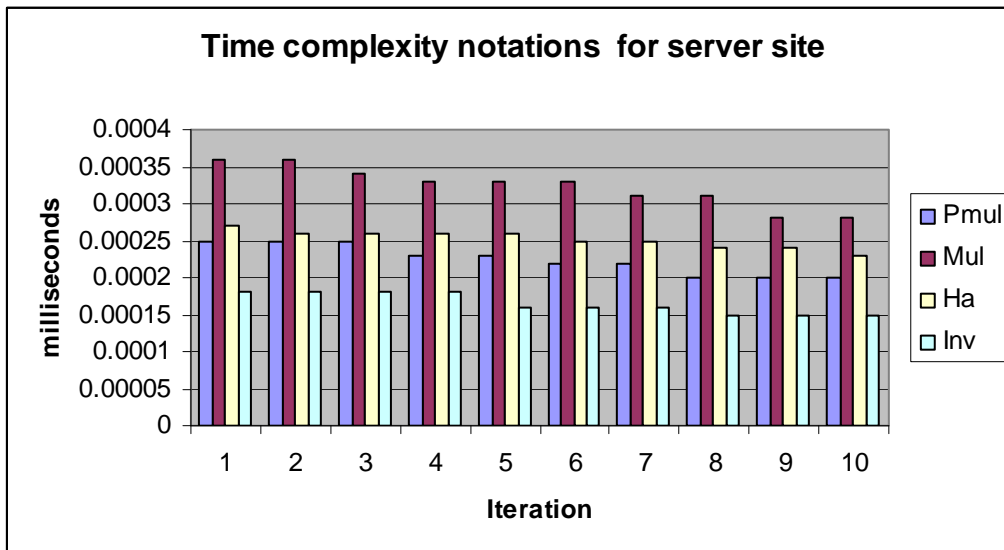
Referred to Table 4.9, the time complexity at server site for first 10 iterations is between 0.00144 milliseconds to 0.00174 milliseconds. Some of the time complexity results at some iteration are same because the values of the time complexity notations are same at those iterations. For example the time complexity at 2<sup>nd</sup> iteration and 3<sup>rd</sup> iteration are same is 0.00171 milliseconds. Same goes to iteration number 6 and iteration number 7 is 0.00157 milliseconds. The results of rough estimation are not same in any iteration because the values of the set of time complexity notations at server site are different

between each other iterations. Figure 4.2 shows the graphical representation of time complexity notations at user site.



**Figure 4.2** Time complexity notations for user site

From Figure 4.2, it can be seen that the  $T_{Mul}$  and  $T_{Ha}$  is decrease from them 1<sup>st</sup> iteration to the 10<sup>th</sup> iteration. The  $T_{Pmul}$  and  $T_{Ha}$  do not differ much if compared with  $T_{Mul}$  from the 1<sup>st</sup> iteration to 10<sup>th</sup> iterations. Figure 4.3 represent the graphical presentation of the time complexity notations for server site.



**Figure 4.3** Time complexity notations for server site

From Figure 4.2, The  $T_{Mul}$  is the highest time to compute when compared to  $T_{Pmul}$ ,  $T_{Ha}$ , and  $T_{Inv}$ . The time for modular multiplication can clearly be seen is decrease from the 1<sup>st</sup> iteration to the 10<sup>th</sup> iterations. The time for Hash operation does not change much from the 1<sup>st</sup> iteration to the 10<sup>th</sup> iterations. Same as  $T_{Ha}$ , time for modular inversion also change slightly from the 1<sup>st</sup> iteration to the 10<sup>th</sup> iterations.

#### 4.4 Conclusion

This chapter covered the comparative study as the qualitative approach of the selected password based authentication scheme and followed by the process of pre-lab testing on the selected password based authentication scheme as the quantitative approach. From the comparative study the appropriate grid security infrastructure and reference password based authentication scheme has been chosen to enhance the reference scheme. In the lab testing findings the time complexity notations has been

searched and used in order to compare the enhanced password based authentication scheme. The result of the pre-lab testing also discussed in this chapter.

## **CHAPTER 5**

### **LAB TESTING AND RESULT ANALYSIS**

#### **5.1 Introduction**

In this chapter, the development of suitable grid security infrastructure is discussed. Beside that, the development and implementation of the selected enhanced password based authentication scheme is also explained and the achieved result is discussed in detail. To understand the enhancement made to the selected password based authentication scheme, the enhancement part is explained and the algorithm for the enhanced password based authentication scheme are presented. Testing has been done on the enhanced password based authentication scheme using real lab environment to test the performance in terms of time complexity notations. The results of enhanced scheme are shown in the form of tables and graphs.

## 5.2 The Development of Suitable Grid Security Infrastructure

As mentioned in Chapter 2, the Password Enable Certificate Free Grid Security Infrastructure (PECF-GSI) is selected as a suitable grid security infrastructure for this project. PECF-GSI is developed by using MIRACL library. MIRACL is a portable C library which implements multiprecision integer and rational data-types, and provides the routines to perform basic arithmetic on them. The development of this infrastructure is started with the development of the Trust Authority (TA). Figure 5.1 shows the part of the algorithm for develop the TA.

```

HRESULT RequestInternalAuthentication(
    WCHAR * wszQueueName,
    WCHAR * wszComputerName,
    ULONG ulAuthLevel
)
{
    // Validate the input strings.
    if (wszQueueName == NULL || wszComputerName == NULL)
    {
        return MQ_ERROR_INVALID_PARAMETER;
    }

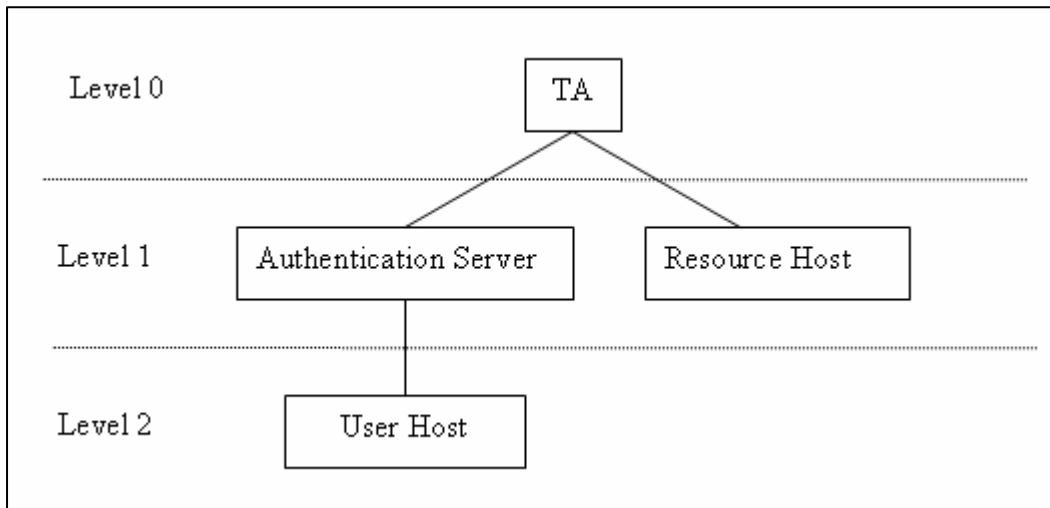
    // Define constants, variables, and structures.
    const int NUMBEROFPROPERTIES = 1;           // Number of properties
    DWORD cPropId = 0;                          // Property counter
    HRESULT hr = MQ_OK;                          // Return code

    MQMSGPROPS msgProps;
    MSGPROPID aMsgPropId[NUMBEROFPROPERTIES];
    MQPROPVARIANT aMsgPropVar[NUMBEROFPROPERTIES];
    HRESULT aMsgStatus[NUMBEROFPROPERTIES];

```

**Figure 5.1** Part of algorithm for Trust Authority

Referred to Figure 5.1, the development of TA is done by using MIRACL library. The TA is known as internal authentication because the user needs to send the request for authentication, then the server will approve the request. User host need to provide the ID to the TA, where that ID is used to generate a key by TA. For this project the hierarchy is used a single TA because one node is involved as a user host. TA is a level 0 entity in that hierarchy. TA will issues private keys to the authentication server. Before user host used the resources from resource host, user authenticate it self to authentication server by password based authentication scheme. The hierarchical relationships between user host and TA is shown in Figure 5.2.



**Figure 5.2** The hierarchical relationships between host and TA

From Figure 5.2, The TA is a level 0 entity where authentication server and resource host are level 1 entity. Just one user host is used for this project, so the user host is alone in level 2 entity. The private key for user host is issued by authentication server and the private key for authentication server is issued by TA.

### **5.3 The Design and Development of Enhanced Password Based Authentication Scheme**

For the enhancement of the password based authentication scheme, the findings of the comparative study that are done in previous chapter are used as the reference. The suitable grid security infrastructure and one of the existing password based authentication scheme has selected to make some enhancement. The enhancement on Rongxing's scheme has made on its elliptic curve cryptosystems function and hash function. The enhancement does not change the protocol of the selected password based authentication scheme. The enhancement which made on Elliptic Curve Cryptosystems (ECC) is about the substitution of supersingular curve in selected scheme to non-supersingular curve in selected password based authentication scheme.

#### **5.3.1 The Enhancement on Elliptic Curve Cryptosystems**

The elliptic curve defined in equation 5.1 is a non-supersingular curve. This non-supersingular curve can be represented by  $F_p$ . It is also known as ordinary elliptic curve which has a nonzero  $j$ -invariant. An elliptic curve over a field of characteristic 2 or 3 is supersingular if and only if it has a zero  $j$ -invariant. For example the elliptic curve defined in equation 5.2 is a supersingular curve. The supersingular curve also can be eliminate the inversions when doubling points or adding a point to itself. The supersingular curve can be represented by  $F_2^f$ .

In term of the level of security, the supersingular curve can provide the same level of security as non-supersingular curve. Arithmetic in  $F_2^r$  can be implemented more efficiently in hardware and software than arithmetic in  $F_p$  (Neal Koblitz *et al*, 2000).  $E(F_2^r)$  can be computed slightly faster than  $E(F_p)$  (Mugino Saeki, 1997). In selected password based authentication scheme, requires a 160-bit prime  $p$  to construct the elliptic curve group  $E(F_p)$  but in enhanced password based authentication scheme, requires a 155-bit to construct the efficient elliptic curve point (Mugino Saeki, 1997). In non-supersingular curve, the limitation for  $p$  is ( $p \approx 160$  bits), but different in supersingular curve. The limitation for  $r$  is  $<160$  bits. The important consequence of using a smaller group or bits in ECC is that low-cost and low-power implementations are feasible. The selected scheme and enhanced scheme requires 1 field inversion and 3 field multiplications, Multiplication in  $F_2^r$  is substantially faster than multiplication in  $F_p$  (Neal Koblitz *et al*, 2000). Figure 5.3 shows the part of the algorithm for elliptic curve point which is enhanced with supersingular curve.

$$y^2 + xy = x^3 + a_2x^2 + a_6 \quad (5.1)$$

$$y^2 + a_3y = x^3 + a_4x + a_6 \quad (5.2)$$

```

//find x^3+ax+B

nres(_MIPP_ x,p->X);

epoint_getrhs(_MIPP_ p->X,mr_mip->w3);

valid=FALSE;

if (x!=y)
{ //compare with y^2
nres(_MIPP_ y,p->Y);
nres_modmult(_MIPP_ p->Y,p->Y,mr_mip->w1);

if (mr_compare(mr_mip->w1,mr_mip->w3)==0) valid=TRUE;
}
else
{ //no y supplied - calculate one. Find square root
#ifdef MR_NOSUPPORT_COMPRESSION

valid=nres_sqrt(_MIPP_ mr_mip->w3,p->Y);
//check LSB - have we got the right root?
redc(_MIPP_ p->Y,mr_mip->w1);
if (remain(_MIPP_ mr_mip->w1,2)!=cb)
mr_psub(_MIPP_ mr_mip->modulus,p->Y,p->Y);

```

**Figure 5.3** Algorithm for elliptic curve point using supersingular curve

From Figure 5.3, the algorithm will set to find the supersingular curve equation. The equation is  $y^2 = x^3 + ax + b$ . The other operation of the elliptic curve point will be done by MIRACL library. MIRACL library is a library for cryptographic computation which is created by Shamus Software Ltd in 2006.

### 5.3.2 The Enhancement on Hash Function

The next enhancement is on hash function. The selected password based authentication scheme is used one-way hash function. The one way hash function will substitute with hash extension. Hash extension is a simple technique to increasing the strength of short one-way and pre-image resistant cryptographic hashes which are counter to the effect of increasing CPU speed (Tuomas Aura *et. al.*, 2007). The hash extension technique also suitable for protocols where the same hash value is verified multiple times. It is useful when a hash of a public key or some other message needs to be communicated over a secure channel that has very limited capacity under 128 bits. Figure 5.4 shows the part of the algorithm of hash extension function.

```

GenerateExtendedHash(M, n, m, out H1, out r)

    r = Rand128();

    ComputeHashes(M, r, n, m, H1, H2);
    while (H2 != 0) r = r + 1;
        ComputeHashes(M, r, n, m, H1, H2);

VerifyExtendedHash(M, r, m, H1, out success)

    n = Length(H1)
    ComputeHashes(M, r, n, m, compH1, compH2);
    if (compH1 == H1 and compH2 == 0)
        success = true;
    else
        success = false;

ComputeHashes(M, r, n, m, out H1, out H2)

    if (m+n <= 160)
        H = SHA-1(M | r);
        H1 = Left(H, n);

```

**Figure 5.4** Algorithm for hash extension generation and verification

Referred to Figure 5.4, the hash extension technique will be generate first than it will be verified. Then after the verification process then the hash will be compute. The bit in hash extension is  $< 160$  bits, but in one way hash is  $< 128$  bits.

#### **5.4 Lab Testing on Enhanced Password Based Authentication Scheme**

For the lab testing on the enhanced password based authentication scheme, again the real lab environment is used, where the same lab setup for the pre-lab testing is used. In previous chapter, there are some results on the time complexity notations, the time complexity and rough estimation in the pre-lab testing which is done on selected password based authentication scheme. The same testing is done on the enhanced password based authentication scheme and the results for both password based authentication scheme are compared and discussed in this section.

The enhanced password based authentication scheme also was run about 20 iterations and the full details of the results as shown in Appendix A5. The results after 10<sup>th</sup> iterations are remaining same. Table 5.1 shows the results for the time complexity notations for the first 10 iterations at user site

**Table 5.1** Results for the time complexity notations at user site

<b>Iterations</b>	<b>T<sub>Pmul</sub></b>	<b>T<sub>Mul</sub></b>	<b>T<sub>Ha</sub></b>
<b>1</b>	0.00017	0.00025	0.00013
<b>2</b>	0.00017	0.00025	0.00013
<b>3</b>	0.00016	0.00024	0.00013
<b>4</b>	0.00016	0.00024	0.00012
<b>5</b>	0.00014	0.00023	0.00012
<b>6</b>	0.00014	0.00023	0.00012
<b>7</b>	0.00013	0.00023	0.00011
<b>8</b>	0.00013	0.00022	0.00011
<b>9</b>	0.00013	0.00021	0.00011
<b>10</b>	0.00013	0.00020	0.00010

Referred to Table 5.1, the time for multiplication of a number an elliptic curve point is between 0.00013 milliseconds to 0.00017 milliseconds. The time for modular multiplication is between 0.00020 milliseconds to 0.00025 milliseconds. The time for hash operation is between 0.00010 milliseconds to 0.00013 milliseconds. Table 5.2 shows the results for the time complexity notations for the first 10 iterations at server site.

**Table 5.2** Results for the time complexity notations at server site

<b>Iterations</b>	<b>T<sub>Pmul</sub></b>	<b>T<sub>Mul</sub></b>	<b>T<sub>Ha</sub></b>	<b>T<sub>Inv</sub></b>
<b>1</b>	0.00015	0.00025	0.00013	0.00011
<b>2</b>	0.00015	0.00025	0.00013	0.00011
<b>3</b>	0.00015	0.00025	0.00012	0.00011
<b>4</b>	0.00015	0.00025	0.00012	0.00010
<b>5</b>	0.00014	0.00024	0.00012	0.00010
<b>6</b>	0.00014	0.00023	0.00011	0.00010
<b>7</b>	0.00012	0.00023	0.00011	0.00010
<b>8</b>	0.00012	0.00022	0.00010	0.00009
<b>9</b>	0.00011	0.00021	0.00010	0.00009
<b>10</b>	0.00011	0.00019	0.00009	0.00009

From Table 5.2, the result for time complexity notations such as time for multiplication of a number and an elliptic curve point is 0.00011 milliseconds to 0.00015 milliseconds. The time for modular multiplication is from 0.00019 milliseconds to 0.00025 milliseconds. The time for hash operation is 0.00009 milliseconds to 0.00013 milliseconds. The hash extension is worked for this enhanced password based authentication scheme. The result for inversion time is 0.00009 milliseconds to 0.00011 milliseconds. To see how well the performance of time complexity notations in selected scheme and enhanced scheme, an analysis is carried out by comparing the selected scheme and the enhanced scheme. The comparison is done in terms of time complexity notations at user site and server site and the results are displayed in Table 5.3.

**Table 5.3** Comparison of time complexity notations

	User site		Server site	
	<i>Selected scheme</i>	<i>Enhanced scheme</i>	<i>Selected scheme</i>	<i>Enhanced scheme</i>
$T_{P_{mul}}$	Less than 0.00023	Less than 0.00017	Less than 0.00025	Less than 0.00015
$T_{Mul}$	Less than 0.00031	Less than 0.00025	Less than 0.00036	Less than 0.00025
$T_{Ha}$	Less than 0.00025	Less than 0.00013	Less than 0.00027	Less than 0.00013
$T_{Inv}$	-	-	Less than 0.00018	Less than 0.00011

Referred to Table 5.3, at user site the time for multiplication a number and an elliptic curve point is less than 0.00023 milliseconds in selected scheme compared to enhanced scheme are 0.00017 milliseconds. The time for modular multiplication at user site is less than 0.00031 milliseconds in selected scheme, but it less than 0.00025 milliseconds in enhanced scheme. The hashing operation time is 0.00025 milliseconds in selected scheme and less than 0.00013 milliseconds in enhanced scheme.

At server site, the selected scheme give the results less than 0.00025 milliseconds for the time for multiplication a number and an elliptic curve point. The time for multiplication is less than 0.00036 milliseconds in selected scheme. Less than 0.00027 milliseconds is the time for hashing operations in selected scheme. Inversion time in selected scheme is less than 0.00018 milliseconds. The time for multiplication a number and an elliptic curve point is less than 0.00015 milliseconds. The multiplication time is less than 0.00025 milliseconds. Less than 0.00013 milliseconds is the results for hashing time in enhanced scheme. The inversion time in enhanced scheme is 0.00011 milliseconds. The table summarize that the results for enhanced scheme are much better than the results from selected scheme. Table 5.4 shows the result for time complexity and rough estimation at user site for first 10 iterations.

**Table 5.4** Time complexity and rough estimation at user site

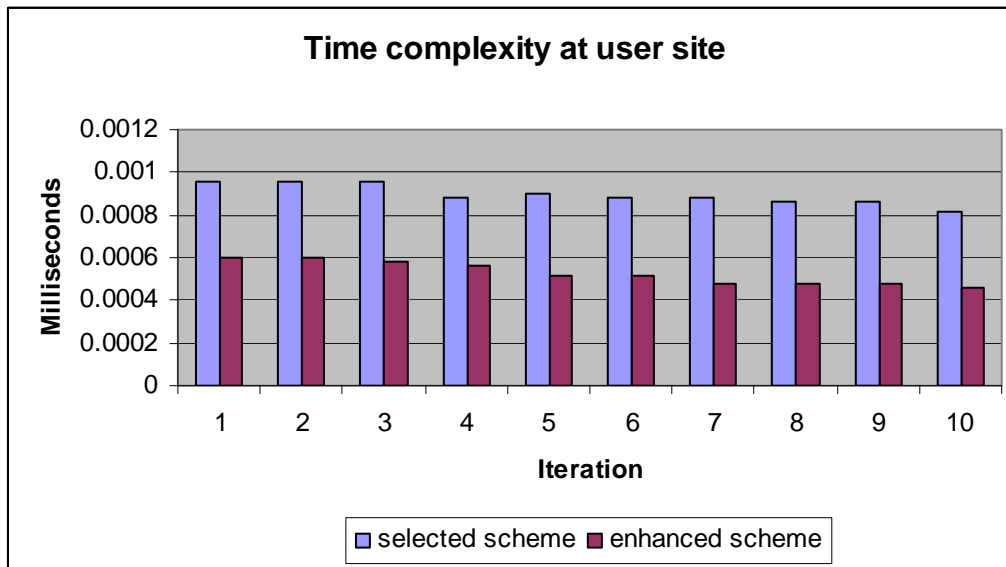
<b>Iterations</b>	<b>T<sub>Pmul</sub></b>	<b>T<sub>Mul</sub></b>	<b>T<sub>Ha</sub></b>	<b>Time complexity (milliseconds)</b>	<b>Rough estimation (milliseconds)</b>
<b>1</b>	0.00017	0.00025	0.00013	0.00060	0.01476
<b>2</b>	0.00017	0.00025	0.00013	0.00060	0.01476
<b>3</b>	0.00016	0.00024	0.00013	0.00058	0.01418
<b>4</b>	0.00016	0.00024	0.00012	0.00056	0.01416
<b>5</b>	0.00014	0.00023	0.00012	0.00052	0.01358
<b>6</b>	0.00014	0.00023	0.00012	0.00052	0.01358
<b>7</b>	0.00013	0.00023	0.00011	0.00048	0.01356
<b>8</b>	0.00013	0.00022	0.00011	0.00048	0.01298
<b>9</b>	0.00013	0.00021	0.00011	0.00048	0.01240
<b>10</b>	0.00013	0.00020	0.00010	0.00046	0.01182

Referred to Table 5.4, the time complexity at user site is between 0.00046 milliseconds to 0.00060 milliseconds. The rough estimation for user site in enhanced password based authentication scheme is from 0.01182 milliseconds to 0.01476 milliseconds. Table 5.5 shows the results of time complexity notations, time complexity and the rough estimation at server site.

**Table 5.5** Time complexity and rough estimation at server site

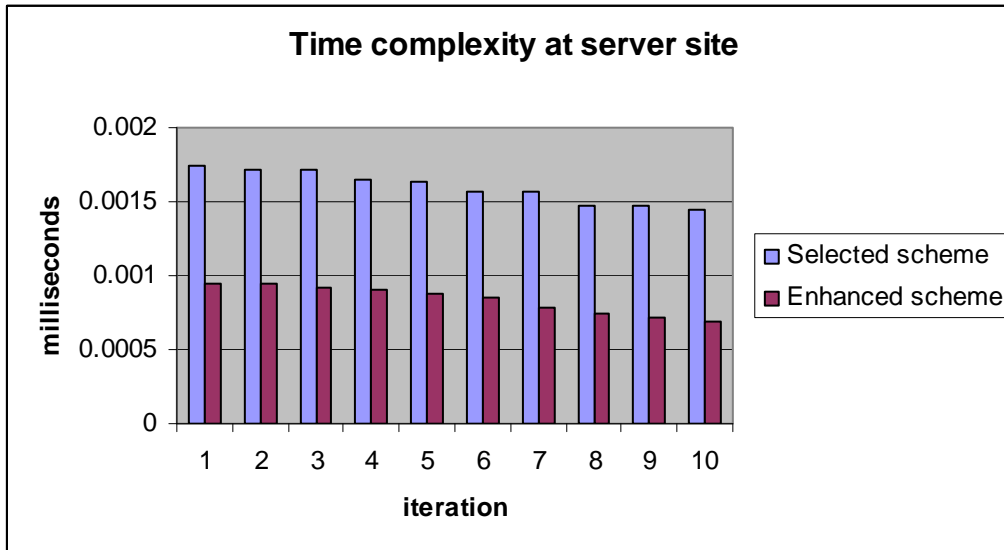
<b>Iterations</b>	<b>T<sub>Pmul</sub></b>	<b>T<sub>Mul</sub></b>	<b>T<sub>Ha</sub></b>	<b>T<sub>Inv</sub></b>	<b>Time complexity (milliseconds)</b>	<b>Rough estimation (milliseconds)</b>
<b>1</b>	0.00015	0.00025	0.00013	0.00011	0.00095	0.02225
<b>2</b>	0.00015	0.00025	0.00013	0.00011	0.00095	0.02225
<b>3</b>	0.00015	0.00025	0.00012	0.00011	0.00092	0.02222
<b>4</b>	0.00015	0.00025	0.00012	0.00010	0.00091	0.02221
<b>5</b>	0.00014	0.00024	0.00012	0.00010	0.00088	0.02134
<b>6</b>	0.00014	0.00023	0.00011	0.00010	0.00085	0.02044
<b>7</b>	0.00012	0.00023	0.00011	0.00010	0.00079	0.02044
<b>8</b>	0.00012	0.00022	0.00010	0.00009	0.00075	0.01956
<b>9</b>	0.00011	0.00021	0.00010	0.00009	0.00072	0.01869
<b>10</b>	0.00011	0.00019	0.00009	0.00009	0.00069	0.01689

From Table 5.5, the time complexity of server site is 0.00069 milliseconds to 0.00095 milliseconds. Around 0.01689 milliseconds to 0.02225 milliseconds are the results for rough estimation at server site. Figure 5.5 shows the comparison of the time complexity between the selected and the enhanced password based authentication scheme at user site.



**Figure 5.5** Time complexity at user site

Referred to Figure 5.5, the time complexity at user site in selected password based authentication scheme is higher than the enhanced password based authentication scheme. It is mean that the enhanced password based authentication scheme is give improved result rather than selected scheme. The comparison on time complexity at server site is displayed in Figure 5.6.



**Figure 5.6** Time complexity at server site

Referred to Figure 5.6, the time complexity at server site in enhanced scheme is much better than the selected scheme. In enhanced scheme, the time complexity is in decreasing order. The difference between time complexity in selected scheme and enhanced scheme is because of the enhancement which is made on the elliptic curve point and also the hashing function. The enhancement on elliptic curve point change the time for modular multiplication, the time for multiplication a number and an elliptic curve point and also the time for inversion. If the supersingular point in elliptic curve point was added, the time for inversion on that moment is 0 milliseconds and it will give a better performance on enhanced scheme. To get 0 milliseconds for inversion time its take some big amount of iterations has to be run. The hashing time in enhanced scheme also give some better results if compared to the selected scheme. Basically, the enhanced scheme gives the better results of time complexity compared to selected scheme. It is mean the aimed of this project is achieved.

## 5.5 Conclusion

In this chapter, details of the suitable grid security infrastructure and enhancement of selected password based authentication scheme have been discussed. The algorithm of the enhancement is included to clear out which part is enhanced from the selected scheme. The results from the lab testing on the enhanced scheme are displayed and discussed. The testing involved the findings of time complexity notations, the time complexity and also the rough estimation for the time complexity at server site and also at user site. Finally, from the results, it can be concluded that the enhanced scheme has an advantage over the selected scheme where it has better performance in term of time complexity while maintaining the security of the password based authentication scheme.

## **CHAPTER 6**

### **DISCUSSION AND CONCLUSION**

#### **6.1 Introduction**

This final chapter covers on the discussion on the result and achievements, the limitations that occur in this project, the future works of the enhanced scheme that can be carried on and finally the conclusion of the overall project.

#### **6.2 Result and Achievements**

In previous chapter, the lab testing has been done on the enhanced scheme to see its performance in terms of its time complexity. The result and analysis in previous chapter shows that the enhanced password based authentication scheme has done some

improvement where the performance in term of time complexity in the time modular multiplication and the time of hashing operation is much lower than the selected scheme while maintaining the security of the scheme. Therefore the aimed for this project which is to enhanced the method of password based authentication scheme and to get better password based authentication scheme has been achieved.

The literature reviews on the existing grid security infrastructure and existing password based authentication scheme become the contribution to the achievement in order to select the most appropriate grid security infrastructure and the selected password based authentication scheme for the enhanced scheme. In addition, all other scopes of project this project have been completed.

The first objective which is to identify, compare and analyze the characteristics of grid computing environments, the security challenges in grid computing, the existing password based authentication scheme, the existing grid security infrastructure and enhance a selected password based authentication scheme in grid computing has been completed by conducting a study focused on the existing password based authentication scheme and existing grid security infrastructure. This study is done by doing comparative studies specifically on the features of existing password based authentication scheme, existing grid security infrastructure, the element of security properties in password based authentication scheme and also the time complexity of the password based authentication scheme.

The second objective is to design and develop the enhanced scheme that will secure the grid computing environment using selected scheme. This has been done by choose an appropriate grid security infrastructure and a selected password based authentication scheme. The enhancement is done by enhancing the elliptic curve cryptosystems and the hash function in selected scheme. The last objective of this project

is to test and implement the enhanced authentication scheme using lab testing. This is done by creating a simple grid environment with three hosts in real lab situation. Then the lab testing is done on the enhanced password based authentication scheme where the results of that is compared with the results of the pre-lab testing and evaluate the performance improvements by its time complexity.

### **6.3 Limitations of the Project**

Conducting a research study on network security specifically on the authentication of grid computing in the duration of 6 months brings some limitations especially in time constraint. Therefore, the overall protocol of the selected password based authentication scheme cannot be enhanced but just its small functions such as elliptic curve cryptosystems and hash function. Doing some research on network security require some simulator to test the scheme and its performance but unfortunately, for this project there are no any simulator is found. Beside that, the real lab environment also need to setup by including many host, but for this project just three hosts are used. Another limitation of the project is that the authors of selected scheme did not specify any result or analysis of their scheme. The low commitment and the incorporation of certain researchers in giving some help or guidance about the method their proposed has also become another constraint in this project.

## **6.4 Future Works**

The improvement of the enhanced scheme should be done in future. For instance, the other part of the selected scheme can be enhanced or improved. The ID of user and resource host should auto detect by the system. Beside that, the enhanced scheme should test or implementing with many more hosts to evaluate its performance accurately.

## **6.5 Conclusion**

As the conclusion, the improvement of the time complexity in the enhanced scheme shown in the lab testing and proved that this project has achieved the project objectives and the aim of the project. However there are some limitations become the reason why the enhanced scheme did not done on the protocol of the selected scheme. The contribution of this project is the enhancement of the elliptic curve cryptosystems and the hash functions of the selected scheme. The development of the suitable grid security infrastructure also becomes another contribution for this project. Even though this project is enhancement of selected password based authentication scheme, there should have some future works done to improve the performance of the enhanced scheme such as in the other part of the protocol.

## REFERENCES

- Aura T, Roe M. *Strengthening Short Hash Values*, Microsoft Research
- Bagwell P. *Ideal Hash Trees*, Es Grands Champs, Switzerland
- Butt. A. R., Adabala, S., Kapadia, N.H., Figueiredo, R. J., Fortes, J.A.B. *Grid Computing Portals and Security Issues*. Journal of Parallel and Distributed Computing. 2003. 63: 1006 – 1014
- Cai Z. A Password based Grid Security Infrastructure. *Second International Conference on the Digital Society*. IEEE Computer Society 2008
- Certicom, *The Elliptic Curve Cryptosystems*, Updated 2000, published 1997.
- Chakrabarti, A. *Grid Computing Security*. Berlin Heidelberg N.Y.: Springer-Verlag. 2007
- Chakrabarti, A., Damodaran, A., Sengupta, S. *Grid Computing Security : A Taxonomy*. IEEE Security & Privacy. IEEE computer society. 2007
- Chivers, H. Grid Security: *Problems and Potential Solutions Computing*. International Journal of Network Security. 2007 Vol.7 No.2 202–206
- Crampton, J., Lim, H.W., Paterson, K.G., Price, G. *A Certificate Free Grid Security Infrastructure Supporting Password Based User Authentication*. UK Engineering and Physical Sciences Research Council (EPSRC). 2008
- Foster I, Kesselman C, Tuecke S, Tsudik G. *A security Architecture for Computational Grids*. Conference on Computer & Communication Security. San Francisco CA USA 1998
- Hu, H., Yao, H. *A scheme for Authentication and Authorization in a Grid Application*. Conference on Advanced Information Networking and Application. IEEE. 2005
- Koblitz N, Menezes A, Vanstone S, *The State of Elliptic Curve Cryptography*, Designs, Codes and Cryptography. 2000. Kluwer Academic Publishers, Boston.

- Lu, R., Cao, Z., Chai, Z., Liang, X., *A Simple User Authentication Scheme for Grid Computing*.
- Manish Mehta, *Authentication services in Open Grid Services*. Ph.D. University of Missouri, USA:2004
- Saeki M, (1997) *Elliptic Curve Cryptosystems*, School of Computer science, McGill University, Montreal.
- Shamus Software Ltd, *MIRACL User Manual*, Ballybough, Ireland.
- Thompson, M. R., Jackson, K. R. *Security Challenges in Supporting Grid Computing and Collaboration*. Distributed System Department - Lawrence Berkeley National Laboratory
- Welch V., Siebenlist, F., Foster I., Bresnahan J., Czajkowaki K., Gawor J., Kesselman C., Meder S., Pearlman L., Tuecke S. *Security for Grids Services*.
- Welch V., Pearlman L., Foster I., Kesselman C., Tuecke S. *A Community Authorization Service for Group Collaboration*
- Wu, R., Li, R., Yu, F., Yue, G., Xu, C. *Research on User Authentication for Grid Computing Security*. Second International Conference on Semantics, Knowledge, and Grid. 2006. IEEE Computer Society. 2006.
- Yoon, E.J., Yoo, K.Y., (2005) *An Efficient Password Authentication Schemes Without Using the Server Public Key for Grid Computing*. In Zhuge H and Fox G.C *Grid and Cooperative Computing* (149-154). Berlin Heidelberg: Springer-Verlag.

## APPENDIX A1

**Protocol of Password Authentication Schemes without Server Public Key**

The notations used in this protocol

- $U, S, E$ : client, trusted server and attacker, respectively.
- $id$ : public user identity of client.
- $pw$ : secret and possibly weak user password.
- $T, K$ : timestamp and secret key of server.
- $p, q$ : large prime numbers  $p$  and  $q$  such that  $q|p - 1$ .
- $g$ : generator with order  $q$  in the Galois field  $GF(p)$ , in which Diffie-Hellman problem is of hard.
- $r1, r2$ : session-independent random exponents  $\in [1, q - 1]$  chosen by client and server, respectively.
- $sk$ : shared session key computed by client and server.
- $flow[i]$ : data transmitted in the  $i$ -th step.
- $E(pw, m), E1(pw, m)$ : symmetric encryption scheme of message  $m$  with  $pw$ .
- $E(sk, m), E2(sk, m)$ : symmetric encryption scheme of message  $m$  with  $sk$ .
- $H(\cdot), ||$ : strong one-way hash function and concatenation symbol.

**Protected Password Transmission Scheme:** the server stores  $ID, E2(K, pw)$  for each client in the database.

$$(1) \quad U \rightarrow S: id, T, E1(pw, gr1 \bmod p)$$

$U$  chooses a random number  $r1$  and computes  $E1(pw, gr1 \bmod p)$ . Then,  $U$  sends the computation result with  $id$  and the timestamp  $T$  to  $S$  as a login request.

$$(2) \quad S \rightarrow U: E1(pw, gr2 \bmod p), E2(sk, H(flow[1]))$$

$S$  uses  $pw$  to retrieve  $gr1 \bmod p$ .  $S$  generates a random number  $r2$  and computes

- the session key  $sk = (gr1)r2 \bmod p$ . Thereupon,  $S$  computes  $E1(pw, gr2 \bmod p)$  and  $E2(sk, H(flow[1]))$ , and sends the results to  $U$ .
- (3)  $U \rightarrow S: id, E2(sk, H(flow[2]))$   
 $U$  uses  $pw$  to retrieve  $gr2 \bmod p$  and computes the session key  $sk = (gr2)r1 \bmod p$ . Then,  $U$  authenticates  $S$  by checking whether the decryption result of  $E2(sk, H(flow[1]))$  with the session key  $sk$  is equal to the hash value of the data sent by himself/herself in Step (1). If it holds,  $U$  computes and sends  $E2(sk, H(flow[2]))$  with  $id$  to  $S$ .
- (4)  $S \rightarrow U: Access\ granted\ or\ denied$   
 $S$  decrypts  $E2(sk, H(flow[2]))$  and compares the result with the hash value of the transmitted data in Step (2).  $S$  will grant  $U$  the access right if it holds.

**Protected Password Change Scheme:** The steps of Chang et al.'s protected password change scheme are almost the same as those of the password transmission scheme, except for an additional password change request in Step (3).

- (1)  $U \rightarrow S: id, E1(pw, gr1 \bmod p)$   
 (2)  $S \rightarrow U: E1(pw, gr2 \bmod p), E2(sk, H(flow[1]))$   
 (3)  $U \rightarrow S: id, E2(sk, H(flow[2])), E2(sk, (pw\_||T))$   
 $U$  sends  $id$ ,  $E2(sk, H(flow[2]))$  and  $E2(sk, (pw\_||T))$  to  $S$ , where  $pw_$  is the new password chosen by  $U$ .  
 (4)  $S \rightarrow U: Access\ granted\ or\ denied$   
 $S$  decrypts  $E2(sk, (pw\_||T))$  with the session key  $sk$  to get  $U$ 's new password  $pw_$  and  $T$ .  $S$  compares the timestamp  $T$  with the transmitted one in  $flow[1]$  to authenticate  $U$ .

## APPENDIX A2

**Protocol of Efficient Password Authentication Schemes without Server Public key**

The notations used in this protocol

- $U, S, E$ : client, trusted server and attacker, respectively.
- $id$ : public user identity of client.
- $pw$ : secret and possibly weak user password.
- $T, K$ : timestamp and secret key of server.
- $p, q$ : large prime numbers  $p$  and  $q$  such that  $q|p - 1$ .
- $g$ : generator with order  $q$  in the Galois field  $GF(p)$ , in which Diffie-Hellman problem is of hard.
- $r1, r2$ : session-independent random exponents  $\in [1, q - 1]$  chosen by client and server, respectively.
- $sk$ : shared session key computed by client and server.
- $flow[i]$ : data transmitted in the  $i$ -th step.
- $E(pw, m), E1(pw, m)$ : symmetric encryption scheme of message  $m$  with  $pw$ .
- $E(sk, m), E2(sk, m)$ : symmetric encryption scheme of message  $m$  with  $sk$ .
- $H(\cdot), ||$ : strong one-way hash function and concatenation symbol.

**Protected Password Transmission Scheme:**

$$(1) \quad U \rightarrow S: id, E(pw, gr1)$$

$U$  chooses a random number  $r1$  and computes  $E(pw, gr1)$ . Then,  $U$  sends the computation result with  $id$  to  $S$  as a login request.

$$(2) \quad S \rightarrow U: gr2, H(sk, gr1)$$

After receiving a login request,  $S$  uses  $pw$  to retrieve  $gr1$ .  $S$  generates a random number  $r2$  and computes the session key  $sk = (gr1)r2$ . Thereupon,  $S$  computes  $gr2$  and  $H(sk, gr1)$ , and sends the results to  $U$ .

(3)  $U \rightarrow S: id, H(sk_-, gr2)$

$U$  computes the session key  $sk_- = (gr2)r1$  and authenticates  $S$  by checking whether  $H(sk, gr1) = H(sk_-, gr1)$  holds. If it holds,  $U$  computes  $H(sk_-, gr2)$  and sends it with  $id$  to  $S$ .

(4)  $S \rightarrow U: Access\ granted\ or\ denied$

$S$  computes the hash value  $H(sk, gr2)$  using its own copies of  $sk$  and  $gr2$  and determines whether  $H(sk, gr2) = H(sk_-, gr2)$  holds or not. If it holds,  $S$  will grant  $U$  access. After mutual authentication between  $U$  and  $S$ ,  $sk = sk_- = gr1r2$  is used as the session key.

**Protected Password Change Scheme:** The protected password change scheme allows  $U$  to change its old password  $pw$  to a new password  $pw_-$ . The proposed protected password change scheme is much the same as the protected password transmission scheme, except for Steps (1) and (4). The proposed protected password change scheme is works as follows:

(1)  $U \rightarrow S: id, E(pw, (pw_- || gr1))$

$U$  chooses a random number  $r1$  and a new password  $pw_-$ , and computes  $E(pw, (pw_- || gr1))$ . Then,  $U$  sends the computation result with  $id$  to  $S$  as a login request.

(2)  $S \rightarrow U: gr2, H(sk, gr1)$

(3)  $U \rightarrow S: id, H(sk_-, gr2)$

(4)  $S \rightarrow U: Access\ granted\ or\ denied$

$S$  computes the hash value  $H(sk, gr2)$  using its own copies of  $sk$  and  $gr2$  and determines whether  $H(sk, gr2) = H(sk_-, gr2)$  holds or not. If it holds,  $S$  will grant  $U$  access and replaces  $E(K, pw)$  with  $E(K, pw_-)$ . After mutual Authentication between  $U$  and  $S$ ,  $sk = sk_- = gr1r2$  is used as the session key.

## APPENDIX A3

**Protocol of Authentication Scheme Based on Elliptic Curve Cryptosystem**

Some used notations in this protocol

- U, S: user and server in grid computing
- ID: public identity of user U
- G, P: subgroup of the elliptic curve group  $\mathbf{E}(\mathbf{F}_p)$  and its generator point of order  $q$ .
- D: uniformly distributed dictionary of size  $[D] = 2^k$ , as usual,  $48 \leq k \leq 112$ .
- $pw$ : low-entropy human-memorable password extracted from D.
- K: secret key of server S, which is only known by the server and must be safeguarded.
- $h$ : secure one-way hash function, where  $h: \{0, 1\}^* \rightarrow \{0, 1\}^l$  and  $l = 160$ .
- $[m]^k$ : the most significant  $k$  bits of string  $m$ .
- $i$ : shelf life of low entropy human-memorable password.

***Registration phase***

In the registration phase, user U submits his identity ID to register himself to the server S. After checking the valid identity ID, the server S chooses a shelf life  $I$  and uses her secret key K to compute the hash value  $v = h(K||ID||i)$ . Then, she generates U's password  $pw = [v]^k$  and returns  $(pw, i)$  to U. And thus user U holds the human-memorable password  $pw$  and its shelf life  $i$ . Server does not need to maintain a verification table in database.

### ***Authentication phase***

Step 1: U chooses a random  $r_1 \in \mathbb{Z}_q^*$ , computes  $R_1 = (pw.r_1)P$ , and sends  $(ID, R_1, i)$  to S

Step 2: S first checks the shelf life  $i$ . If it is valid, continue; otherwise stop. Then, S computes  $v = h(K||ID||i)$ ,  $pw = [v]^k$  and  $r_1' = pw^{-1}R_1 = (pw^{-1}.pw.r_1)P = r_1P$ . S chooses another random  $r_2 \in \mathbb{Z}_q^*$ , computes  $R_2 = r_2P, sk = r_2R_1' = r_1r_2P$  and  $h_1 = h(sk||R_2)$ . Finally, S sends  $(R_2, h_1)$  to U.

Step 3: U computes  $sk=r_1R_2 = r_1r_2P$  and checks whether  $h(sk||R_2) = h_1$  holds. If it does hold, S is authenticated. Then, U computes  $h_2 = h(sk||ID)$  and sends it to S

Step 4: S computes  $h_2' = h(sk||ID)$  and compares whether  $h_2' = h_2$  or not. If they are equal, U is authenticated and granted to access the resources by S. In addition, after the mutual authentication between U and S,  $sk = r_1r_2P$  will be used as a session key for further operations.

### ***Password change phase***

After a common session key  $sk = r_1r_2P$  is shared between U and S, they can establish a secure channel between them. Then, when U wants to change his password in its shelf life, he can securely request a new password as follows:

Step 1: U sends his identity ID, old password pw and the shelf life  $i$  to S.

Step 2: S checks whether  $pw = [h(K||ID||i)]^k$  hold or not. If it does hold, S chooses a new shelf life  $i'$  and  $pw' = [h(K||ID||i')]^k$ , then  $(pw', i')$  back to U. Thus U can hold a new password  $pw'$  and its shelf life  $i'$ .

## APPENDIX A4

**Result for Selected Password Based Authentication Scheme**

User site

<b>Iterations</b>	<b>T<sub>Pmul</sub></b>	<b>T<sub>Mul</sub></b>	<b>T<sub>Ha</sub></b>	<b>Time complexity (milliseconds)</b>	<b>Rough estimation (milliseconds)</b>
<b>1</b>	0.00023	0.00031	0.00025	0.00096	0.01848
<b>2</b>	0.00023	0.00031	0.00025	0.00096	0.01848
<b>3</b>	0.00023	0.00031	0.00025	0.00096	0.01848
<b>4</b>	0.00021	0.00030	0.00023	0.00088	0.01786
<b>5</b>	0.00022	0.00029	0.00023	0.00090	0.01728
<b>6</b>	0.00021	0.00029	0.00023	0.00088	0.01728
<b>7</b>	0.00021	0.00027	0.00023	0.00088	0.01612
<b>8</b>	0.00022	0.00027	0.00021	0.00086	0.01608
<b>9</b>	0.00022	0.00027	0.00021	0.00086	0.01608
<b>10</b>	0.00020	0.00025	0.00021	0.00082	0.01492
<b>11</b>	0.00020	0.00025	0.00021	0.00082	0.01492
<b>12</b>	0.00020	0.00025	0.00021	0.00082	0.01492
<b>13</b>	0.00020	0.00025	0.00021	0.00082	0.01492
<b>14</b>	0.00020	0.00025	0.00021	0.00082	0.01492
<b>15</b>	0.00020	0.00025	0.00021	0.00082	0.01492
<b>16</b>	0.00020	0.00025	0.00021	0.00082	0.01492
<b>17</b>	0.00020	0.00025	0.00021	0.00082	0.01492
<b>18</b>	0.00020	0.00025	0.00021	0.00082	0.01492
<b>19</b>	0.00020	0.00025	0.00021	0.00082	0.01492
<b>20</b>	0.00020	0.00025	0.00021	0.00082	0.01492

## Server Site

<b>Iterations</b>	<b>T<sub>Pmul</sub></b>	<b>T<sub>Mul</sub></b>	<b>T<sub>Ha</sub></b>	<b>T<sub>Inv</sub></b>	<b>Time complexity (milliseconds)</b>	<b>Rough estimation (milliseconds)</b>
<b>1</b>	0.00025	0.00036	0.00027	0.00018	0.00174	0.03231
<b>2</b>	0.00025	0.00036	0.00026	0.00018	0.00171	0.03228
<b>3</b>	0.00025	0.00034	0.00026	0.00018	0.00171	0.03054
<b>4</b>	0.00023	0.00033	0.00026	0.00018	0.00165	0.02967
<b>5</b>	0.00023	0.00033	0.00026	0.00016	0.00163	0.02965
<b>6</b>	0.00022	0.00033	0.00025	0.00016	0.00157	0.02962
<b>7</b>	0.00022	0.00031	0.00025	0.00016	0.00157	0.02788
<b>8</b>	0.00020	0.00031	0.00024	0.00015	0.00147	0.02784
<b>9</b>	0.00020	0.00028	0.00024	0.00015	0.00147	0.02523
<b>10</b>	0.00020	0.00028	0.00023	0.00015	0.00144	0.02520
<b>11</b>	0.00020	0.00028	0.00023	0.00015	0.00144	0.02520
<b>12</b>	0.00020	0.00028	0.00023	0.00015	0.00144	0.02520
<b>13</b>	0.00020	0.00028	0.00023	0.00015	0.00144	0.02520
<b>14</b>	0.00020	0.00028	0.00023	0.00015	0.00144	0.02520
<b>15</b>	0.00020	0.00028	0.00023	0.00015	0.00144	0.02520
<b>16</b>	0.00020	0.00028	0.00023	0.00015	0.00144	0.02520
<b>17</b>	0.00020	0.00028	0.00023	0.00015	0.00144	0.02520
<b>18</b>	0.00020	0.00028	0.00023	0.00015	0.00144	0.02520
<b>19</b>	0.00020	0.00028	0.00023	0.00015	0.00144	0.02520
<b>20</b>	0.00020	0.00028	0.00023	0.00015	0.00144	0.02520

## APPENDIX A5

**Result for Enhanced Password Based Authentication Scheme**

User site

<b>Iterations</b>	<b>T<sub>Pmul</sub></b>	<b>T<sub>Mul</sub></b>	<b>T<sub>Ha</sub></b>	<b>Time complexity (milliseconds)</b>	<b>Rough estimation (milliseconds)</b>
<b>1</b>	0.00017	0.00025	0.00013	0.00060	0.01476
<b>2</b>	0.00017	0.00025	0.00013	0.00060	0.01476
<b>3</b>	0.00016	0.00024	0.00013	0.00058	0.01418
<b>4</b>	0.00016	0.00024	0.00012	0.00056	0.01416
<b>5</b>	0.00014	0.00023	0.00012	0.00052	0.01358
<b>6</b>	0.00014	0.00023	0.00012	0.00052	0.01358
<b>7</b>	0.00013	0.00023	0.00011	0.00048	0.01356
<b>8</b>	0.00013	0.00022	0.00011	0.00048	0.01298
<b>9</b>	0.00013	0.00021	0.00011	0.00048	0.01240
<b>10</b>	0.00013	0.00020	0.00010	0.00046	0.01182
<b>11</b>	0.00013	0.00020	0.00010	0.00046	0.01182
<b>12</b>	0.00013	0.00020	0.00010	0.00046	0.01182
<b>13</b>	0.00013	0.00020	0.00010	0.00046	0.01182
<b>14</b>	0.00013	0.00020	0.00010	0.00046	0.01182
<b>15</b>	0.00013	0.00020	0.00010	0.00046	0.01182
<b>16</b>	0.00013	0.00020	0.00010	0.00046	0.01182
<b>17</b>	0.00013	0.00020	0.00010	0.00046	0.01182
<b>18</b>	0.00013	0.00020	0.00010	0.00046	0.01182
<b>19</b>	0.00013	0.00020	0.00010	0.00046	0.01182
<b>20</b>	0.00013	0.00020	0.00010	0.00046	0.01182

Server site

<b>Iterations</b>	<b>T<sub>Pmul</sub></b>	<b>T<sub>Mul</sub></b>	<b>T<sub>Ha</sub></b>	<b>T<sub>Inv</sub></b>	<b>Time complexity (milliseconds)</b>	<b>Rough estimation (milliseconds)</b>
<b>1</b>	0.00015	0.00025	0.00013	0.00011	0.00095	0.02225
<b>2</b>	0.00015	0.00025	0.00013	0.00011	0.00095	0.02225
<b>3</b>	0.00015	0.00025	0.00012	0.00011	0.00092	0.02222
<b>4</b>	0.00015	0.00025	0.00012	0.00010	0.00091	0.02221
<b>5</b>	0.00014	0.00024	0.00012	0.00010	0.00088	0.02134
<b>6</b>	0.00014	0.00023	0.00011	0.00010	0.00085	0.02044
<b>7</b>	0.00012	0.00023	0.00011	0.00010	0.00079	0.02044
<b>8</b>	0.00012	0.00022	0.00010	0.00009	0.00075	0.01956
<b>9</b>	0.00011	0.00021	0.00010	0.00009	0.00072	0.01869
<b>10</b>	0.00011	0.00019	0.00009	0.00009	0.00069	0.01689
<b>11</b>	0.00011	0.00019	0.00009	0.00009	0.00069	0.01689
<b>12</b>	0.00011	0.00019	0.00009	0.00009	0.00069	0.01689
<b>13</b>	0.00011	0.00019	0.00009	0.00009	0.00069	0.01689
<b>14</b>	0.00011	0.00019	0.00009	0.00009	0.00069	0.01689
<b>15</b>	0.00011	0.00019	0.00009	0.00009	0.00069	0.01689
<b>16</b>	0.00011	0.00019	0.00009	0.00009	0.00069	0.01689
<b>17</b>	0.00011	0.00019	0.00009	0.00009	0.00069	0.01689
<b>18</b>	0.00011	0.00019	0.00009	0.00009	0.00069	0.01689
<b>19</b>	0.00011	0.00019	0.00009	0.00009	0.00069	0.01689
<b>20</b>	0.00011	0.00019	0.00009	0.00009	0.00069	0.01689