



Norwegian University of  
Science and Technology

# Secure Mobile Authentication for Linux Workstation log on

**Usman Habib**

Master of Telematics - Communication Networks and  
Networked Services (2 year)

Submission date: June 2010

Supervisor: Van Thanh Do, ITEM

Co-supervisor: Ivar Jørstad, Ubisafe AS

Norwegian University of Science and Technology  
Department of Telematics



# Problem Description

The Master thesis work will study and propose how workstation identity management can be made more user-friendly and secure by using the mobile phone in the Linux workstation log on process. The current log on process is neither user friendly nor sufficiently secure. This work consists of the following tasks:

1. Study of existing identity management and authentication for enterprise network and workstations.
2. Study of Platform security architecture of Linux.
3. Study of Authentication and identity management standards and protocols (e.g. EAP, RADIUS/DIAMETER, SAML etc.)
4. Design and implementations of solutions for integration of the mobile phone with the workstation logon processes in different operating systems
5. Threat and vulnerability analysis of proposed solution(s).

Assignment given: 21. January 2010

Supervisor: Van Thanh Do, ITEM



# Preface

This thesis is done as Master thesis in the Master of Telematics: Communication Networks and Networked Services program at the Norwegian University of Science and Technology (NTNU). The thesis has been performed in the spring semester 2010 at the department of Telematics in collaboration with Telenor R&I.

The supervisor for this thesis has been Ivar Jørstad at UbiSafe AS and the academic responsible has been professor Do van Thanh. I would like to thank them both for their help and support during the work with the thesis and especially Do Van Thanh for the valuable suggestions and comments.

Trondheim, June 17, 2010

Usman Habib.



# Abstract

Password based logon schemes have many security weaknesses. For secure environments smart card and biometric based authentication solutions are available as replacement for standard password based systems. Nevertheless, the cost of deployment and maintenance of these systems is quite high. On the other hand, mobile network operators have a huge base of deployed smart cards that can be reused to provide authentication in other spheres significantly reducing costs. In this project we present a study of how mobile phones can be used to provide a secure low-cost two-factor workstation logon solution.

To find and study the available mobile phone based authentication architectures and come up with workstation logon architecture the study of relevant technologies utilized in these solutions: UMTS networks, Bluetooth communication, Remote Authentication Dial in User Service (RADIUS), authentication and authorization in Windows, Linux, and MAC OS X. The analysis of available mobile phone based authentication schemes like SIM Strong schemes based on EAP-SIM, Session-ID based schemes, and OTP based schemes are also added.

A solution for Linux workstation logon process has been proposed in the thesis using the Pluggable Authentication Module (PAM). The Solution uses 2 factors for authentication, one is the static password and the second factor is the mobile based authentication using a 13 character long OTP. With adding the existing technology and giving the administrator the option of selecting the authentication method for user makes the solution more suitable for an enterprise.





# Table of Contents

<b>1. Introduction</b> .....	<b>1</b>
1.1 Motivation.....	1
1.2 Problem definition.....	2
1.3 Objectives .....	2
1.4 Organization of the thesis .....	3
<b>2. Background</b> .....	<b>4</b>
2.1 Identity and identity management.....	4
2.2 Identity management in enterprise systems.....	8
2.3 Smart card technology.....	19
2.4 Identity management in UMTS system.....	23
2.5 Authentication and authorization in Linux.....	27
2.6 Authentication and authorization in Windows .....	36
2.7 Authentication and authorization in Mac OS X .....	42
2.8 Remote Authentication Dial In User Service (RADIUS) .....	46
2.9 Bluetooth security .....	48
<b>3. Mobile Phone Based Authentication Systems</b> .....	<b>53</b>
3.1 General Mobile Phone Authentication Schemes .....	53
3.1.1 SMS Authentication with Session-ID check .....	53
3.1.2 SIM Strong Authentication .....	54
3.1.3 EAP-AKA .....	56
3.2 One Time Password Schemes .....	56
3.2.1 One Time Password From PC to SMS .....	57
3.2.2 One Time Password From SMS to PC .....	59
3.2.3 Enhanced OTP From PC to SMS authentication.....	61
3.3 Proposed Solution for Mobile Based Linux Workstation Logon Service.....	64
<b>4. Analysis</b> .....	<b>67</b>
4.1 Requirements .....	67
4.1.1 Functional Requirements .....	67
4.1.2 Non Functional Requirements .....	67

4.2 Use Case Diagrams .....	67
4.3 Interaction Diagrams .....	72
4.3.1 Successful Authentication.....	72
<b>5. Design .....</b>	<b>74</b>
5.1 Components .....	74
5.1.1 PAM Module .....	74
5.1.2 Mobile Phone Module .....	75
5.1.3 Authentication Server .....	76
5.1.4 MoBA Server .....	76
5.2 Package and Class Diagrams.....	77
<b>6. Implementation .....</b>	<b>81</b>
6.1 Deployment.....	81
6.2 Implementation of the Components of the solution .....	81
6.2.1 PAM Module .....	81
6.2.2 Authentication Server .....	82
6.2.3 Authenticator .....	82
6.2.4 One Time Password (OTP).....	83
6.2.5 RADIUS Authentication Server.....	83
6.2.6 Mobile Phone Module .....	83
6.2.7 SMS Gateway.....	83
6.3 Testing the Prototype .....	83
<b>7. Validation and Evaluation .....</b>	<b>87</b>
7.1 Validating the Functionality against Requirements .....	87
7.2 Different Types of Attack .....	88
7.3 Security Evaluation of the solution.....	89
<b>8. Conclusion.....</b>	<b>91</b>
<b>References.....</b>	<b>93</b>
<b>Appendix A.....</b>	<b>98</b>
<b>Appendix B .....</b>	<b>100</b>

# List of Figures

<b>Figure 1:</b> Partnership Identity Management [2].....	8
<b>Figure 2:</b> Identity and access manager architecture .....	11
<b>Figure 3:</b> Kerberos authentication model .....	13
<b>Figure 4:</b> ALL IP UMTS.....	24
<b>Figure 5:</b> UMTS architecture [40] .....	25
<b>Figure 6:</b> Mutual Authentication and Key Agreement [40].....	26
<b>Figure 7:</b> Login Process in Linux.....	31
<b>Figure 8:</b> Mapping between ACL entries & File mode Permission Bits [50] .....	33
<b>Figure 9:</b> PAM Application Structure .....	34
<b>Figure 10:</b> Interaction of NetInfo and Users [72] .....	42
<b>Figure 11:</b> Administrator Management with NetInfo [72] .....	43
<b>Figure 12:</b> Shared Domain in NetInfo [72] .....	43
<b>Figure 13:</b> NetInfo Hierarchies [72].....	43
<b>Figure 14:</b> RADIUS Architecture [75].....	46
<b>Figure 15:</b> RADIUS Different Authentication Operations .....	47
<b>Figure 16:</b> RADIUS Message Format.....	47
<b>Figure 17:</b> SSP Link Key Establishment for pairing [78] .....	49
<b>Figure 18:</b> Bluetooth Authentication [42].....	51
<b>Figure 19:</b> Bluetooth Encryption [78] .....	52
<b>Figure 20:</b> SMS authentication with sessionID check .....	54
<b>Figure 21:</b> EAP-SIM authentication.....	55
<b>Figure 22:</b> EAP-AKA authentication .....	56
<b>Figure 23:</b> OTP from PC to phone authentication [3].....	58
<b>Figure 24:</b> OTP Applet [3].....	59
<b>Figure 25:</b> OTP SMS to PC authentication [3] .....	60
<b>Figure 26:</b> Component Architecture [5] .....	61
<b>Figure 27:</b> Authentication process.....	62
<b>Figure 28:</b> Key Exchange procedure [5] .....	63
<b>Figure 29:</b> Mobile Based Authentication System for Linux Workstation Logon.....	65
<b>Figure 30:</b> General Use Case Diagram.....	68
<b>Figure 31:</b> OTP Generation Use Case .....	69
<b>Figure 32:</b> Sequence Diagram- Authentication.....	73
<b>Figure 33:</b> Main Component Diagram. ....	74
<b>Figure 34:</b> PAM Module Component Diagram. ....	75
<b>Figure 35:</b> Mobile Phone Component Diagram.....	76

<b>Figure 36:</b> Authentication Server Component Diagram. ....	76
<b>Figure 37:</b> MoBA Server Component Diagram. ....	77
<b>Figure 38:</b> MoBA System for Linux Login Service Package Diagram. ....	78
<b>Figure 39:</b> PAM Module Class Diagram. ....	79
<b>Figure 40:</b> Mobile Phone Module Class Diagram. ....	79
<b>Figure 41:</b> MoBA Server Class Diagram. ....	80
<b>Figure 42:</b> Authentication Server Class Diagram. ....	80
<b>Figure 43:</b> Secure MoBA for Linux Workstation Logon Deployment Diagram .....	81
<b>Figure 44:</b> Users in the LDAP Authentication Server.....	84
<b>Figure 45:</b> User Information in LDAP Server .....	84
<b>Figure 46:</b> Server Waiting for Clients. ....	84
<b>Figure 47:</b> LDAP Authentication .....	85
<b>Figure 48:</b> OTP Generation Snapshot .....	85
<b>Figure 49:</b> OTP received at User Mobile.....	85
<b>Figure 50:</b> User Authentication Snapshot.....	86
<b>Figure 51:</b> Component Threat Model.....	89

# List of Tables

<b>Table 1:</b> Identities in UMTS components [40].....	25
<b>Table 2:</b> Linux built-in groups.....	28
<b>Table 3:</b> Types of ACL entries [50] .....	32
<b>Table 4:</b> Masking of permissions [50] .....	33
<b>Table 5:</b> Return Codes .....	35
<b>Table 6:</b> Built-in accounts in Windows.....	36
<b>Table 7:</b> Well-known system groups in Windows .....	37
<b>Table 8:</b> Logon rights in Windows .....	38
<b>Table 9:</b> Authorization in MAC OS.....	45
<b>Table 10:</b> Rule Attributes and Description [73] .....	45
<b>Table 11:</b> Tokens installed with MAC OS .....	46
<b>Table 12:</b> Use Case-Login Service .....	68
<b>Table 13:</b> Use Case-Maintain User Information .....	69
<b>Table 14:</b> Use Case-OTP Generation.....	70
<b>Table 15:</b> Use Case-Get User Information.....	70
<b>Table 16:</b> Use Case-Generate OTP.....	71
<b>Table 17:</b> Use Case-Send OTP.....	71
<b>Table 18:</b> Use Case-Perform Authentication.....	71
<b>Table 19:</b> Use Case-OTP SMS .....	72



# Abbreviations

AAA	Authentication, Authorization and Accounting
ACL	Access Control List
AES	Advanced Encryption Standard
AK	Authentication Key
AKA	Authentication and Key Agreement
API	Application Programming Interface
AS	Authentication Server
AuC	Authentication Center
BSC	Base Station Controller
BSS	Base Station Subsystem
BTS	Base Transceiver Station
CK	Cipher Key
DAC	Discretionary Access Control
EAL	Evaluation assurance level
EAP	Extensible Authentication Protocol
ECDH	Elliptic Curve Diffie Hellman
EIR	Equipment Identity Register
FAST	Flexible authentication secure tunneling
GGSN	GPRS support node
GID	Group Identity
GINA	Graphical Identification and Authentication
GSM	Global System for Mobile Communications
HLR	Home Location Register
HMAC	Hash-based Message Authentication Code
HSS	Home subscription server
HTTP	Hypertext Transfer Protocol
IAM	Identity and access control management
ID	Identity
ID-FF	Identity Federation Framework
IdM	Identity Management
IK	Integrity Key
IMEI	International Mobile Equipment Identity
IMEISV	International Mobile Station Equipment Identity and Software Number
IMSI	International Mobile Station Identity
ISIM	IP multimedia Services Identity Module
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IMSI	International Mobile Subscriber Identity
JAD	Java Application Descriptor File
JME	Java Micro Edition
KDC	Key Distribution Center
LDAP	Light weight directory Access protocol
LFSR	Linear Feedback Shift Register
LMP	Link Management Protocol
MAC	Message Authentication Code
ME	Mobile Equipment
MGW	Media Gateway
MITM	Man -In-The-Middle
MNO	Mobile Network Operator

MoBA	Mobile Based Authentication
MS	Mobile Station
MSC	Mobile Switching Center
MSISDN	Mobile Subscriber Integrated Services Digital Network
NIS	Network Information Service
NIST	National Institute of Standards and Technology
Node B	Base Station Transceiver
NONCE	Number used once
OTP	One-Time Password
OS	Operating system
PAM	Pluggable Authentication Module
PC	Personal Computer
PDA	Personal Digital Assistant
PIN	Personal Identification Number
PLMN	Public Land Mobile Network
RADIUS	Remote Authentication Dial In User Service
RAM	Random Access Memory
RAND	Random number
RES	Response
RNC	Radio Network Controller
ROM	Read only Memory
RFC	Request for comments
RPC	Remote Procedure Call
SAML	Security Assertion Markup Language
SAP	SIM access protocol
SASL	Simple Authentication and Security Layer
SGSN	Serving GPRS Support Node
SHA	Secure Hash Algorithm
SID	Security identifier
SIM	Subscriber Identity Module
SIP	Session Initiation Protocol
SMS	Short Messaging System
SP	Service Provider
SQL	Structured Query Language
SQN	Sequence Number
SSO	Single sign-on
SSP	Secure Simple Pairing
TGS	Ticket Granting Service
TGT	Ticket Granting Ticket
TLS	Transport Layer Security
TMSI	Temporary Mobile Subscriber Identity
TOTP	Time-based One-time Password Algorithm
UE	User Equipment
UI	User Interface
UID	Unique Numerical User Identity
UICC	Universal Integrated Circuit Card
UMTS	Universal Mobile Telecommunications System
USIM	Universal Subscription Identity Module
UTRAN	UMTS terrestrial radio access network.
VLR	Visitor Location Register
XML	Extended Markup language
XRES	Expected Response



# 1. Introduction

This chapter gives the overview of the Thesis. It includes the motivation, problem definition, objectives and how the thesis is organized.

## 1.1 Motivation

Every person in our society has one or several identities. A common person normally has a national identity, an employee or student identity, an alumni identity, a driver license and a set of other digital identities. People are constantly required to prove their identity. While in the real world proving your identity usually means showing a document that identifies you but in the digital world this process can be much more sophisticated. There are digital identity attributes like login, name, etc. that can be easily copied, now to prevent the identity theft some secret credentials known only to a person and an authentication authority have to be used [1] for secure identification of a person. Therefore Identity management is one of the important concepts of the modern society.

Identity management is a nonstop process that encompasses on identity lifecycle management, authentication, and access control [2] among other things. In the perspective of the modern enterprise information systems the main idea of the identity management is to manage access and control to enterprise resources and information. Enterprise information systems provide a range of services to support business processes. With the development of e-business the functionality of such systems grows along with increase in complexity as well, which makes it difficult to provide full protection from unauthorized access. The Authentication process plays an important role, as the decision whether to give access to services or provide resources is based on identity. Therefore, it is important to ensure that an attacker (fraudster) should not be able to steal the identity or masquerade as a user to a system in some other way.

The authentication party (user) presents a proof of identity to the authenticating authority for getting access to the system. The authentication schemes are mostly based on one or the combination of the following factors: something you know (password), something you have (ID card), and something that you are (fingerprints). One of the most common authentication schemes nowadays is a static password authentication. The password authentication schemes have been in use for a very long period of time. The reason for using static password scheme is that they are easy to use and people are used to them. Though, from the security point of view these schemes have many weaknesses.

The constantly increasing computational power of modern computers is making it possible to launch a brute force extensive search attack on password based authentication schemes if the passwords used are weak. As with study of cryptographic protocols and people (hackers) new vulnerabilities of security are coming in observation quite often, thus dramatically decrease the theoretical security level in cryptographic protocols which allows attackers to launch much more efficient attacks than the extensive search. So in strong designed cryptographic schemes the use of longer keys which have higher entropy provides higher security level. Therefore, it is required to use randomly generated passwords with adequate length that consist of a combination of letters, numbers, and special symbols to overcome this problem. Though, such passwords are difficult to remember for humans and this is one of the reasons that usually people choose passwords that are easy for them to learn but can be easily compromised with a simple dictionary

attack. Additionally, the user-name/password systems that the user has to use can be large in number which may result in reusing the credentials for all systems. It is also important to notice that by using strict administrative measures may not strengthen the security of a system as a too strict password policy (difficult to remember passwords or frequent change of password, etc.) can actually weaken it, since users can end up writing down passwords. Use of default passwords or careless users who reveal their passwords either accidentally or as a consequence of social engineering attacks can further decrease the security level of a system.

## **1.2 Problem definition**

There are systems where the ordinary static password based authentication may be sufficient as depending on the sensitivity of activity or data it has, but systems that processes sensitive data requires stronger authentication schemes e.g. Biometric systems, smart card based systems or one-time password based authentication schemes as these are considered to be much stronger than the ordinary user-name/password authentication scheme. However the cost of deployment and maintenance makes these systems less common.

In smart card development, security is a keystone to be considered. The blend of logical and physical security mechanisms makes a high level of security. The ability of storing information (e.g. identity information) and executing cryptographic protocols resulted in a huge success for smart cards to be used in security sensitive areas. The other benefit is that the Mobile network operators have already deployed smart cards to authenticate subscribers, hence having the infrastructure which can be reused to provide authentication in other services.

Thus a mobile phone authentication can provide strong authentication scheme based on the possession of the Universal Integrated Circuit Card (UICC) with a 3GPP application and based on the authentication of the card owner to the card. This solution which uses a mobile phone with a UICC card as a security token provides a much stronger scheme of security than the ordinary user-name/password authentication and at the same time decreases operational costs [3].

As currently there is no workstation logon architecture based on the mobile phones for Linux, thus we need to come up with the idea of using mobile phone based authentication scheme to integrate it in Linux operating system logon process. Thus, a study of related technologies and a detailed analysis of Linux logon process and its internal mechanisms responsible for authentication and logon are done in this thesis.

## **1.3 Objectives**

There is a lot of research done and several solutions have been proposed, using the mobile phone based authentication [3, 4, 5]. The main objective of the thesis is to study available mobile phone based authentication solutions and to come up with the most appropriate architecture that uses mobile phones for Linux workstation logon in an enterprise environment. As in my knowledge there is no available system or a published work that deals with the mobile phone based workstation logon using Linux operating system.

The other tasks that are required for the thesis are as following:

- Study of existing identity management and authentication for enterprise network and workstations.
- Study of Platform security architecture of Linux.

- Study of Authentication and identity management standards and protocols (e.g. EAP, RADIUS/DIAMETER, SAML etc.)
- Design and implementations of solutions for integration of the mobile phone with the workstation logon processes in different operating systems
- Threat and vulnerability analysis of proposed solution(s).

## 1.4 Organization of the thesis

The document has been organized in the following way:

Chapter	Description
Chapter 1: Introduction	This chapter gives the overview of the thesis and consists of motivation, problem definition, objectives and how the thesis document is organized.
Chapter 2: Background	This chapter gives the background study of the technologies required for the implementation of thesis. It starts with identity and identity management then identity management in enterprises and UMTS, SIM technology, Authentication and Authorization in different operating systems and at the end covering Bluetooth security and the Radius technology. Most of the work in this chapter has been covered in the project and the thesis is the continuation of work done in the project.
Chapter 3: Mobile Based Authentication Systems	This chapter presents the different architectures proposed for the mobile based authentication system. At the end a solution for mobile based authentication system for Linux workstation has been proposed.
Chapter 4: Analysis	In this chapter the functional requirements, non functional requirements, use cases and the activity diagram of the system is discussed.
Chapter 5: Design	In this chapter the component diagrams, package diagram and the class diagrams are discussed.
Chapter 6: Implementation	In this chapter the deployment diagram has been discussed along with the test cases of the system implemented.
Chapter 7: Security Evaluation	This chapter discusses the security aspect of the proposed architecture.
Chapter 8: Conclusion	This chapter discusses the results of the knowledge obtained from the thesis with the future work.

## 2. Background

To come up with the solution for the mobile based Authentication for Linux workstation logon process, a study of relevant technologies utilized in these solutions have been made. First the concept of identity has been explored; thereafter a study of different identity management systems in enterprises has been performed. As the mobile phone contains a UICC smart card, it is important to study how smart cards can be helpful to provide a secure storage and execution environment. Nowadays, different authentication schemes use Bluetooth to connect the mobile phone with the computer. Therefore, it is required to study whether Bluetooth can provide a sufficient level of security for using it in the workstation logon solution.

The study of identity management in UMTS is done as many mobile phone based authentication schemes uses GSM/UMTS secrets and identities and they rely on a mobile network encryption. The authentication and authorization in different operating systems is analyzed, as it was important for the thesis. At the end Remote Authentication Dial in User service (RADIUS) has been explored.

### 2.1 Identity and identity management

#### Identity

Every object around us has some characteristics which make easy for us to recognize them e.g. shape, weight, size, height etc. Identity can be defined as: *“The similarities and differences between objects that make them unique to identify.”*

The word identity is easily understood by everyone but at times it is difficult and confusing to understand when we consider the same characteristics of objects at different time and place.

Identity in Logic: It is defined as the relationship between a thing and itself i.e. it can be defined with a predicate “=” such if “ $x=y$ ” is true when  $x$  is the same thing as  $y$ . Identity is transitive (if we have  $A, B, C$  where  $A=B$  and  $B=C$  than  $A=C$ ), symmetric (For  $A$  and  $B$  if  $A=B$  then only  $B=A$ ), and reflexive (For every  $A$  is  $A$ ).

The **Identity of Indiscernibles** states that: *“No two distinct substances exactly resemble each other”*. This means that: “There does not exist two objects that can have exactly the same properties or characteristics”.

There are some laws that explain the above rule as if two entities are numerical identical then they must be qualitatively equal (same properties) and similarly if the two entities are qualitatively equal then they must be numerical identical. There are problems with these laws like the symmetric of the world, the infinity problem, the impact of quantum mechanics as discussed in [1].

#### Personal Identity

Identity of human beings is more complex. The personal identity is about the characteristics that make them unique and recognizable. Personal identity is also about ones knowing himself as well. There are certain issues that are relevant to personal identity are as follows:

- What should be the characteristics that are necessary to be a person?

- What are the conditions that are necessary to make sure that same person exists in different times?
- How a person can be recognized?
- How a person is unique from others?
- How people should perceive a person?
- Can a person change himself?
- Who is the model of a person whom he wish to be?

As there are some human characteristics that change with time like weight, height, so then how we can make sure that the same person exists in different times? This question can be answered with the following approaches:

- **The Psychological Approach:** It is about the psychological things like memories, beliefs. So according to this approach it is necessary to for a person to have psychological continuity to exist in different time.
- **The Somatic Approach:** This approach is about the physical relation. So identity of a person in different time consists in the identity of his body.
- **The Simple View Approach:** The approach says that the person in different can only exist in different times if they are identical.

### Citizen Identity

Governments throughout the world are introducing digitized personal identification and authentication systems into their service relationships with citizens [6].

This has improved the efficiency of and effectiveness of public service provision as discussed with case studies in [7].

The citizen identity depends on physical characteristics and needs physical continuity for identification. There can be infinite set of characteristics so we need to restrict to a few characteristics to make the process easy. The characteristic properties are called Attributes [1].

- An **attribute** is a characteristic attached with an individual. An attribute can be **intrinsic** (by nature) like finger prints, eye etc. or **extrinsic** (acquired from outside) like name, address etc. Attribute can be persistent or temporary.
- An **identifier** is an attribute that is most representative for an entity with in a context. E.g. a bank account and a person have identifiers. The person can be associated with account with extra information like social security number.
- **Personal Identifiers** are the unique attributes associated with a human being that are impossible to change like biometric attributes.
- **Biometrics** is systems used to recognize a person on physical attributes like fingerprints, iris, face, retinal, veins.
- An **identity** is a set of attributes that are permanent and associated with an individual which are unique and always make possible to recognize a person.
- **Identification** is a process of associating personal identification to an individual who have some attributes.

- **Authentication** is a process of proving the association between identifier and the individual.
- **Authorization** is a process of deciding to give access to which action on the basis of identifiers or attributes.
- An **identification document** or **identity card** is a credential used to verify the person identity. It may include name, age, profession, sex, rank, religion and new technologies are adding the biometrics as well. Passport is one of the example of identification document.
- The ID cards are expensive but one of the serious concerns is privacy with electronic ID. Electronic ID cards can be used to track movements of a person.
- **Digital Identity** is representing the identity of an individual in computers. Like we do online banking transactions the bank needs to confirm the identity of the user, so for that reason we need digital identity that may consist of name, password etc.

One of the problems that are arising is to have many login names and passwords that each user is getting. To solve the problem single sign on solutions are introduced to sign in once and have access based on user credentials to all processes without entering the credential again and again.

### **Identity Management**

Identity management is one of the hot areas for growth as it is expected to rise to more than USD 950 million by 2009 [2].

Identity management can be defined as: *“a set of data management systems and practices designed to increase confidence in the identity of individuals where appropriate”* [8].

Identity management can also be defined as: *“a discipline that consists of processes, policies and technologies to manage the complete life cycle of user identities across the system and to control the user access to the system resources by associating user rights and restrictions, these resources include information, services, process capability, buildings and physical asset”* [2].

The main benefits of Identity management systems are as following.

- It allows having control over user-to-application interaction which makes easy for auditing and reporting.
- It helps in lower operational costs.
- We have enhanced security of assets with the use of Identity management systems.
- We have good productivity. Users can access from outside the enterprise securely.
- Different companies can gave access to specific services securely without having to expose the whole system to outsiders.

There are some functions that are performed by an Identity management system are discussed as follows:

- **Identity Administration:** This function includes the management of the digital identities in the system which can be creation of digital identity (fingerprints, name, address etc) then maintaining it and termination if required.

- **Access control:** This functionality consists of providing the necessary access to the resources when required and is enforced by using access rights. It can be giving deletion, writing access to the user for some data.
- **Authorization:** This process shows what resources the user is allowed to access. The permission to access is granted by the authority.
- **User Self Service:** This functionality includes having such process to enable users in the administration of his identity like resetting password etc.
- **Auditing:** This is one of the important functionalities of Identity management systems. It includes analysis of all the process which helps in making the system secure e.g. it may include digital forensics for determining the access of and illegal user.
- **Identification:** This is a process of recognizing the users at the login.
- **Authentication:** This is the process to make sure that the person who claims to be user is the real one.
- **Single Sign-on:** There are certain processes that requires authentication at different stages and many times, so single sign-on helps to access all resources at once and can help in increasing productivity.
- **Federation:** The achievement of this process is to enable users of one domain to access the services, data of another domain securely which will help in removing redundancy and at low cost.
- **Reporting:** This process consists of reports that are generated by auditing. The report should be meaningful for the users that may be the owner.
- **Directory:** It stores the identities. It can include Light weight directory Access protocol (LDAP), relational databases, flat files and other types of data stores.
- **Meta Directory:** It provides the organizational view of the information in the heterogeneous directories and other storages.

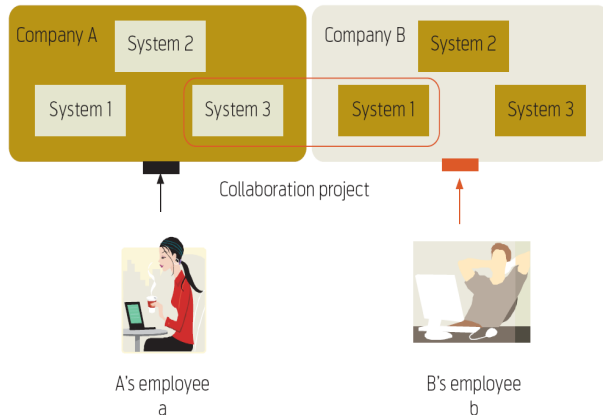
### **Enterprise Identity Management**

The organizations usually have different departments. Users in each department are given identity like user name and password which is valid for that department. If the user is given some task in another department for some time, he has to be registered in that department which can cause serious problems if he is not deleted from there than and difficult for user as well to remember many passwords and usernames. So to solve this problem it is recommend having a centralized Identity management system and the user is granted access to the resources according to the company policies, thus reducing cost, time and making it more secure.

### **Partnership Identity Management**

Nowadays many companies have to make collaborations with other companies and terminate them as soon as possible for survival. Collaboration between companies requires sharing of data and resources, so it would be better to have control over the access given to the users of other company to avoid any serious damage. So it would be to have an identity management system

that allows the flexibility of adding the users of other company and easily assigning the required services securely without damaging the ongoing system. Figure 1 explains that system 1 can access the data of system of another company.



**Figure 1: Partnership Identity Management [2]**

### **Customer Relation Identity Management**

In this kind of Identity management it is encouraged to give the management task to the user as maximum as possible, which will help in reducing management cost. It needs careful evaluation that which system the user is allowed to access and which not to for security reasons.

### **Identity Management as Commercial Service**

Identity management is a complex process which requires resources and knowledge. Companies don't have that much resources so they go for outsource the identity management.

Three parties are involved in this process. One is an **Identity provider** that manages the identities of the users. They issue and validate the identity credentials.

Second is the **Service provider** that provides the services for the end user. They are the relying party. They provide the services to the user on the identity credentials validated by the identity provider.

Third is the **User** who is using the services and proves his identity by his credential like user name and password.

So for a company who provides services have not to worry about Identity management. He works as a relying party and the Identity management is provided by the Identity provider, reducing his management cost.

## **2.2 Identity management in enterprise systems**

To provide a range of services for supporting business processes, the enterprise information systems are made. The Business objectives are fulfilled by the development of e-business, enterprise IT systems and that is the reason behind the growth of functionality provided by IT systems. Yet, the complexity and vast functionality of these systems makes it harder to provide



protection against the unauthorized access to the enterprise information. The Access control management is required for outsiders, semi-trusted parties like partners, customers and insiders. For managing the access to the enterprise resources, access control system strongly depends on the identity management system. Identity and access control management (IAM) has to meet the requirement of business to have external links with partners, suppliers, customers and other people. Thus, managing the identities of corporate users is not enough; identity management goes beyond the boundaries of single enterprise.

The other fact that makes it more complicated is that many organizations have heterogeneous IT platforms, and each has its own identity database. The centralization of identity management will simplify the management tasks, will reduce the management costs, and decreases the risk that something will not be taken into account during data flows planning and privileges setup [10].

Additionally, IAM systems must fulfill the government legislation in order to provide specific services like processing credit card payments [10].

### **Centralized identity and access control management**

With Centralized identity and access control management approach the identities and access controls for many heterogeneous systems with different repositories in an enterprise can be managed with a single interface. A directory service forms the basis of an identity and access control management system as it stores the information about the state of an enterprise system. This state information consists of identities, account-related information, policies, roles, groups, workflows etc. A directory is a unique type of database which is optimized for read operation, whereas the other databases are optimized for the write/modify operation [11]. In addition, directories are mainly accessed through Lightweight Directory Access Protocol (LDAP) and databases utilize Structured Query Language (SQL) for this purpose [11]. If we use only one server that stores all data in one location, then we can say that the directory is centralized. The directory can be distributed as well, if there is more than one server and information has been replicated between servers in order to have all the same data or it is divided between the servers so that each can hold certain part of the data.

It is difficult to manage different repositories of identities and access control rules which can also cause mistakes that can lead to weakness in security. Synchronization and consolidation of data will decrease the management costs and will make the whole system much more easily manageable. There are different methods which can provide an integrated enterprise directory service. The options can be a single directory, a meta directory or a virtual directory [10].

**Single directory:** In this method there is only one directory of identity information for the whole system. A single directory will simplify the management tasks as compared to a bunch of different repositories; however there have to be some modification in some applications to be able to work with a single directory [10]. Uniqueness is not in physical terms but in logical. In order to not create a single point of failure and share the load a single directory can be distributed in to several servers having the same data. Legal, political or security issues may create a barrier preventing the creation of a single directory [12]. In this case a single interface will be provided by the IAM to manage all these systems.

**Meta directory:** In this method all the information from different repositories is copied into a single directory with a unified namespace. The synchronization between the meta directory and satellite/original repositories is controlled through bidirectional synchronization mechanism if

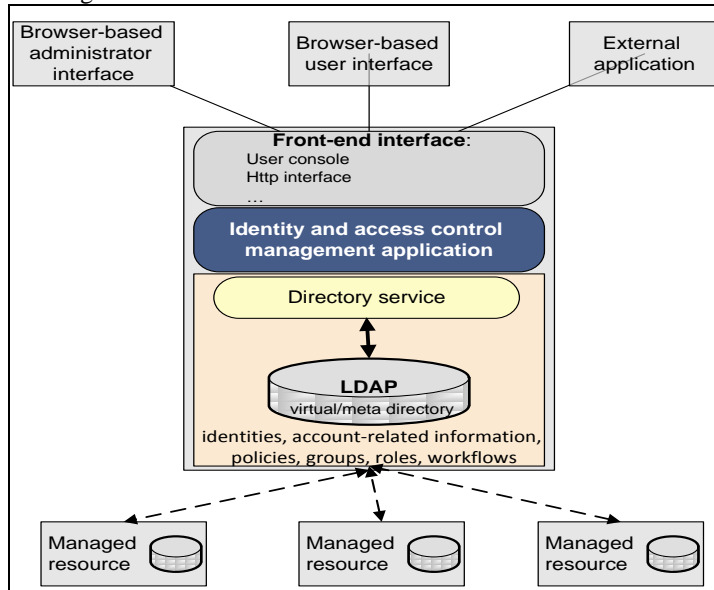
there is some change occurred in the data [12]. With using this approach the requirement of modification of applications will not be needed, that will work with the particular repositories as these repositories will remain in the system [10].

**Virtual directory:** In this method the virtual directory will serve as an abstraction layer between various repositories and applications by providing a single logical directory which will gather the information from all the repositories in real-time. The logical presentation of data can be modified for each application. By using the same input data different application-specific views of data, optimized for application needs can be derived [13]. There is no central physical directory which will contain a copy of all data.

The main functions of IAM are:

- One of the main tasks is managing users and accounts which include creating, editing, deactivating, and deleting users, setting/changing user passwords, etc.
- The other task is policy and workflow based management which helps to automate management process. The Policies which can be used in an enterprise consists of account provisioning, password, authentication policy, etc. The workflow can be defined as a predefined sequence of automated processes that automates some time-consuming actions like gathering approvals. Workflows also enforce consistency (it is the sequence and the set of involved actions to be constant) and completeness (it is to not start some action until all previous are completed successfully). The Request for approvals always takes the same predefined path and is automatically delivered to the person in charge.
- The functions of IAM include Privileges and Access control management for setting permissions to make sure that only those entities that have permission can access the resources. The common models used for the access control are Discretionary Access Control (DAC), Mandatory Access Control (MAC) and Role-based access control model.
- The IAM functions include Enterprise single sign-on mechanism that enables a user to authenticate once to a system and can allow user to reuse this authentication for many enterprise applications during the current session. In this case the requirement to have different accounts for different service providers and remember corresponding passwords is not needed. A user gets enterprise wide access to data at once.
- The IAM also provides strong authentication which imply the usage of a two factor authentication. The Technologies such as smartcards and biometrics can be used to provide two-factor authentication. According to [12] SIM-based strong authentication that uses mobile phones can be quite cost-efficient as compared to the other two-factor authentication solutions, and it is comparable in price with password-based authentication.
- The IAM provides Monitoring and audit services in which system events are automatically checked against policies and rules for violations. If the violation is detected then some action is triggered like account blocking.
- The IAM provides Federated identity management in which user authenticated identity information is communicated across security domains to trusted partners that exist in the same Circle of trust. Thus, a user is not required to authenticate when he accesses the resources of a collaborating company.

Identity and access manager architecture:



**Figure 2:** Identity and access manager architecture

A standard administration and management toolkit for a single directory can be used without using IAM software. But all the functions that are automated in an IAM suite have to be done manually. IAM suites can provide a benefit that they can define policies that automatically create user accounts, mail boxes, and group memberships in real-time. Moreover, creating a single directory is not always possible.

### Centralized authentication and trust

At the early phase of the development of authentication solutions, the use of decentralized autonomous approach was common. In a system every station performs the authentication and authorization autonomously by maintaining its own user's file or user's database, and the access to the services and resources was provided to that station depends on that file. To allow users to use services provided by some server the administrator had to add information about those users to the users file of the server. A user is required to have an account on every station for the services and resources he uses. This kind of management approach had serious scalability issues. In case where different passwords were used for different accounts so the remembering of all passwords and the synchronizing password changes between many accounts in case one password was used for many accounts makes both these approaches a big problem in management and security. All identity and access control management procedures requires a lot of work and were prone to errors.

The solution for the problem was to use a trusted third-party authentication approach. In this method the third-party authentication authority maintains a centralized repository with identities and account relevant information. Each user is given only one account which is maintained by the third-party authority. The authentication is responsibility of trusted third party and is solely

done by it. The rest of the systems trust it. The third-party should be highly secure thus can be trusted; otherwise the compromise of the third-party would lead to the compromise of a whole system. The third-party authority can be a single point of failure; if it crashes the whole system will stop working. There are two third-party authentication based schemas [14]:

- Implicit authentication schema: In this schema an authenticating entity (ex. Service provider) does not request authentication service from a centralized authentication authority. The authentication is cryptographically produced from the encrypted message given by the third-party to the entity that is being authenticated. E.g. Kerberos v5 protocol uses this kind of approach.
- Explicit authentication schema: In this schema an authenticating entity explicitly requests third-party to make authentication.

There are many technologies which can provide authentication and access control, but among all the preferred are Kerberos security service and LDAP directories [15]. Kerberos prime task is to provide authentication but it can also provide some simple authorization services. Alternatively, LDAP directories are primarily used for storing and managing authorization data, but they can also provide some authentication services [15] as well. And this is the reason that these technologies can be used separately or can be integrated in one system.

### **Kerberos protocol**

The Kerberos protocol deploys implicit authentication schema. Kerberos version 5 which is defined in RFC 4120 is the present version of the protocol. Kerberos performs mutual secure authentication in the network which is not secure, however it does not provide accounting and offers very basic authorization [16]. To provide secure authentication service Kerberos uses secret key cryptography. The three types of entities in the Kerberos architecture are as following:

- Clients: They are the one that wants to use the services provided by the service providers.
- Service providers: They provide the services for clients to use.
- Kerberos servers: They manage Kerberos authentication. They are known as Key Distribution Centers (KDC). A secret key is shared by the KDC with every principal (a client or a service provider) in a network. The service providers and clients trust KDC. KDC is composed of three different elements. Authentication server that serves the client authentication requests. The Second element is the Ticket granting server whose job is to issue Ticket Granting Service (TGS) tickets to clients, and the third element is the database that stores identities, secret keys, policies, etc.

The security in the Kerberos schema is based on tickets and the corresponding authenticators. A ticket is an encrypted message that contains the client name, session key and the life time of the ticket. The encryption of the ticket is done with a secret key shared by a server and KDC. The authenticator consists of an encrypted client's name, client's realm, timestamp. The encryption of authenticator is done with the session key that is in the corresponding ticket. A client can authenticate by providing the ticket and the authenticator to a server.

The KDC is trusted by all principals in a realm which is an authentication domain with some administrative boundaries. The management of a realm is done by only one KDC (to share the load a KDC can be distributed between several servers, but logically it is one). Each realm has its

own KDC database. The principal identity is formed with the Name and the Realm parts.

### Kerberos authentication model

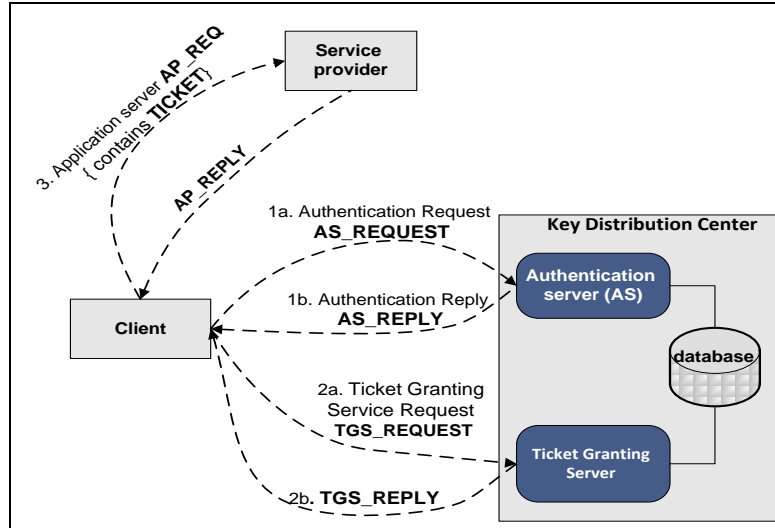


Figure 3: Kerberos authentication model

The Kerberos authentication procedure consists of three distinct exchanges which are discussed as following:

- **The Client & Authentication server exchange:** The Client authenticates the KDC and obtains a Ticket Granting Ticket (TGT) which will be used to get the credentials for authenticating to a Service provider. The client can be authenticated by the Authentication server before issuing the TGT (pre-authentication), or it can wait until the Client – Ticket granting server exchange [15]. From the security prospective it is better to make pre-authentication, because this will provide a protection against the attacks on the principal's secret key [17] that is used by KDC to encrypt the response.
- **The Client & Ticket granting server exchange:** The Server authenticates the Client (this authentication is done in every case) and grants a Ticket Granting Service (TGS) to the Service provider which is specified in the request by the client. The TGS granted will be used in the Client – Service provider authentication.
- **The Client & Service provider authentication exchange:** In this procedure the Client sends a request to the Service provider which normally contains the authentication information and the initial request [17]. The Client at all times authenticates to the Service provider, but the client must requests for mutual authentication explicitly [17]. The TGS and the authenticator that are in the Client's request facilitate the Service provider to authenticate the Client and obtain a shared session key which can be used for further protection of communication with the client [17]. If mutual authentication is requested by the Client then the reply of the Service provider contains the authentication information that enables the Client to authenticate the Service provider.

The main services that are provided by the Kerberos are as following:

- **Mutual Client & Server authentication:** In this process a client is able to authenticate both KDC and various Service providers which will help by not allowing a malicious server to deceive a client and obtain confidential information.
- **Centralized management of authentication information:** The Kerberos uses a centralized database for storing identity and account information.
- **Delegation:** If a principal wants to allow a service provider to perform operations on its behalf e.g. a client can assign rights to a printing server to access client's files on a file server to print them out [17]. A principal can ask the KDC for a new TGT with different network address that will allow a service with that network address can act on behalf of the principal. The principal is required to transmit the new TGT and the corresponding session key to the service provider to facilitate delegation.
- **Single sign-on:** The Kerberos supports single sign-on with caching tickets and the corresponding session keys. Thus the next time the authentication is needed, a user is not required to type in the password again. The cached credentials are used on user's behalf.
- **Cross-realm authentication:** In this process a client in one realm can authenticate to a service provider from another realm. For establishing an inter-realm communication the service provider's KDC should be registered as a principal in the client's KDC. After that the client asks its KDC for a ticket to the service provider's KDC. It presents this ticket to the service provider's KDC and requests for a ticket to the service provider. It is possible to pass through several realms to authenticate to the remote service provider. The Inter-realm communication can be controlled hierarchically. So the authentication path through multiple realms can be easily created [17].
- **Multi-factor authentication:** the traditional Kerberos authentication is done with the secret keys (for a user the secret key is derived from a password). Though, the public key cryptography for initial client – KDC authentication (PKINIT which is defined in RFC 4556) can also be used which makes it possible to use smart cards and other cryptographic tokens when authenticating through Kerberos [18]. PKINIT needs the usage of trusted by KDC and its principals Certification Authority. We can use One-Time Password mechanism and biometric scanners for the initial authentication.

There exist several different implementations of the Kerberos Protocol e.g. MIT version, Heimdal version, Windows Active Directory version etc [19]. Active Directory does not simply use Kerberos as its default authentication protocol, leaving the NTLM for compatibility [20]. But it tightly integrates it in its framework which results in some issues in environments with Windows and non-windows systems [18].

### **Explicit authentication schema**

The authentication in this schema is entrusted to a third-party that has all the identity information. The typical authentication procedure is discussed as following:

- a client request a service from the application server and it forwards its credentials (identity and password) to the application server
- the application server requests the trusted third-party server to carry out the authentication

It is required to cryptographically protect the exchanges between the client and the application server and between the application server and the third party.

To provide a centralized third-party directory service for storing identities and account-related information, the technologies like Network Information Service (NIS), NIS+, and directory services with Lightweight Directory Access Protocol (LDAP) interface are used. Normally the NIS is used to centralize the storage of */etc/passwd*, */etc/shadow*, */etc/group* files from all stations in the Unix-based domain, and after that this information can be accessed by clients [21]. The changes in the centralized directory are also propagated to all source stations. From administration point of view the NIS is easy to manage, but it lacks the security mechanisms. It does not provide authentication and authorization for directory access and the communication is unencrypted [16].

Both NIS and NIS+ protocols are based on Remote Procedure Calls (RPC). NIS+ is advance version of the NIS protocol. Besides providing hierarchical namespace, NIS+ offers a stronger security mechanism. Though NIS and NIS+ are legacy technologies and it is recommended to transfer to LDAP [22].

### **Authentication against LDAP server**

The Lightweight Directory Access Protocol (LDAP), defined in RFC 4511, is a protocol that accesses the directory services which comply with X.500 standard. There are various directory servers that provide LDAP support e.g. Active Directory, IBM Tivoli Directory Server, OpenLDAP, Novell eDirectory, Sun Java System Directory Server etc.

LDAP is a client-server protocol which runs on the top of TCP. It is recommended that the servers use port 389 for an incoming request. A client requests to a server for performing some operation in the directory. The server responds to the request, performs the operation and returns a response. Some of the operations used in LDAP are bind, unbind, add, delete, search, modify, compare and startTLS.

The Bind operation is the same like authentication. The Bind request specifies the authentication identity. The Bind operation uses different authentication methods that are simple authentication method and SASL authentication method [23]. The simple authentication method is further decomposed into the following methods that are anonymous authentication, unauthenticated authentication, and name/password authentication. In the name/password authentication, for validation the client sends both the name and the password to the server. This method of authentication should only be used in environments where confidentiality protection is provided [23]. A TLS can be established for the LDAP session by sending a request with StartTLS operation by the client. The TLS will provide confidentiality and integrity protection for LDAP session, so a simple name/password authentication can be performed in a secure manner.

The Simple Authentication and Security Layer (SASL) defined in RFC 4422, is a framework which enables the use of various security mechanisms in protocols. SASL provides the abstraction layer which allows any protocol to use any mechanism. By using SASL

authentication method LDAP allows authentication through any SASL mechanism [23].

An indication of a success/failure of the authentication request is communicated by the Bind response message. LDAP has become one of the key elements in enterprise identity and access control systems [19]. It can provide a centralized storage for identity and account-related information and can be utilized to authenticate principals. The procedure of authentication starts by the client sending its identity and password to the application server over the protected channel. E.g. protection can be provided by TLS/SSL. If the LDAP server is configured in such a way that the principals need to authenticate to it, in that case the application server sends the Bind request to the LDAP server using the credentials of the client. If this authentication succeeds (in that case the LDAP server sends a Bind response message with success status), then the client is considered to be authenticated by the application server. The other simple solution for the application server is to simply retrieve the client's identity and password from the directory and compare it with those received from the client [24]. For using this solution it is required that the application server should authenticate to the LDAP server prior to the information retrieval and the application server should be approved by the LDAP server to perform those actions.

### **Enterprise Single sign-on**

The mechanism of Single sign-on (SSO) enables a user to authenticate once to a system and can reuse this authentication during the current session. The technologies that can use to provide a single sign-on can be divided into three main categories [25] that are ticket-based, cookie-based, and PKI-based.

In the Ticket-based SSO the user authenticates to the authentication service and in response receives a cryptographic ticket. This ticket can later be used to authenticate to service providers. Kerberos is a one of the examples of ticket-based SSO system.

In Web-based environments the Enterprise SSO can be achieved through cryptographically protected HTTP cookies. The Sun OpenSSO Enterprise 8.0 uses this method for providing a SSO solution. The process used by OpenSSO involves the following main steps [26]:

- The user request to a service provider by sending a HTTP request. The policy engine intercepts the request that protects the resource. After exploring the request and if it does not find an HTTP cookie then the policy engine redirects the user to another URL for authentication.
- The browser follows the URL and sends a new HTTP request to OpenSSO Enterprise authentication service. The authentication service using one of its authentication modules, e.g. LDAP authentication, validates the user's credentials. The HTTP response which contains a cookie that carries an encrypted session token is sent to the client. The HTTP response redirects it to the original location.
- Again the browser sends an HTTP request to the service provider for one more time. As this time the request contains the cookie with the session token. The policy engine receives the request and it checks it for the session token. The check is done by contacting the OpenSSO Enterprise service. OpenSSO Enterprise decrypts the token and it inquires whether the session data related with the session token exists. The policy engine receives a response defining whether the token is valid. After the



validation of session token the policy agent makes the decision whether the user should be granted access or not.

- When the user contacts some other service provider next time, the cookie with the session token is incorporated in the request. Thus the policy engine that receives the request needs only to certify the token. So the second authentication procedure is not required from a user in this case.

The Public key based SSO needs to use the public key infrastructure. The trusted third party certification authority is in charge for examination of user's credentials and issuing certificates. The certificate and the user's private key can be stored on a user's computer or on some cryptographic token [27]. The user is authenticated by the service provider itself, the certification authority checks the identity of the user only while issuing certificate. The PKI infrastructure also provides the service of non-repudiation which is considered quite important for business [25].

### **Identity federation**

Now days the business-to-business communication is carried out by the extensive use of internet technologies. The Business processes in a company require external connections with partners, suppliers, contractors, clients, etc. Though, creating and managing the accounts for external users locally is not an efficient solution in the perspective of security, management and operational cost [28]. Identity federation enables the inter-organization identity and management sharing and secures the external access to the required set of company's resources.

Identity federation depends on the trust relationship between collaborating organizations. So, one organization trusts in authentication made by another organization for users. The trust relationship is the basic requirement that makes the Single Sign-On service for cooperating organizations possible. The organization that is responsible for maintaining and managing the user's identity information is called the identity service provider. The functionality of service provider is to provide some service. A circle of trust is formed by at least one identity provider and a group of service providers that trust in this identity provider. There are many organizations that play the role of service provider and identity provider at the same time [4], but for some companies it is favorable to outsource identity management to an identity provider as this can help in focusing on the services they provide rather than the identity management.

Identity Federation enables flawless interaction between different organizations with completely different and independent environments. It is not required for Collaborating organizations to have similar security systems or have in depth knowledge of systems used by the partner [12]. There are many frameworks which define federated identity management; some of the examples are Liberty Alliance Identity Federation Framework (ID-FF), Web Services Federation (WS-Federation) and Security Assertion Markup Language (SAML) framework. The SAML v2.0 is considered as a favorite solution [26] as the Liberty ID-FF and SAML v1.x were contributed to OASIS consortium and are the foundation of SAML v.2.0. WS-Federation is an alternative solution for interacting with the Active Directory Federation Services [26].

SAML [29] is an XML-based framework which is used for exchanging identity, corresponding attributes and authentication information between collaborating organizations. The Information exchanged is expressed in the form of SAML assertions. Assertion consists of a set of statements about a principal (which can be a user, computer or company etc.). The types of statements that a

SAML assertion can contain are as following:

- **Authentication statements:** It is created by the party that successfully authenticates a user. It describes the authentication mechanism and the time of authentication, etc.
- **Attribute statements:** It contains a specific principal's attributes like e-mail, telephone number etc.
- **Authorization decision statements:** It describes the authorized actions

The SAML assertions are sent in a SAML protocol messages (both assertions and protocol messages consist of XML documents). The SAML bindings explain how protocol messages are carried by the underlying transport protocols. The SAML defines SOAP-based bindings and HTTP-based bindings. The security protection for message exchange is not provided in SAML protocol [30]. It depends on other protocols like TLS/SSL or IPsec for providing the functionality of confidentiality and integrity protection. The security can also be provided by XML encryption and digital signatures.

The foremost use cases of SAML are Single Sign-on and Federated identity. A common SSO scenario is when the client uses a browser application and sends a request to a service provider. For authentication the service provider redirects the user to the identity provider. An assertion is issued to the user after the authentication by the identity provider which will be used by the service provider to authenticate the user for granting access to the resource. Any requests from the same user in the same session for other service providers in the same circle of trust do not require the user to repeat the authentication procedure again.

A federated identity use case is a user is required to have a federated identity and when the collaborating organizations agreed on how user will be referred [29]. This shows that the organizations have the same meaning of the identity of the user which is referred in the message exchange. When the identity is federated then the information about that specific user can be shared between different organizations.

There are many factors on which an Identity federation model depend e.g. some of the factors that influence identity federation are like whether the exchange of identity attributes about users should be allowed, whether the users have existing local identities, whether the temporal identifiers that are used for identity federation should be destroyed after session termination etc.

Some of the use cases defined by the SAML for identity federation [29] are as following:

- **Federation through Out-of-Band Account Linking:** In this case the identity federation is established without the use of SAML protocol e.g. it could be done through the database synchronization.
- **Federation through Persistent Pseudonym Identifiers:** In this case the permanent SAML pseudonym identifier is used for dynamically establishment of identity federation during the web SSO exchange
- **Federation through Transient Pseudonym Identifiers:** In this case a temporary identifier is used to temporary federate identity till the termination of user's web session. The advantage of using this approach is that an organization is not required to manage the local accounts of users from a collaborating organization.
- **Federation Termination:** the elimination of an existing federation.

The use of Identity federation enables the organizations to provide access to external users from the collaborating organizations without managing these accounts locally. Thus, simplify the identity management, reduce the administrative costs and enhance the security. In addition it also provides the advantage for external users in presenting a web single sign-on service. SAML is the standard solution which supports both the identity federation and single sign-on.

One of the important things for business environment is Identity and access management. In this mechanism the enterprise resources are protected from an unauthorized access by the inside and external users. Business objectives require the extreme usage of internet-based applications and a limited and secure access to the enterprise resources for external users in the collaborating organizations which can be partners, contractors or clients etc. The identity and access control management tasks are complicated by the variety and complexity of systems used by the business. The use of centralized storages for identity and account-related information and centralized authentication makes the things more manageable. But the possibility of always having a single directory is less because of the administrative, security and some other issues. For this purpose many identity and access management suites are developed to provide a centralized interface for management.

## 2.3 Smart card technology

Smart card technology is considered one of the important technologies for the modern information systems. There is huge deployment of smartcards because of the widespread use of smart cards in mobile phone networks, international payment systems e.g. MasterCard and Visa and transport and ticketing systems. Other than this the smart cards are also widely utilized in identification and access-control systems. The main reason for this success of smart cards is in the security services provided by it.

### Smart card definition and types

The definition of a smart card can be as a card that's size is about of a credit card and it contains an embedded integrated circuits. We are not considering the magnetic stripe cards which can be used only for information storing as they do not provide sufficient level of security and can be easily forged [31].

There are different types of smart cards with different functionality and purpose which are discussed as following

**Memory chip card:** The purpose of this type of cards is to store the information. They normally do not have on-board processing facilities. There are almost no security gains of Memory chip cards when compared to the magnetic stripe card [31, 32]. Although there is benefit of memory chip card of containing one-time-programmable memory which can be written once and cannot be rewritten later [33] as compared to the magnetic stripe card, but it is still easy to read the stored value and make the copy of the cards [31, 33].

There wired logic-integrated smart cards are more sophisticated memory cards that provides write/erase protection and a restriction on read access as well [34]. They have few predetermined extra functionalities and redesigning of the chip is the only way to alter the available small command set [32]. Even though the arithmetic logic unit is very limited but still it is able to perform simple cryptographic operations for the authentication and data encryption. E.g. the authentication of the reader based on the stored keys with encryption of all subsequent memory

operations is provided by MIFARE classic [35].

**Microprocessor chip cards:** It contains a microprocessor, an operating system, different types of memory and I/O circuits in this type of smart cards. For acceleration of execution of cryptographic operations, the smart cards may optionally contain a crypto coprocessor. The types of memory cards that Smart cards contain are Read Only Memory (ROM), Random Access Memory (RAM), and Electrically Erasable Programmable Read Only Memory (EEPROM). ROM contains the data which is stored during the manufacturing process and cannot be modified; the only operation allowed is read during the card operation. In ROM the operating system of a smart card is stored. RAM is a volatile type of memory as it is used as dynamic data storage and can lose its content on power shutdown. EEPROM is a non-volatile memory, which means that the data is still saved when power is off. EEPROM is used for the data and application program codes. The major problems of EEPROM memory is that it has limited number of write cycles, even though it can be read for unlimited amount of times. There is a slow gaining popularity of other memory types like flash memory with shorter write access time and longer lifetime as compared to EEPROM in smart cards [31].

There are two different chip card interfaces of the Smart cards which are contact and contactless. The Contact and contactless smart cards are standardized in ISO/IEC 7816 and ISO/IEC 14443 standards respectively. The major difference between the contact and contactless interfaces is that the contactless reader produces energizing radio frequency field for energy transfer to the contactless smart card through air and the modulation of this field enables transfer of data. Therefore, embedded antenna is available in contactless smart cards. There are some cards that have dual-interface which means that it have both the contact and contactless interfaces.

As defined in ISO/IEC 14443 the operational frequency for contactless operation is 13,56 MHz. The Contactless smart cards can be further divided into proximity and vicinity smart cards. There exists a limitation with Proximity smart cards that they must be in a close proximity (up to 10 cm) from a reader to function properly. The vicinity cards standard is described in ISO/IEC 15693. The vicinity cards operational range is up to approximately 1 m [33]. As according to the ISO/IEC 14443 the data rates supported for contactless smart cards are 106, 212, 424, and 848 Kbit/s [31]. The vicinity cards provide greater operational distance but the data transfer rate is lower than as for proximity card which is only 26.48 kbps [31].

The wireless interface introduces a honey spot for attackers to attack on smart cards which include but not limited to attacks like eavesdropping, denial of service and man in the middle. Though according to [34] contact and contactless cards provide the same level of security if the threats particular to contactless interface are taken into account in the security architecture of a smart card.

### **Security provided by smart cards**

One of the basic functions that a smart card should provide is secure storage for data [36]. Thus, a non-amendable memory is of great importance to a smart card security as it can be used for storage of system secret keys [33] (typically top keys of the key hierarchy). The other approach that can be used instead of storing the system secret key in ROM is to compute it on the basis of a unique chip serial number that is stored in the ROM [31, 33]. The secure microprocessor is the important factor of the smart card security system. The word “secure” is used where we mean that the microprocessor is protected against physical and side-channel attacks. One of the main

reasons of introducing the microprocessors in smart cards is of security reasons as the microprocessors made the cryptographically protected communications possible [34]. From the security perspective the main functions of the microprocessor is to generate a pseudo-random numbers for crypto protocols, generating the temporal keys, digital signature generation, encryption/decryption and performing the operating system security checks like checking whether the access to smart card resources should be granted. As a result along with providing the secure storage, the smart cards can also be used for secure execution of cryptographic algorithms [33].

The access to microprocessor smart card's resources is managed by an operating system which is run on a microprocessor. Therefore the logical level security is provided by the operating system access control system that is responsible for granting or denying the access to smart card resources. The modern smart card operating systems provide the following security mechanisms:

- **Access control:** In this mechanism smart card resource is accessed on the basis of specific access conditions and only allows authorizing the entities. The Access control can also be based on either a state-oriented or a command-oriented access model [33]
- **Authentication:** This mechanism is used for card reader authentication, user authentication (so before performing any operation the user should be authenticated) and authentication of communicating parties
- **Process isolation:** This mechanism is required so a process can't access resources owned by other processes
- **Atomic transactions:** This mechanism is required so either all of the operations that make up a transaction are performed or none of them
- **Secured communication:** It is required so that smart cards should use a secure protocol for communications as to protect the information during transfer.
- **Cryptographic protection:** The smart cards support the use of various cryptographic mechanisms e.g. encryption, hashing, digital signature, random number generation for security of data.
- **Key management:** In smart card the operating system is in charge for secure generation, distribution, usage and destruction of cryptographic keys
- **Security monitoring and audit:** In this process the system events can be monitored and analyzed for any potential security violations
- **Secure data deletion:** It is required so that the data after deletion should not be accessible or recoverable
- **Card locking:** This is required so to have an ability to either temporary or permanently disable a specific application or the whole card

On the other hand an attacker can directly attack the memory or buses during data transfer to get the data and avoiding the security checks of the operating system. Therefore it is required to handle this threat; therefore, smart cards also provide protection against physical attacks.

The Physical attacks on smart cards consist of two types which are invasive and non-invasive attacks. In the invasive physical attacks case it is required to remove the microprocessor from a card so that to get direct access to it. Sophisticated equipment is required in this kind of attacks (e.g. a microscope, laser, micromanipulators, focused ion beams) and in-depth technical knowledge [31, 33]. The analysis of the microprocessor structure is done first for starting the

Invasive attacks. That is why it is required that the first line of protection measures is deployed at this stage so that it can complicate the analysis for an attacker. The steps that can be followed in protection measures are to make a small size of IC components which will make it hard to extract information [33, 34]; make dummy structures on the chip which have no meaning to the functions of the chip but used for misguiding [33], the layout of the chip's blocks can be random [31], make the chip cover with a metal layer which will hide the layout of the chip [31]. Even though the hiding of the structure with cheap material will make the analysis difficult for an attacker, still it should not be considered as a suitable protection against the physical attacks. The mechanisms discussed below provide stronger protection specifically against intrusive attacks:

- **Buried and scrambled chip buses:** In this mechanism the buses are buried inside the chip to prevent direct contact. Other than this the buses are jumbled to complicate the understanding what these buses are responsible for [33, 34].
- **Current-carrying metal layer on top of the chip:** In this mechanism the whole chip is shielded with a metal layer on the top which can be current-carrying. In addition to the protection from internal structure analysis the metal layer also protects against the attacks that use electrical measurements from the chip's surface [33]. As a result, if the shield of metal is removed then the chip will not be operational.
- **Memory encryption:** In this case the volatile and non-volatile memory and some microprocessor registers are encrypted with some specific keys [33, 34]. But in this case if the data is read from memory still it will be required to be decrypted.
- **Buses encryption:** In this the data in buses during transfer is encrypted [34]
- **Anomaly sensors:** In this case the sensors are used to detect the abnormal environmental conditions e.g. Voltage monitoring, temperature monitoring and external clock monitoring are examples of available sensing techniques [31, 33, 34]. It is important to notice that the clocking is provided by an external source, and as a result the processing speed of a smart card is also controlled by this external clocking source. It will be quite useful for an attacker to lower the frequency significantly as it will make the measurements easier [33]. Therefore to defend against this attack it is required to implement the clocking sensor.

One of the examples of non-invasive attacks is side-channel attacks. For side-channel attack a monitoring of device during its normal operation is carried out in order to obtain some information. Some of the factors that can reveal information are timing information, power consumption levels, electromagnetic leaks that are correlated with execution of some commands/computations on secret data. In timing attack execution time of particular operations is measured to get some required information about the secret data. In power analysis the information about which operation and with what parameters is being executed is leaked by monitoring the power consumption levels. One of the powerful technique is Differential power analysis (DPA) as the statistical analysis is applied to the power measurements done while repeatedly processing the known data and then the unknown data to observe the difference in the power consumption for different input data [31, 22]. Through this way the unknown data can be exposed either partially or fully. The principle in electromagnetic analysis is the same as discussed but it deals with the electromagnetic radiations from a chip.

To protect against the timing attacks we need to use the constant execution time algorithms which consume the same execution time for different input values. But the microprocessors that have constant power consumption because of hardware tweaks are very expensive and are not

realistic [37]. Other than that we can use Random delays in an algorithm and other more sophisticated masking techniques to decrease/destroy correlation between measured parameters and secret data [31, 37]. Though, according to [37] most of these protection measures are fragile to differential fault analysis.

Injection of faults to disturb the operation of a microprocessor is the basis for the Differential fault analysis. Some of the factors are abnormal voltage, temperate, clocking and electromagnetic influence that can lead to skipped commands, misinterpreted commands and data read with errors [31]. The countermeasures [31] for such attack can be using checksums, having a variable redundancy when a variable has a copy that is modified along with the original and then they are compared, and execution of the same commands several times and comparison of results etc.

As Smart cards are complex devices so the security is achieved by the blend of logical and physical protection measures. A weakness in one of these two layers can partially or completely compromise the smart card. Thus it is required to careful design the software and for smart card security.

It is necessary to test the claimed security functionality of smart cards against some international standard against the claimed security by the company. It is necessary to have an international security evaluation standard that assures the security functionality of a product against some requirements. In that case the evaluation is done by independent certified laboratories. The depth and the scope of the evaluation concludes the evaluation assurance level (EAL) issued by the certifying organization. Thus EAL is the level of confidence that the security functionality of a product is the same as required [38]. The EAL is divided into seven assurance levels ranging from EAL1: “functionally tested” to EAL7: “formally verified design and tested”. Protection profiles explain a set of security features that should be provided for a specific product type.

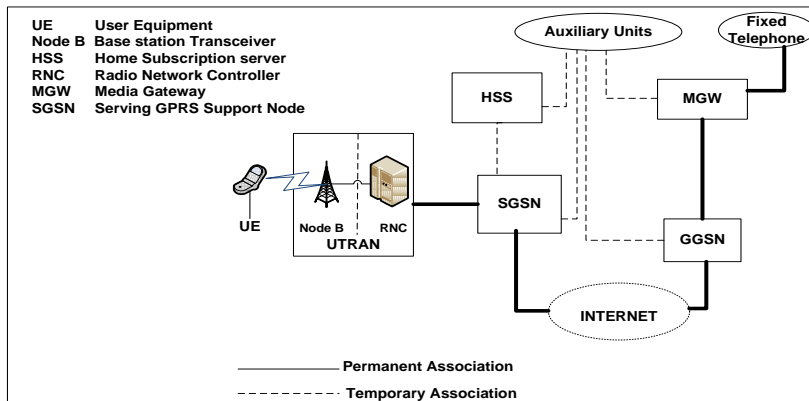
In a smart card there are three distinct layers/modules that are integrated circuit, operating system, applications. These components can be analyzed separately or as a whole system for security [34]. The modular approach (evaluating each module separately) is more suitable since the change in one module means only checking for that particular module and not for the whole platform [31, 34]. Moreover, after the modular security evaluation the composition evaluation (the whole system) can be done [31].

Security is the foundation of smart card development. The blend of logical and physical security mechanisms that form a unified system can be evaluated according to international standards to ensure a high level of security. Thus the ability to store information and execute cryptographic protocols in a secure manner provides a great success for smart cards to be used in sensitive security areas.

## **2.4 Identity management in UMTS system**

Universal Mobile Telecommunications System (UMTS) is one of the 3<sup>rd</sup> generation technologies, and now the 4<sup>th</sup> Generation, where “All IP” concept is introduced, is being developed.

UMTS phones are designed in such a way that they easily roam in UMTS networks. Besides, they have a capability to switch to GSM where UMTS coverage is not available, thus providing flexibility.



**Figure 4: ALL IP UMTS**

The General UMTS architecture consists of the following elements:

- User Equipment (UE) in the above diagram is the mobile terminal.
- Node B is a base station transceiver.
- Radio Network Controller's (RNC) are interconnected calls can be called transparently between them. RNC and node B together are called UMTS terrestrial radio access network (UTRAN).
- The Home subscription server (HSS) contains data including identities, IP/telephone numbers, service provisioning information and security support.
- The Auxiliary units support functionalities related to multimedia handling e.g. session control, multicasting, conference facilities.
- The location information is in Gateway GPRS support node (GGSN).
- Media Gateway (MGW) is a place where there is a format conversion between telephone network and an internet.

### Security of Identity in UMTS architecture

The architecture for the security of identity in UMTS is inherited from the previous architectures (GSM), added with some extra functionality [39]. It is one of the important part of current E-commerce and other applications to have secure and protected identity and anonymity of user. The security features of identity that are specified with the security of UMTS architecture discussed in [40] are as follows:

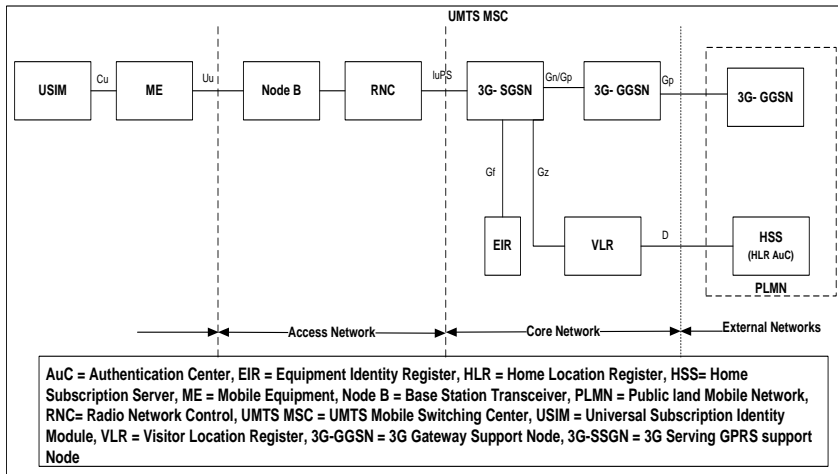
- **User Identity Confidentiality:** The property in which International mobile equipment identity is not be identified during communication.
- **User Location Confidentiality:** The property in which the location of the user cannot be found be found during the active link on radio interface.
- **User Untraceability:** To be unable to find out that the user is currently using which services.

### Identities in UMTS Architecture

In the UMTS architecture there are a lot of places where identity can be leaked as can be seen in



the diagram given below.



**Figure 5:** UMTS architecture [40]

To avoid the leakage of identity, different identities are used to protect user privacy. They are as following:

- Mobile Subscriber Integrated Services Digital Network (MSISDN): This identity represents the user phone or mobile number.
- International Mobile Equipment Identity (IMEI): This is the Mobile Equipment (ME) serial number. This can be helpful in fraud prevention.
- International Mobile Station Equipment Identity and Software Number (IMEISV): It is a like IMEI which represents both hardware and software.
- International Mobile Station Identity (IMSI): It is the permanent identity stored in the Universal Subscriber Identity Module (USIM) secure component i.e. smart card
- [Packet] TMSI is a temporary identity assigned to ME by the local network in which the user is registered.

The use of different identities is described in the table given below.

**Table 1:** Identities in UMTS components [40]

Parameter	Type	HLR	VLR	SGSN	GGSN
MSISDN	T	M	M	M	M
IMEI	T	-	-	C	-
IMSI	P	M	M	M	M
P-TMSI (Signature)	T	-	-	C	-

Legend: M = mandatory, C = conditional, T = temporary, P = permanent

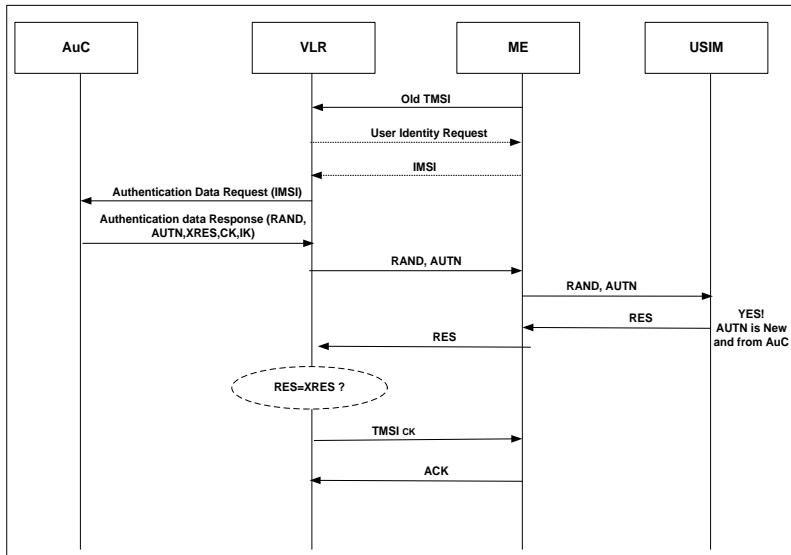
## Mutual Authentication in UMTS

The mutual authentication protocol in UMTS depends on the challenge-response pattern. The main aim of this authentication is two ways:

- Authentication of Mobile Equipment (ME) at HLR or Base Station (BS) for billing and other purposes.
- Authentication of BS from ME to avoid from fraud BS.

The ME contains an USIM that is a tamper resistant smart card embedded with some cryptographic algorithm, a master key that is shared with the Authentication center (AuC) and a permanent identity IMSI. During the authentication procedure of ME, the AuC generates an integrity key (IK) for message authentication and a cipher key (CK) for message encryption.

The mutual authentication and key agreement are showed in the following diagram.



**Figure 6:** Mutual Authentication and Key Agreement [40]

The mutual authentication can be described by the following steps:

- When ME discover new VLR, it sends the old TMSI as acquired in the previous execution. If VLR does find the IMSI against the given TMSI then it asks for IMSI.
- When the VLR identifies the ME, it sends request for authentication data from Authentication center (AuC) of ME Home network.
- The AuC generates a challenge for ME from the master key with IMSI. The replay attacks are avoided by adding sequence no (SQN) and random no's (RAND). The AuC also computes an authentication token (AUTN), expected response (XRES), Integration key (IK) and Cipher key (CK) forming an authentication vector and send it to VLR

- When the VLR receives the authentication vector it challenges the ME with RAND and AUTN
- With master key embedded in the USIM and RAND, the USIM can authenticate the challenge through AUTN.
- Then the USIM computes the response (RES) to challenge the cipher key (CK), Integrity key (IK) and authentication key (AK). The RES is returned to VLR
- The VLR compare it with the expected response (XRES). If they match the keys CK and IK can be transferred to RNC, which can establish a secure communication channel with ME.

At times we send the TMSI clear over the connection which is against the security features for UMTS architecture (User location confidentiality and user Untraceability), as an eavesdropper is able to link the different connections established under the given TMSI.

There are some solutions proposed in [40], it discussed the problem and its solutions in detail. In [41] other attacks on UMTS are discussed.

## 2.5 Authentication and authorization in Linux

There are many ways of having secure authentication and authorization in Linux, and with time the mechanisms are improving.

### Passwd File

Normally for all Linux distributions user information is saved in the *passwd* file [44] which can be found at */etc/passwd*. This text file contains the following information

- User login
- Encrypted password
- Unique Numerical User ID (UID)
- Numerical group ID (GID)
- Optional Comment field (Such as name, phone etc)
- Home Directory
- Preferred shell.

The entry in the *passwd* file can be seen as

**<User login: Encrypted password: UID: GID: optional comments: Home Directory: Preferred shell>**

For example, we can see the entry as [44]:

```
Pete:K3xcOIQnx8LFN:1000:1000:Peter Hernberg,,1-800-FOOBAR:/home/pete:/bin/bash
```

### Shadow Passwords

As *passwd* file is readable for all users and it contains all information including the encrypted passwords, so making it easy for accessing the password cracking. To get rid of this problem shadow password were developed. In this method the password field in *passwd* file has been replaced by “x” and the encrypted password is stored in the shadow file which is stored at */etc/shadow*. This file is only accessible by root user so making it a bit safer than *passwd* file.

The shadow file contains the following information:

- User Login
- Encrypted password
- Number of fields relating to password expiration.

The entry in passwd can be seen now after shadow password as [44]

```
Pete:x:1000:1000:Peter Hernberg,,1-800-FOOBAR:/home/pete:/bin/bash
```

And the entry in shadow file will be [44]

```
pete:/3GJllg1o4152:11009:0:99999:7:::
```

### Group Information

The group information in Linux is stored in a group file located at */etc/group*. The data in the group file are as follows

- Group name
- Password
- Numerical ID of group (GID)
- Comma separated list of group members

The entry in the group file can be seen as

```
pasta:x:103:spagetti,fettucini,linguine,vermicelli
```

“X” in the above example is a group password and it can also be shadowed. The shadowed group password is stored in */etc/gshadow*.

Some of the built in groups in Linux are showed in the table 2 given below.

**Table 2:** Linux built-in groups

Group	Rights
Root	This is a group which has full all the rights and is administrator. This group has ID 0 [45]
Wheel	The wheel group is a group which limits the number of people who are able to “su” to root, meaning that other user can have root privileges.[46]
Lpr	lpr is for the printer. Adding a user to this group allow that user to use the printer.
Shadow	for programs needing access to shadowed-passwords
Users	By default every new user is assigned this group.

### Encrypted Passwords

UNIX used crypt () function for encrypting the passwords. But with time the encrypted passwords were cracked, so new mechanism were introduced to make the cracking of password difficult. MD5 is one of the hashing algorithms that are added for encryption by most Linux distributions, but still this does not make it the full safe from cracking.

## Pluggable Authentication Model (PAM)

Nowadays Pluggable Authentication module (PAM) comes with many Linux distributions and is one of good mechanism for authentication of users. Before this scheme as discussed above, we do not have any method for authentication for more than one program. If a program needs user information for validation then there was no solution if we have different authentication schemes for different programs. PAM is the solution of it by enabling programs to transparently authenticate users.

The purpose of PAM project was to have separate development for privilege granting software's and secure authentication schemes. It doesn't deal where the password is saved like in `passwd` file or on a server in different place. When a program needs to authenticate a user it provides library which contains the proper functions for authentication.

PAM strength is its flexibility. We can configure PAM in many ways for a program that can be as follows:

- we can allow certain programs to authenticate users
- we can only allow certain users
- to warn when some program attempt to authenticate
- we cannot allow any user to have login privileges

In short PAM enables strong per-service authentication features, shadow passwords, strong hashing functions, and change mechanisms with changing system requirements. The flexibility is available at the small cost of increased complexity [47]. Thus, it provides a complete control on the authentication.

## Support of PAM by Linux distributions

The PAM is supported by nearly all distributions. Some of the names of the distribution are as following [44]

- Redhat since version 5.0
- Mandrake since 5.2
- Debian since version 2.1 (partial support in 2.1 -- complete support in 2.2)
- Caldera since version 1.3
- Turbolinux since version 3.6
- SuSE since version 6.2
- Ubuntu

## PAM Configuration Files

PAM configuration files are stored in the `/etc/pam.d` directory or all the relevant configuration information can be stored in the `/etc/pam.conf` file. If `pam.d` directory and `pam.conf` file are present then `pam.d` is given preference and `pam.conf` file is ignored.

In `/etc/pam.d` directory each service has its own configuration file, which is named as the program or service name. E.g. the login application (`/bin/login`) is configured in `/etc/pam.d/login` file. When the application programmer defines their applications, they should install its configuration file for installation of that service. Usually the operating system installers do this.

## Format of Configuration file

The command of a PAM configuration file consists of four possible arguments which are as following

- Module interface
- Control flag
- Module path
- Module arguments.

The command in configuration file can be seen as:

```
interface control_flag module_path [module_arguments]
```

In the above command structure, the module arguments field is optional. The following is the example line that has the first three fields:

```
auth required pam_unix.so
```

**Module interface:** It shows the type of authorization of a module. A PAM module may consider one or all four possible interfaces. The administrator can specify for each interface, in the configuration file for a service and for the module. The four possible interfaces discussed are as following [47]:

- **Account:** This interface checks if an account is authorized to use the system, e.g. to check if it exists, its expiration time, either it is allowed to access at a particular time or a particular service.
- **Auth:** This interface authenticates a user either by than checking a password or can be by another mechanism. **auth** modules are also allowed to set credentials such as group memberships or Kerberos tickets.
- **Password:** This interface is for checking and setting password authentication.
- **Session:** This interface configures and manages a user's session.

**Control flags:** In configuration file a control flag is specified for each interface that determines that what PAM will do next, depending upon the result of the check performed. There are four control flag types which are discussed as following [44, 47]:

- **Optional:** In case of Failure to authenticate using this module results in direct denial of authentication.
- **Required:** In this case Failure will result in denial of authentication, although PAM still will call the other modules programmed for this service before completely denying authentication.
- **Requisite:** If this module authenticates successfully, then PAM will authenticate it, even if the previous required module results were failed.
- **Sufficient:** The importance of this module success or fail is only considerable if it is the only module for this type of service.

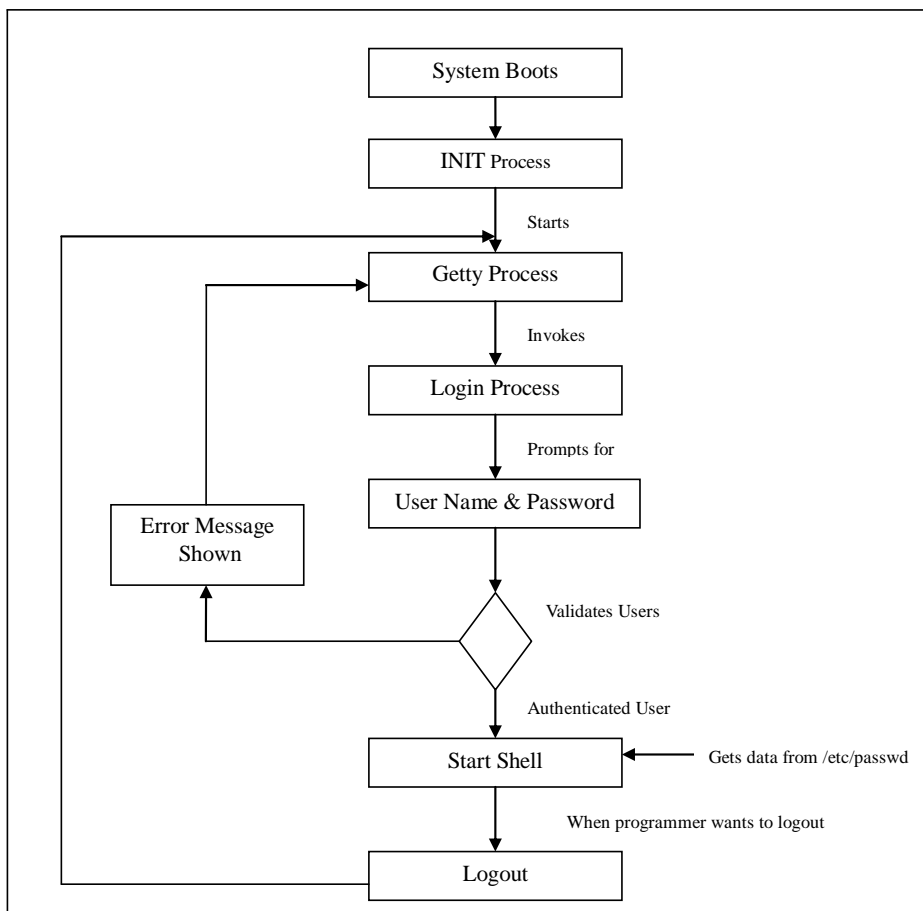
**Module path:** It tells PAM the location of the module, which is normally a full pathname that includes the module name and extension, e.g. */lib/security/pam\_unix.so*. In case where no path is specified, then the PAM by default searches the module in */lib/security*. The module path may also include the variable *\$ISA* as shipped from the vendor.

## Login and Authentication Process in Linux

When the system boots, there will be login prompt on screen. This login prompt is generated by a program, generally **getty** or **mingetty**, and it is regenerated by the init process whenever a user ends a session on the console. The process of login can be defined as following [48]:

- When the system boots, the init process starts the **getty** process.
- The **getty** process invokes the login process and the system asks for user name.
- The user enters the user name to login
- After this, the login process asks the user for a password, authenticates it.
- If the authentication is successful, the user's shell is started. The login program will get the data required from `/etc/passwd` file to decide which shell program to run. While on failure the program displays a message of error and then **init** will again start the **getty**.
- When the user logout, the shell program ends and we are again in the process.

The process is depicted in the chart below as well.



**Figure 7:** Login Process in Linux

## Access Control Lists (ACL) in Linux

Access Control Lists are an element of the Linux kernel and they are supported by Ext2, Ext3, ReiserFS, JFS and XFS. With the help of ACLs, complex problems can be solved without implementing complex permission models on the application level.

The use of ACLs can be in such situations where the traditional file permission concept is not sufficient. It allows assigning permissions to individual users or groups even if they are not from the owner or the owning group.

As discussed in [49] and [50], the traditional POSIX (Portable operating system interface) Access control lists permission concept which is similar to traditional file system, uses 3 classes for assigning permissions in the file system, the *owner*, the *owning group*, and *other* users. It uses 3 permission bits for setting each user class, giving permission which can be read (r), write (w), and execute (x).

In this method, the owner class permissions describe the access rights of the file owner; the group class permissions describe the access rights of the owning group, and the other class permissions describe the access rights of all users that are not in the two previous classes.

An ACL has a number of entries which describes the rights of each file system object with an ACL representation. e.g., “rwx r-- ---” means that for a regular file it has read, write and execute access for the owner class, read access for the group class, and no access for others. The entry types are as in the given table below.

**Table 3:** Types of ACL entries [50]

Entry Type	Text Form
Owner	user::rwx
Named user	user:name:rwx
Owning group	group::rwx
Named group	group:name:rwx
Mask	mask::rwx
Others	other::rwx

Each of these entries has the following syntax: *Type: qualifier: set of permission*

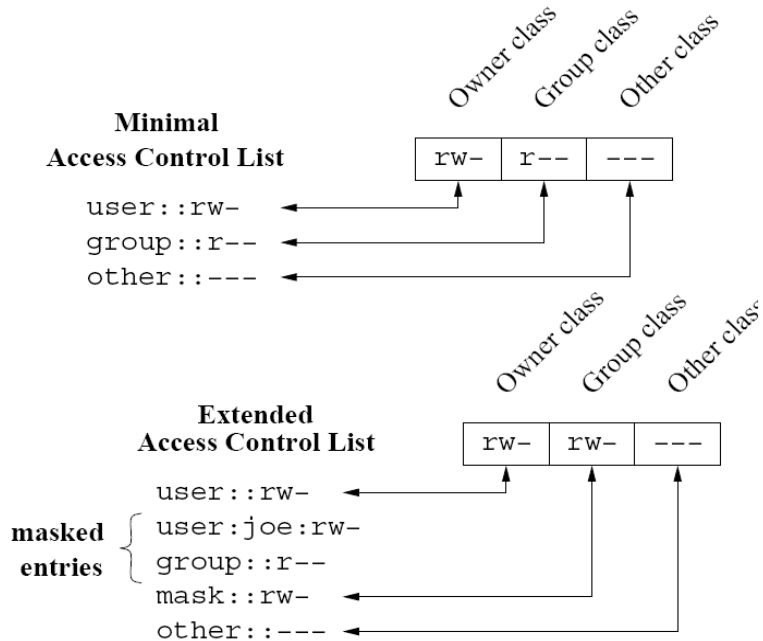
Qualifier defines to which user or group the entry applies. The qualifier is not defined for entries that require no qualification.

When ACLs are same as the file mode permission bits are then it is called **minimal ACLs** and they have three ACL entries. The ACLs having more than three entries are called **extended ACLs** and these ACLs also contain a mask entry and they may contain any number of named user and named group entries.

The named group and named user entries are allocated to group class, which already have the owning group entry. So now the group class may contain ACL entries with different permissions. The group class permissions are no longer only enough to represent all the detail of permissions of all ACL entries. Now the group class permissions represent an upper bound of the permissions.



In minimal ACLs, the group class permissions are exactly the same as the owning group permissions. But in extended ACLs case, the group class permissions are mapped to the mask entry permissions using mask. The mapping of the group class permissions can be seen in the figure given below.



**Figure 8:** Mapping between ACL entries & File mode Permission Bits [50]

As in the group class permissions denote the upper bound of the permissions granted by any entry. So With extended ACLs, this is implemented by masking permissions which can be seen in the table given below.

**Table 4:** Masking of permissions [50]

Entry Type	Text form	Permissions
Named User	user : joe : r-x	r-x
Mask	mask : : rw-	rw-
Effective permissions		r--

**Access ACL:** It determines the user and group access permissions for all kinds of file system objects.

**Default ACL:** These ACLs can only be assigned to directories. They conclude the permissions a file system object inherits from its parent directory when it is created.

### Support of Smart cards in Linux

The Linux-PAM login module also supports X.509 certificate based user login. For every user the unique digital certificates for that user can be stored on smart card. To access the certificate and its dedicated private key in Linux we can use an appropriate *PKCS #11* module.

For the approval of the ownership of a certificate (smart card), to login as a specific user, *pam-pkcs11* uses several modules called *mappers* that perform certificate-to-login mapping.

We can configure PAM PKCS#11 module to authenticate the user with smart card, normal username/password method or both at the same time.

### Writing PAM Module

Some time we need to write our own PAM module for carrying out some task that we want to be additionally added with the existing functionality. PAM modules are mostly written in C language. For development of PAM modules we need to install *libpam0g-dev* package and the following header files should be added in the file while writing a PAM module.

```
#include <security/pam_appl.h>
#include <security/pam_misc.h>
```

A sample code has been given in Appendix A with compilation steps. The steps involved in the writing a PAM module can be seen in the figure given below as discussed in [52]:

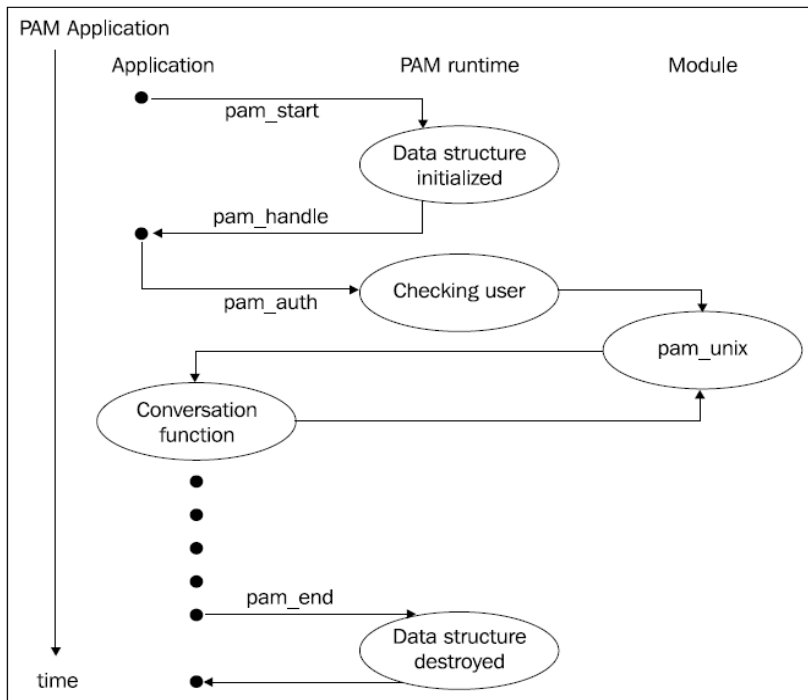


Figure 9: PAM Application Structure

- The first step in writing a PAM module is to initialize a PAM session. With initialization of PAM session we initialize all the required data structures and we can access the data with a PAM handler which is returned after initialization of PAM session. The following function starts PAM session and initialize the data structure

```
retval = pam_start("servicename", user, &conv, &pamh);
```

In above function the first parameter is the service name e.g. Login. The second parameter is the user name. Third parameter is the pointer to the conversation function, and the last parameter is PAM handler with which we can access the data structure later. The type of return values are discussed later

- We can authenticate the user by using the following statement

```
retval = pam_authenticate(pam, 0)
```

The first parameter is the PAM handler which is created in the first step. The return value types are discussed later.

- We can manipulate with the data structure now at this step. The data of the user can be accessed with the *passwd* structure in the *pwd.h* header file. The *passwd* header file contains the following data.

```
struct passwd {
    char *pw_name; /* user name */
    char *pw_passwd; /* user password */
    uid_t pw_uid; /* user id */
    gid_t pw_gid; /* group id */
    char *pw_gecos; /* real name */
    char *pw_dir; /* home directory */
    char *pw_shell; /* shell program */
};
```

- At the end it is required to end the PAM session. The session is ended with the following statement.

```
pam_end(pamh, retval);
```

**Conversation Function:** conversation functions are required to handle the call backs. The conversation function is used by modules to interact for the authentication like login name or password during authentication. The Linux has its own default conversation function. We can write our own conversation if required.

**Return Codes:** There are specific return values that are returned by the function as can be seen above. The *retval* shows the return value in the above statements.

**Table 5:** Return Codes

Return code	Management group	Meaning
PAM_SUCCESS	All	Everything went well
PAM_USER_UNKNOWN	Auth, Account, Password	The authentication token (user name) is not known
PAM_SESSION_ERR	Session	Any error related to opening or closing sessions
PAM_AUTH_ERR	Auth, Account	Authentication failed
PAM_ACCT_EXPIRED	Account	Account has expired

## 2.6 Authentication and authorization in Windows

### Security principals and access to objects

The Security in Windows system depends on the subject-object-actions relationship. In case when the subject requests for some action to perform on an object, it is checked by the operating system whether the access should be granted on the basis of security permissions associated with that object. As defined in [53] a subject can be a user, a computer or a service. And according to [54] an object can be a file, a job, a device, a process, a thread, shared memory sections or volumes etc.

The identity of the principal must be known by the operating system for making decision whether to grant some access to an object. So, to access the resources a user must be authenticated to the system during Windows logon process. Windows operating system uses unique security identifiers (SIDs) for identification of security principals. The SID's are assigned during the logon process.

The structure of the SID is as following:

$$\underbrace{S-1-5-21-2443930396-124871960}_{\text{Domain identifier}}-1960245352-1000_{\text{RID}}$$

S – SID designator. It consists of a character "S";

1 – It is the revision number of the SID specification;

5 – It is authority identifier. The value 5 is for SECURITY\_NT\_AUTHORITY. Other values are: SECURITY\_WORLD\_SID\_AUTHORITY, SECURITY\_LOCAL\_SID\_AUTHORITY, etc;

Domain identifier – it is used to identify a domain or computer that issued SID.

RID – It is Relative identifier. They are used to guarantee uniqueness of SIDs. The RIDs for the non-default users and groups starts from 1000 and increase by 1 with every new principal.

The user and group accounts are divided in to two types in the Windows systems which are user-defined and automatically created by the system. The Automatically created groups are further classified in built-in and system groups. System groups have automatic and dynamic membership which depends on the principal's activity type. The Built-in groups are not much different from the user-defined groups and the reason for using is to support the default Windows security model [55].

Some of the well-known built-in accounts in Windows system are as following.

**Table 6:** Built-in accounts in Windows

SID	user/group	description
S-1-5-domain-500	Administrator	Built-in system administrator account
S-1-5-domain-501	Guest	Built-in account that can be used to give limited access to those who do not have personal account
S-1-5-domain-502	KRBTGT	Service account used by Kerberos KDC
S-1-5-domain-512	Domain Admins	Members of this group are authorized to administer a domain.
S-1-5-domain-513	Domain Users	Includes all user accounts in a domain
S-1-5-domain-515	Domain Computers	This group includes client stations and servers in a domain
S-1-5-32-544	Administrators	Initially contains Administrator account as the only member. After the system joins a domain, Domain Admins group is added to this group.

**Table 7:** Well-known system groups in Windows

S-1-1-0	Everyone	This group includes completely all users
S-1-2-0	Local	This group includes users who log on to computer locally via connected terminal
S-1-5-2	Network	Contains users who log on via a network
S-1-5-4	Interactive	Includes users that log on interactively. Terminal server users are in this group, but they are not included in Local group [52].
S-1-5-5-X-Y	Logon Session	This SID is used to identify a logon session, not a principal. Each logon session of a user is assigned a unique ID (X and Y values are changed)
S-1-5-11	Authenticated users	Contains authenticated users

An access token is created by the Windows after a user is successfully authenticated. The Access token consists of the SID of the user, SIDs of all groups in which the user is a member that includes built-in and system groups, the SID which identify the logon session, the assigned privileges of the user and user's groups etc. If there is some change in assigned privileges of the user or change in one of the groups in which the user is a member of, or the user membership in some group changes, after that the user must have to relogin for the changes to take affect [55]. A new access token which contains updated information is created after the user relogs. Every process or thread executed by the user contains a copy of the user's access token. In case if a process or a thread requires access to some secured object then they present an access token to the system for getting access.

There is a security descriptor associated with every secured object in the system. The security descriptor defines who can access an object and what kind of access is allowed. The security descriptor consists of the SID of the owner of the object, SID of the primary group of the owner (as discussed in [53] the primary groups in Windows access control model are only used for POSIX compliance), a system access control list (SACL) and a discretionary access control list (DACL).

The DACL defines the users and groups and if they are allowed to access an object. In case when an object does not have associated DACL then it can be accessed by everyone [56]. The DACL contains a number of access control entries (ACEs). Each ACE have the SID of the subject, the flag that shows if access is either allowed or denied, and an access mask which describes the access rights for subject. In case if the DACL of object contains no ACEs, then no one can access the object [57]. The following are the access rights defined in Windows [58]:

- **Generic access rights:** It includes the basic rights like READ, WRITE, EXECUTE and GENERIC\_ALL that is combination of all three
- **Standard access rights:** it includes the rights like DELETE: the right to delete the object, READ\_CONTROL: right to read object's security descriptor, WRITE\_DAC: right to modify DACL, WRITE\_OWNER: to change the owner of the object and SYNCHRONIZE right.
- **SACL access right:** It is for getting or setting SACL in the object's security descriptor
- **Object-specific access rights:** One of the example can be the right to create files in a directory is FILE\_ADD\_FILE for this kind of rights.

The SACL controls in case of attempts to access an object are logged. The ACEs in the SACL indicate which access attempt types and subjects should be logged. Failures and successful

attempts both are logged. The majority of objects in the Windows system do not have SACL [53].

In the case of discretionary access control model the owner of an object declares the access permissions for other users and groups. The owner of an object and the system administrator has the full control over the object in Windows. If the object has empty DACL no one can access the object including the owner, but still the owner can change access permissions [53]. Though, if the DACL of the object has ACE with the SID S-1-3-4 “OWNER\_RIGHTS”, in that case the implicit READ\_CONTROL and WRITE\_DAC rights of the owner are ignored. This technique was introduced to prevent the users from modifying permissions for their own files [53].

When a secured object requested to be accessed by any process or a thread, the system checks the provided access token against the DACL of the object taking into account the requested access type. ACEs in the DACL are analyzed sequentially for SIDs that match those in access token until access is granted or denied or reach the DACL end. Thus, access is granted when one or more access permitted ACEs allow all requested access rights for any subset of SIDs in the security token of the principal [59]. The Access is denied in that case when the deny ACE, for one of the SIDs in the access token which denies any of the requested access rights is encountered in the DACL [59]. The access is denied in case if the end of the DACL was reached and there is some requested access right that was not allowed by ACEs.

For every user in Windows system there are rights and privileges. As defined in [60] a privilege is a right to perform system-related management operations which can be changing time, rebooting the system, loading drivers etc. Hence the privileges differ from the access rights that the subjects have for the securable objects. The Privileges reside in the assigned access token while access rights are described in the security descriptor of the securable objects. It is required to differentiate between user rights and access rights for the securable objects. The Logon rights are same as privileges except that they are used to allow a user for log on to the system and privileges describes what a user can do after logon [53, 55]. Some of the rights for logon are discussed in the following table.

**Table 8:** Logon rights in Windows

<b>Logon right</b>	<b>Description</b>
Access this computer from a network	Determines which users/groups can logon via a network. By default this right is granted to everyone.
Allow log on locally	defines who can interactively logon to the system via connected terminal
Allow log on through Terminal Services	defines who can logon to a remote computer via Remote Desktop Protocol [54]
Deny access to this computer from network	defines for whom network logon is denied
Deny log on locally	defines for whom local logon is denied

### **Role-based access control**

The alternative to discretionary access control system is Role-based access control (RBAC) which is used in Windows. The resources are accessed on the bases of role. Thus, privileges are assigned to roles. The role defines which operations and job functions a user can perform. When a user has the assigned role he obtains the privileges of that role.

The support for RBAC in Windows is provided by Windows Authorization Manager (AzMan). The Administrative tools allow defining the role-based authorization policies against which the access control decision will be made.

## **Authentication and logon process**

The two types of authentication/logon models in Windows are interactive and non-interactive. In case of interactive authentication the user is prompted for credentials. On the other hand non-interactive authentication uses the credentials that are previously entered by a user during the interactive authentication. Thus it can be concluded that the interactive authentication must always precede the non-interactive authentication. Non-interactive authentication is done when the user requests connection to other stations/servers in the domain.

The interactive logon can be either local or a domain logon. With local interactive logon the user can only access the local system resources while in case of the interactive domain logon a user can access the resources of the whole domain.

### **Logon process in Windows XP**

The components include in the Windows XP interactive logon architecture are Winlogon process, Local Security Authority (LSA), Graphical Identification and Authentication (GINA) dll and authentication packages (NTLM and Kerberos). The process that is responsible for managing logon procedure is Winlogon process. It makes sure that no other illegitimate processes can interrupt the logon information provided by the user [54]. Winlogon depends on the GINA for obtaining the user logon information. After obtaining the credentials the GINA calls LSA to authenticate the user by using one of the authentication packages. The result of authentication is forwarded to GINA which returns it to the Winlogon process. If the authentication is successful then Winlogon process starts the user's shell.

The logon process can be started by a user with pressing CTRL+ALT+DEL which is the secure attention sequence (SAS). During the boot process Winlogon registers this sequence and other processes cannot interrupt this sequence other than Winlogon [54]. After detecting the SAS Winlogon calls GINA for obtaining the user's credentials. An interface is provided by GINA to get the credential from the user. To modify the standard interactive logon procedure, the default GINA dll (MSGina.dll) can be replaced by a custom GINA. E.g. a custom GINA can communicate with an external device for getting the user's credentials. If the default authentication packages are unable to analyze the credential information obtained through the custom GINA, then a custom authentication package should be used. LSA supports custom authentication packages. The ability to use a custom GINA and authentication packages gives us the advantage of implementing any authentication scheme virtually.

After the user's credentials are obtained the GINA calls the LsaLogonUser function to authenticate a user by using one of the authentication packages. LSA uses particular authentication package to authenticate the user. In case of local logon the local LSA authenticates the user, but in case of the domain logon, the LSA on the domain controller authenticates the users [61]. There are two packages provided by Microsoft authentication. One is MSV1\_0 authentication package which is used for the local logon and the second is the Kerberos authentication package for the domain logon. The MSV1\_0 compares the user name and hashed password with the stored user data in the Security Account Manager (SAM) database [54]. In case of the cached domain logon, MSV1\_0 can be used and in that case the cached credentials are stored in LSA database in the encrypted form. The result of authentication is returned by MSV1\_0 to LSA which is then forwarded to GINA, and if the authentication succeeds then a logon session is created. The Kerberos authentication package operates principally in the same manner but the only difference is that authentication exchange is done through the network and the authentication decision is made on the domain controller.

If the authentication succeeds then the LSA checks the local policy database for the logon rights of the user. The logon session is terminated and the failure notification is sent to Winlogon in case the user does not have appropriate logon rights [54]. LSA creates access token with appropriate account SID, group SIDs, session SID, and a set of privileges retrieved from the LSA policy database in

case when the user has appropriate logon rights. LSA passes the authentication result to GINA and then forwarded to Winlogon by GINA. Winlogon additionally receives the user's access token if the authentication succeeds and the Winlogon launches user's shell (default is Explorer.exe) and provides it with user's access token [55] so that the shell can perform operations on behalf of the user. Other processes are created from the shell and each process inherits the user's access token.

## **Logon process in Windows Server 2008, Vista, and Windows 7**

Credential Provider model is used for the logon architecture in Windows Server 2008, Vista and Windows 7 and redesigning of the authentication model is done so that GINA is not used [62]. Furthermore, in previous Windows versions a console session along being interactive logon session also hosted system processes and services [63]. In the new architecture session 0 (Console session) became the non-interactive authentication process, so the users log on to separate sessions starting from session 1. This makes the services in session 0 isolated from the user applications and services that run with higher privileges are protected from attacks and malicious application code [64].

The new interactive logon architecture consists of the components that are Winlogon process, the logon user interface (LogonUI) process, credential providers, LSA and authentication packages. The default credential providers support password and smart card authentication [62]. Thus, makes possible to install multiple custom credential providers. To support different identification mechanisms, custom credential providers can be developed. It can be decided by the user to use which credential provider or the selection of the credential provider can be event-driven.

When a user enters the SAS the logon process begins. After beginning of logon process the Winlogon starts the LogonUI for providing the user interface for logon. LogonUI requests the credential providers for obtaining the user's credentials. LSA is called to authenticate the user through one of the authentication packages after obtaining credentials. If authentication succeeds, LSA checks the local policy database for the logon rights of the user. If the rights are sufficient then it creates the access token and forwards the authentication result and the access token to Winlogon. If authentication is successful then the Winlogon launches the user's shell.

As discussed in [65], the logon customization is much easier and more secure in the new architecture with credential providers as compared the old model where the custom GINA dll had to be developed and GINA was responsible for the graphical logon interface. In the new architecture the graphical logon interface responsibility is taken by LogonUI. In new technology the credential provider informs the LogonUI where graphical control elements like checkboxes or edit boxes are required to obtain user's credentials. If LogonUI doesn't work for some reason, then Winlogon will simply restart it [62].

### **Standard smart card logon**

The advantage of smart cards is that it provides a two-factor authentication that is based on the possession of a smart card and the knowledge of a PIN. The smart card stores a private key and a corresponding X.509 certificate with a public key securely and the private key never leaves the card. Moreover, it is stored only on the smart card [66]. For authentication the smart card performs the cryptographic operations using the private key for proving to the authentication server that the principal's smart card contains the key. Before performing the cryptographic operations, a user must first authenticate to the card by presenting the PIN. The user is only prompted for the PIN. The workstation communicates with the smart cards via smart card reader.

In Windows systems smart cards can be used to log on with domain accounts and not with the local accounts. That's why the standalone smart card logon is not natively supported in Windows systems still there are some commercial products that offers this functionality. The domain smart card logon supports an offline logon capability which means that even in case of a network service disruption or a failure it is still possible to logon to a workstation that belongs to that domain.



The process of Smart card domain logon session for Windows XP [66] is done as following:

- First the Smart card is inserted into the card reader. This will automatically start the logon process.
- GINA is called by Winlogon for obtaining user's credentials. A logon screen is presented by GINA to the user and the user is prompted only for a PIN.
- GINA forwards the received PIN to the LSA;
- The PIN is used by the LSA for accessing the smart card.
- Kerberos Authentication Package (Kerberos SSP) is called by LSA. Kerberos SSP creates a Kerberos Authentication Service Request and sends it to the KDC. The request contains principal's certificate and a cryptographic signature generated with the corresponding private key for the Kerberos pre-authentication [67].
- The KDC validates the certificate which includes verification of the certification path, checks revocation status, etc. and checks the digital signature. After these checks the KDC retrieves the user account information from Active Directory. This information is required to construct a TGT which is created. The TGT Authorization data fields contains the principal's SID, the SIDs for domain groups to which the user belongs and (in a multi-domain environment) the SIDs for any universal groups in which the user is a member. For encryption of symmetric encryption session key, the public key from the certificate is used. The KDC digitally signed the response with other things that are TGT, the KDC certificate, and the encrypted session key. The client will be able to decrypt the session key and use it for subsequent interactions with KDC in case if the client possesses the private key that corresponds to the public key in the certificate.
- When the client receives the response, it validates the KDC certificate and checks the digital signature. With using the private key, the client can decrypt the session key for communication with KDC. For log on to the computer it is required to obtain the Ticket Granting Service (TGS) to the local computer from KDC. The remaining part of the authentication procedure is identical to the standard logon session.

Windows caches credentials after successfully authentication which gives the capability of performing authentication local or offline log on to the computer with the domain account even if the domain controller or the network connection is failed. Though, during the local smart card logon with cached credentials a Certificate Revocation List (CRL) check is not performed [68].

### **Windows Vista Smart Card Infrastructure and logon procedure**

Windows Vista supports the following not like the previous version of Windows [69]:

- Smart cards that contain several certificates only for logon purpose other than the certificates for other purposes. The smart card memory space tells that what number of the certificates can be stored.
- Another option is to change the PIN and unblock a smart card without requiring logging on first with a standard user name and password.

The Windows Vista supports a password credential provider and a smart card credential provider by default. For enabling the custom authentication mechanisms, a custom credential provider should be used.

The Logon steps in Windows Vista are discussed as following [69]:

- The logon does not start with insertion of card only; it starts after the SAS is pressed. In that case the WinLogon requests the logonUI for obtaining the credential information.
- A list of smart card readers and a list of inserted smart cards exist in the smart card credential provider. It checks for every card whether the logon certificate exists on the

card. The Found logon certificates are retrieved from the smart card and copied into a temporary secure cache. After this the smart card credential provider provides the logon certificates to the LogonUI.

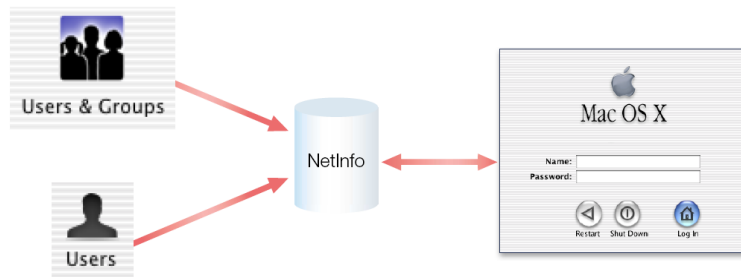
- A logon user interface is displayed by the LogonUI with found certificate logon tiles to a user. The user has the option to select one of the tiles and a PIN input box is displayed for getting the PIN from the user. The PIN entered is encrypted by the smart card credential provider.
- The smart card credential provider forwards the encrypted PIN, user name etc. to the LogonUI which calls the LsaLogonUser function and provides the received data to the LSA.
- Furthermore, the LSA calls Kerberos Authentication Package (Kerberos SSP) to create a Kerberos Authentication Service Request. The remaining part of the Kerberos authentication procedure is identical as in Windows XP.
- In case of successful authentication the certificates are read from the card (including the root certificates) and stored in the user's certificate store (MYSTORE).
- When the card is removed, the certificates are also removed from the temporary secure cache. Still the certificates will be present in the user's certificate store (MYSTORE).

As discussed in [70] Windows 7 and Windows Server 2008 R2 have some negligible enhancements to the smart card platform as compared to the Windows Vista that are mainly related to the Plug and Play service and smart card drivers.

## 2.7 Authentication and authorization in Mac OS X

One of the most obvious differences between UNIX implementations and Mac operating system is that the user information is stored in database called NetInfo, not in flat files like `/etc/password` as in UNIX systems.

As discussed in [71], in old versions of Mac the hashed password is used to be stored in the database and was accessible by every user, but now they use to store it in a shadow file to make it secure and only accessible by root user. The shadow files can be found at `/var/db/shadow/hash/` directory. The password is stored in each user name file which is not the same like the user name but generateduid NetInfo field.

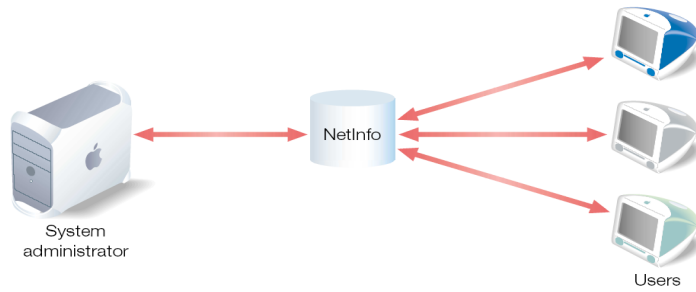


**Figure 10:** Interaction of NetInfo and Users [72]

In the figure above all the information of users and groups are stored in the NetInfo so when user logs in the information is authenticated with the information in the database.

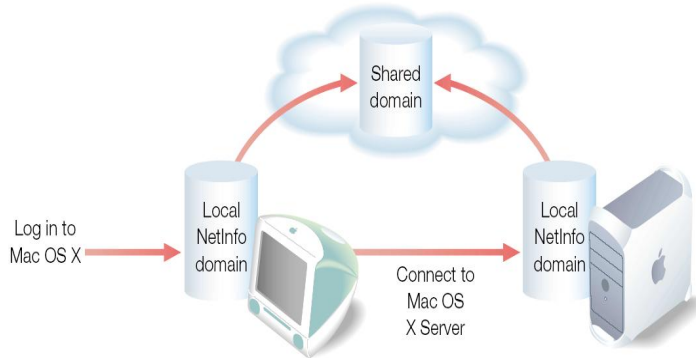
With NetInfo it is not required to know where the information is stored. When any process requires some information, it request for the required information which is provided by NetInfo. Second benefit of NetInfo is for administrators who are managing more than one computer. They can easily

manage the computers as can be seen in the figure given below.



**Figure 11:** Administrator Management with NetInfo [72]

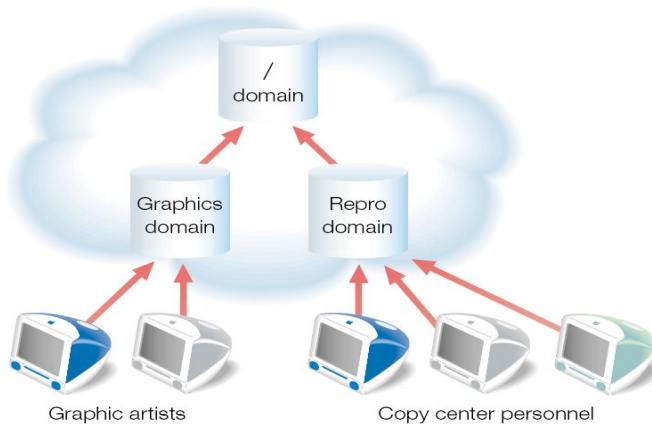
One of the real powers of NetInfo is storing data not only in local but can save it in shared domain which makes the work of administrators much easier.



**Figure 12:** Shared Domain in NetInfo [72]

In shared domain as shown in diagram shown above the user data is first search in Local NetInfo domain, if not found then it goes to the shared domain for getting the required data.

We can share the data in hierarchal form with NetInfo as well. The root node is denoted as “/”. The following diagram can explain scenario.



**Figure 13:** NetInfo Hierarchies [72]

In the above diagram there are two domains one is Graphic artists and the second is Copy center personnel. The “/” is the shared between both the domains. The child and parent are attached with the method called Binding. There are three kinds of binding discussed in [72] are as following:

- Broadcast Binding
- Static Binding
- DHCP Binding

Each record of user in the NetInfo consists of the following items:

- User ID (UID)
- Short Name
- Real Name (Users full name)
- User’s password encrypted using one-way encryption algorithm.
- Home Directory style. (It is used by Server admin to distinguish among home directory styles like none, local, custom)
- Absolute path of user’s home directory
- Home location is present if home directory is on an Apple file server.
- Group ID (GID)

The administrator account is called root. Normally it is given UID 0 and can make the changes in the database. UID’s below 100 are reserved for system use. When we add the user with add user pane the UID’s starts from 500 and are assigned automatically. The GID’s are assigned automatically when we add a group from menu and starts from 100.

When a Mac OS X computer boots and domain binding occurs, then the following steps occur:

- NetInfo daemon called nibindd is started.
- The nibindd daemon starts another daemon called netinfod for each domain on the computer. The netinfod process is sometimes referred to as a NetInfo server
- The nibindd listens for requests from netinfod processes asking for parents, checking for the appropriate netinfod process and initiating binding. The nibindd and netinfod run in the background.
- Another process that is relevant to NetInfo is called lookupd. It is used to interact with NetInfo when legacy UNIX software requests administrative information which is now stored in NetInfo. The lookupd process makes it likely for software that uses Posix or BSD calls to retrieve administrative information from NetInfo.
- Then the user data is used for authentication. If validates than allowed otherwise rejected
- The UID in the record describes the operations that user can access.
- The GID also has effect on the access of a user privileges as group has assigned rights. It check at local system if not found that proceed towards the root domain.

## **Authorization**

In Mac OS X we have a security server i.e. a core services daemon which deals with authorization and authentication. It determines whether everyone is allowed or some users have the access.

The Security process is same like an official doing visa processing which can be seen in the table 9.

**Table 9:** Authorization in MAC OS

<b>Immigration</b>	<b>Authorization</b>
The immigrant provides a passport and visa to the immigration official.	The application provides the authorization reference, authorization rights set, and authorization options to the Security Server.
The immigration official uses the visa number to access information about the immigrant.	The Security Server uses the authorization reference to access credentials.
The immigration official uses the picture in the passport to validate the identity of the immigrant.	The Security Server asks the user to provide a user name and password for authentication.
The immigration official uses the privileges requested in the visa to look up the laws in the policy book.	The Security Server uses the rights in the authorization rights set to look up the rules in the policy database.
The immigration official uses the credentials to determine if the immigrant complies with the laws and should be granted the privileges requested in the visa.	The Security Server uses the credentials and authorization options to determine if the user complies with the rules and should be granted the rights requested in the authorization rights set.
The immigration official informs the immigrant whether or not he grants the privileges requested in the visa.	The Security Server returns a result granting or denying the authorization rights.

There is a policy database which contains a set of rules that the security server uses to authorize the rights of the user. Each rule has a set of attributes which is discussed in the given table.

**Table 10:** Rule Attributes and Description [73]

<b>Rule attribute</b>	<b>Generic rule Value</b>	<b>Description</b>
key		The key is the name of a rule. A key uses the same naming conventions as a right. The Security Server uses a rule's key to match the rule with a right. Wildcard keys end with a '.'. The generic rule has an empty key value. Any rights that do not match a specific rule use the generic rule.
group	Admin	The user must authenticate as a member of this group. This attribute can be set to any one group.
shared	True	If this is set to true, then the Security Server marks the credentials used to gain this right as shared. The Security Server may use any shared credentials to authorize this right. For maximum security, set sharing to false so credentials stored by the Security Server for one application may not be used by another application.
timeout	300	The credential used by this rule expires in the specified number of seconds. For maximum security where the user must authenticate every time, set the timeout to 0. For minimum security, remove the timeout attribute so the user authenticates only once per session.

## Support of Smart cards in MAC OS X

We can use smart cards in MAC OS X for authentication. Smart cards can exchange information with a personal computer through a smart card reader. MAC OS X provides a Smart Card Services software development kit (SDK), which has code that can be used to implement a PC/SC-Supported application. (The PC/SC is a work group that defines standards for accessing data from smart card reader).

The files can be found in Mac OS X at `/System/Library/Frameworks/PCSC`. The support of smart card in MAC OS X is based on the Movement for the Use of Smart Cards in a Linux Environment (MUSCLE) Open Source implementation of the PC/SC standard.

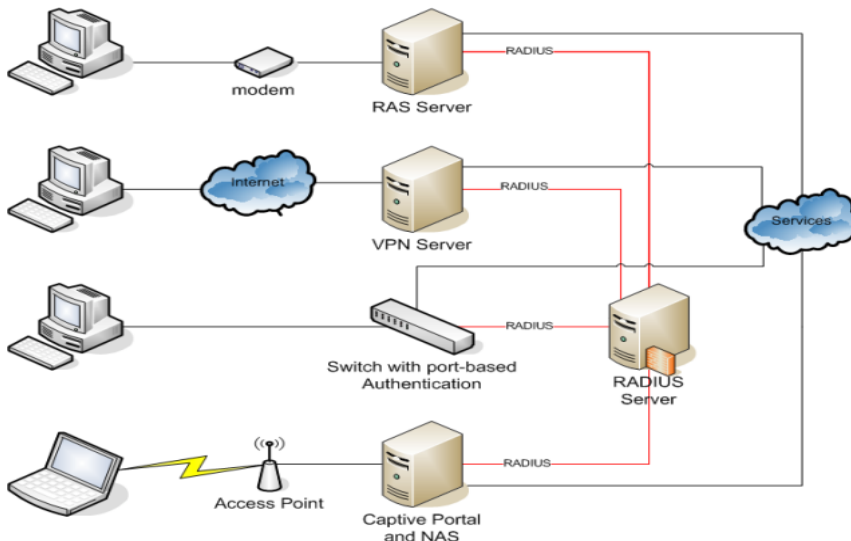
In MAC OS X v10.4 support login with smart card, the card must support signing with a public key. MAC OS X v10.4 and later version uses Keychain Services for supporting smart cards. It uses a plug-in called Tokend which serves as an interface that allows smart card developers to make their cards appear to be keychains. The three tokens installed with base MAC OS X are [74].

**Table 11:** Tokens installed with MAC OS

Token	Smart card specification
"CAC"	US Federal Government: CAC: Common Access Card GSC-IS: Government Smart Card Interoperability Specification v2.1
"BELPIC"	Belgian Personal Identity Card
"JPKI"	Japanese Public Key Infrastructure card

## 2.8 Remote Authentication Dial in User Service (RADIUS)

In a network we need to have a centralized Authentication and Authorization system to make the work of administrator much easy and the resources more accountable. For this purpose we can use Remote Authentication Dial In User Service (RADIUS) which is a networking protocol that can provide a centralized Authentication, Authorization and Accounting management system for computers to connect and use a network service [75, 76]. RADIUS is mostly used by ISPs and enterprises to supervise access to the Internet or any kind of internal networks which can be wired or wireless networks or integrated e-mail services. These networks may integrate modems, access points, DSL, access points, network ports, web servers or VPN's [75] which can be seen in figure 14 given below.



**Figure 14:** RADIUS Architecture [75]

RADIUS is a client server protocol which operates in the application layer, using UDP protocol for transport. The main functionality of the RADIUS server is as following

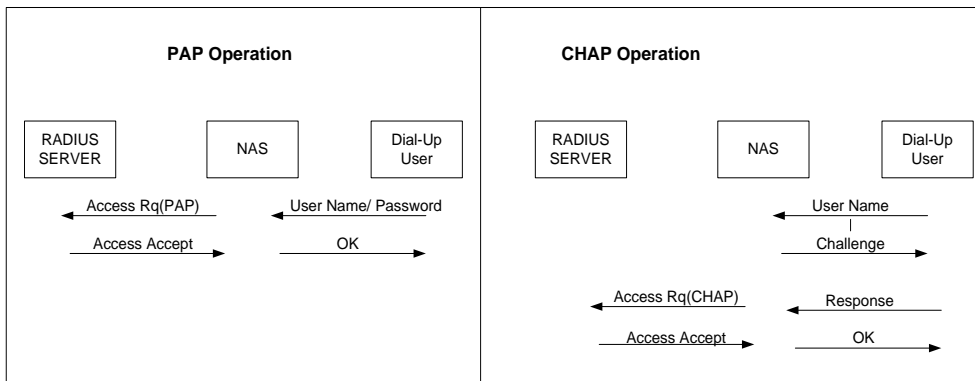
- The first task is to authenticate users or devices before giving them access to a network.
- The second task is to authorize users or devices for specific network services.
- The third task is to have an accounting for usage of those services.

### Core Messages

The core protocol of RADIUS is very simple and consists of basic four messages which are as following:

- Access-Request (From Network Access Server (NAS) to Authentication Server (AS))
- Access-Challenge (From AS to NAS)
- Access-Accept (From AS to NAS)
- Access-Reject (From AS to NAS)

From the above messages we can conclude that the point to point (PPP) dial-in modem protocol, have two options for authentication which are PAP and CHAP [76]. PAP is a simple method of authentication and requires the user name and password for authentication. While in CHAP a server is required to send a random number called challenge and the dial in system must encrypt and send it back for security checking. In CHAP password is not sent as clear but encrypted making it more secure than PAP. It is still vulnerable to dictionary attack because both the unencrypted and encrypted versions of the challenge are accessible to an attacker. The process of PAP and CHAP can be seen in the figure 15 given below



**Figure 15: RADIUS Different Authentication Operations**

### Core Message Format and Attributes

As there are four basic message but we can use attributes to change the other messages. The main structure of the RADIUS message consists of a series of attributes; each is a self-contained package of information that has meaningful message to both communicating parties. The structure of the message format can be seen in figure 16 as given below:

Code	Identifier	Length	Authenticator	Attributes.....
------	------------	--------	---------------	-----------------

**Figure 16: RADIUS Message Format**

**Code:** The code byte represents the type of the message e.g. 1 for Access-Request, 2 for Access-Accept, 3 for Access-Reject and 11 for Access-challenge etc.

**Identifier:** It is an arbitrary number used to match the request and response.

**Length:** The length field shows the number of bytes in the message.

**Authenticator:** The Authenticator field is 16 bytes (128 bits) long and depends on the type of the message. In Access-Request message this field contains the unique number called nonce. It is used for two purposes. One if password is being sent then it is used in combination with secret key for encryption. And the second purpose is it is used in reply message for integrity check. So we can see that in reply messages like Access-Accept, Access-Reject, and Access-Challenge, the nonce value is used to check whether the response is from a valid RADIUS Server or not.

**Attributes:** It is one of the fields that give RADIUS the power of extending itself to other technologies like EAP. Each message can carry one or more attributes and each is a self-contained package of information. Each attribute has the same format [76]:

- A Type field of one byte to identify the attribute
- A Length field of one byte that defines the number of bytes in the whole attribute
- Attribute specific data

## 2.9 Bluetooth security

Bluetooth is an open wireless standard which can be used for short range frequency communication. Bluetooth is integrated in many devices like PDAs, mobiles, laptops, printers etc, so we can transfer data without using cables.

Like normal wireless networks Bluetooth is vulnerable to service attacks, eavesdropping, man in the middle attack, and other attacks. Some of the attacks are: Bluesnarfing, Bluejacking, Bluebugging, Car whisperer, Denial of Service (DoS), Fuzzing attack. So, proper steps should be taken to make the communication secure.

Some of the features of Bluetooth that help in protecting from eavesdropping and malicious access to some extent are the frequency hopping and radio link power. But we need to take care of it so that our communication is safe.

### Bluetooth security modes

There are four security modes defined in various Bluetooth specifications which are as follows:

- **Security Mode 1:** In this mode no security is offered so attackers can attack easily.
- **Security Mode 2:** In this mode service level imposed security can be applied. First the Link Management Protocol (LMP link) is established and then the security procedures are initiated before Logical Link control and adaption protocol (L2CAP). L2CAP is in the link layer and provides connection oriented and connection less communication to the upper layers. In this mode of security the administrator can allow certain services to be accessed and can remove the access rights as well for some services. The authentication and encryption techniques are applied at LMP layer in this mode of security.
- **Security Mode 3:** In this mode the link level imposed security can be applied. And the security procedures are implemented before the physical link is established in this mode of security. In this mode all connections from and to the device are authenticated and encryption mechanism is applied to it making it secure. The authentication and encryption mechanism in this mode depend on the secret link key shared between the paired devices before the communication. In this mode like mode 2 the authentication and encryption is applied at LMP layer. This mode supports unidirectional and mutual authentication as well.

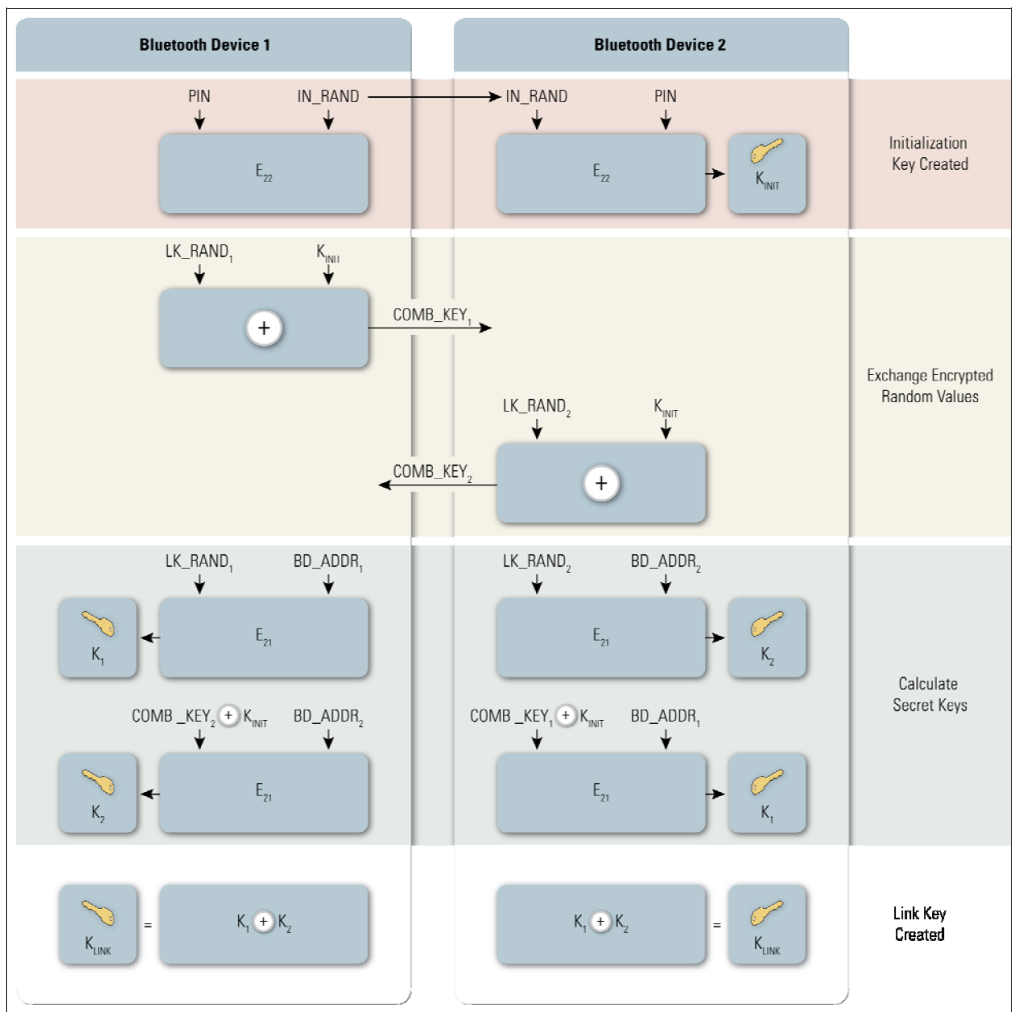


- Security Mode 4:** This mode is like mode 2 where service level imposed security can be applied and security procedures are applied after the link connection establishment. In this method Secure Simple Pairing (SSP) uses Elliptic Curve Diffie Hellman (ECDH) for key exchange and generation. There are four classes defined in this mode:
  - Authenticated link key required
  - Unauthenticated link key required
  - No security required

Secure Simple Pairing (SSP) was introduced in Bluetooth v2.1 + EDR specifications. SSP surpasses the maximum security level provided by the use of a 16 character alphanumeric PIN in Bluetooth v2.0 + EDR and earlier versions [77]. Bluetooth v2.0 + EDR devices, operating in security modes 2 and 3, derive the link key from the shared secret PIN [78]. SSP does not use permanent shared secret PIN to derive the link key, ECDH exchange is used instead.

Secure Simple Pairing uses ECDH thus providing protection against passive eavesdropping [77]. It also provides protection against man-in-the-middle attacks [77, 78], but not in all association models.

Figure 17 shows the SSP link key establishment. The link key established here is used in next steps for authentication and encryption.



**Figure 17: SSP Link Key Establishment for pairing [78]**

SSP association models provide authentication service, and whether the link key is authenticated depends on the used model. The authentication procedure/pairing confirmation by the user according to these models take place in the SSP Authentication Stage 1. There are four models offered by the SSP:

- I. **Numeric Comparison:** In this case it is considered that both the Bluetooth enabled devices have screens to show data and have “YES” or “NO” option to decide. A six digit is displayed on both screens while pairing and if match succeeds the user is given the option of “YES” on both devices. If the user selects yes the connection will be established otherwise the connection will not be established. The main benefit of it over using PIN is that the six digits are not used in link key, so if the attackers see it, will not benefit him. Numeric comparison model provides protection against man-in-the-middle [77].
- II. **Passkey Entry:** This scenario is designed for those devices in which one have input and display capability but the other device only have display capability. So the device that has display capability shows six digit number, which is entered by the user in the other device with, input capability. In this case like numeric comparison the six digits are not used to generate link key. This model also provides protection against man-in-the-middle attacks.
- III. **Just Works:** This situation is designed for the devices which don't have display and input capability. In this case the authentication is done as in the numeric comparison case but the only difference is that the digits are not shown on the screen. Just Works model does not provide authenticated link key [78] and does not provide protection against man-in-the-middle [77, 78].
- IV. **Out of Bound (OOB):** This designed for such cases where other means are used for other technology (e.g. Near Field Communication). In this case the user accepts the pairing via single push button. In this case the other technology used should be made secure to keep the system secure from the attackers.

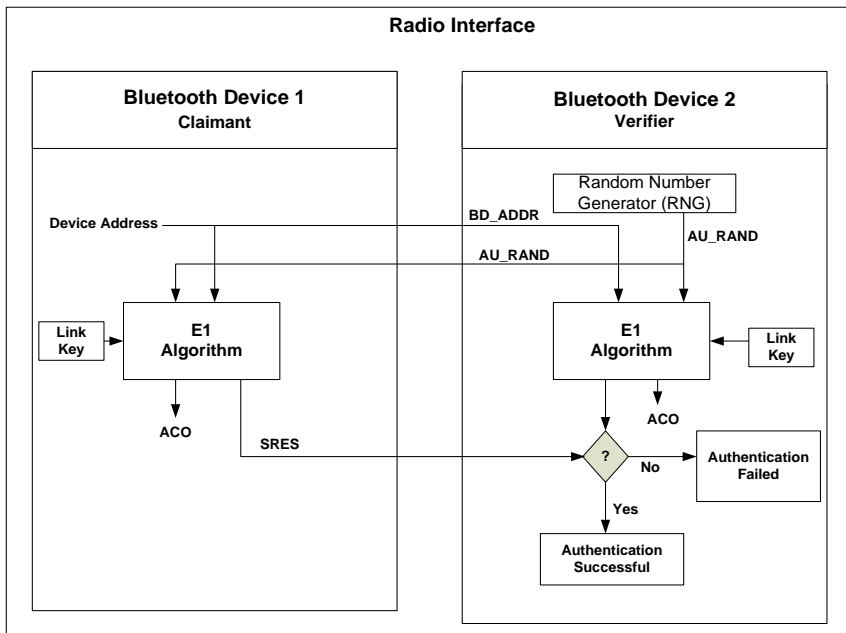
Authentication stage 2 is used to ensure that peers successfully completed exchange in Authentication stage 1 and to confirm the peering between devices. Unlike in Authentication stage 1, there is no user intervention in stage 2. After the peering has been confirmed the link key is generated from the derived shared key and some publicly exchanged values.

### **Bluetooth Authentication**

This phase starts after the link key was derived. In authentication process it uses the link key produced in the key generation process which can be seen in the figure 17. Bluetooth authentication uses a challenge-response protocol to authenticate the peer based on the knowledge of the link key. One device takes the role of claimant who has some identity and other is verifier who proves the identity of claimant. The authentication consists of the following steps:

1. At the verifier side it generates a random challenge of 128 bits (AU\_RAND), saves a copy of it and sends it to claimant side as well.
2. The claimant has now the unique 48 bit **Bluetooth Address** (BD\_ADDR), the **link key** generated before and the **Random number** received.
3. The claimant side uses **secure and fast encryption routine** (SAFER). The input to the algorithm is BD\_ADDR, AU\_RAND and link key, which produces a 128 bit output.
4. The most significant 32 bits of the output (SRES) are used for authentication while the remaining 96 bits which is called Authenticated Ciphering offset (ACO), is used in encryption in later stages.

5. The SRES (32 bits) is send to the verifier side.
6. Step 2 to 4 is also performed at the verifier side.
7. It compares both the SRES. If matches than authentication is successful, otherwise authentication fails. The procedure is also shown in the figure below:



**Figure 18:** Bluetooth Authentication [42]

For the mutual authentication the claimant and the verifier must switch roles.

### Bluetooth Encryption

There are three modes of encryption used which are as following:

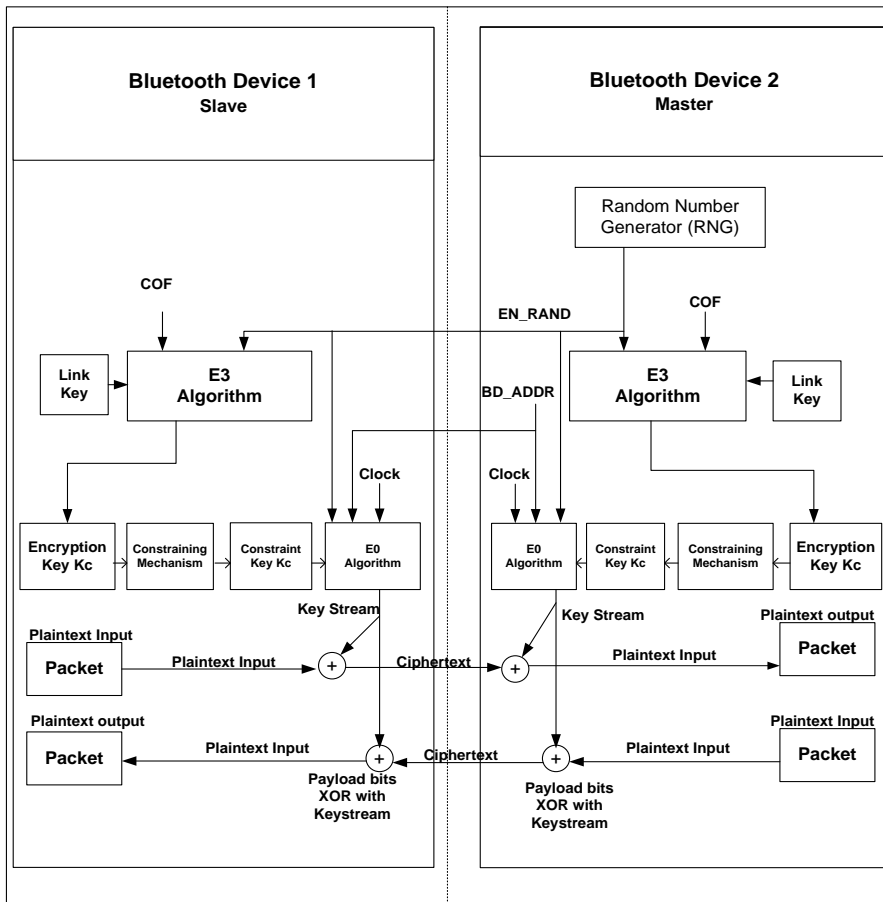
- **Encryption Mode 1:** No encryption is used in this mode.
- **Encryption Mode 2:** In this mode individual traffic is encrypted using keys generated from the link key produced at first stage, but the broadcasted data is not encrypted.
- **Encryption Mode 3:** In this mode all data is encrypted using keys generated from the link key.

The steps involved in the encryption are as following:

1. The keys for the encryption are produced using internal key generator (KG).
2. The internal key generator produce steam cipher keys (Kc) established from 128 bit link key, 128 bit random number (EN RAND), 96bit cipher offset (COF) which is the Authentication off set (ACO) produced in the authentication phase.
3. The stream cipher key (Kc) varies from 128 bits to 8 bits. There is key negotiation between the master and the slave Bluetooth devices. By default the key size is kept 128 bit to provide maximum security but can vary. The least security is 8 bits keys, which will make easy for attacker to attack.

4. The encryption Algorithm E0 as shown in the diagram below, used is based on the linear feedback shift register (LFSR).
5. The input to the algorithm is master identity (BD\_ADDR), 128 bit random number (EN\_RANDOM), a slot no and encryption key and the output is XORed with the plain text. (The slot no changes with each packet making some change with each packet).
6. The linear feedback shift register (LFSR) is initialized before sending a packet. The LFSR is reinitialized with each packet with having other static values and the in this case the slot no helps in bringing change with each packet.

The encryption process is shown in the following diagram below:



**Figure 19:** Bluetooth Encryption [78]

Bluetooth Security mode 4 with SSP is the most secure solution to be used for device peering. Besides, this mode, unlike legacy modes, does not require the knowledge of the shared secret PIN from both sides. The “Just works” association model being susceptible to the man-in-the-middle attack cannot be used in environments where high security is required. Thus we consider the SSP “Numeric comparison” and “Passkey entry” association models to be suitable solution for the secure dynamic establishment of the Bluetooth channel.

### 3. Mobile Phone Based Authentication Systems

The process of Authentication is verifying the identity of a user, user device, or some other entity as defined in [79]. For authentication it required to present a proof of identity to the authenticating party. All authentication schemes are based on the combination of the facts that something you know, something you have and something that you are. One of the most common authentication schemes nowadays used is static password authentication. But it has many weaknesses which are:

- It is hard to remember the strong randomly generated passwords that consist of the combination of letters, numbers, and special characters with adequate length.
- The human chosen passwords are often compromised with a simple dictionary attack;
- Several user-name/password systems make it hard to remember all data which consequently costs some users either with reuse of passwords or write them down.
- A Too strict password policy likes hard to remember passwords or frequent change can instead of strengthening the overall security of a system can actually weaken it, as users can end up writing down passwords.

The use of default passwords and careless users who reveal their passwords either accidentally or because of social engineering attacks can further reduce the security level of the system.

The username/password authentication may be sufficient for some systems but systems that have sensitive data require stronger authentication schemes e.g. biometric, smart card based, or one-time password based authentication schemes are considered to be much stronger than the ordinary user-name/password authentication, still the cost of the deployment and maintenance makes these systems less common.

An authentication solution that uses a mobile phone with a UICC card as a security token provides much stronger level of security as compared to the user-name password authentication with reduced operational costs [3].

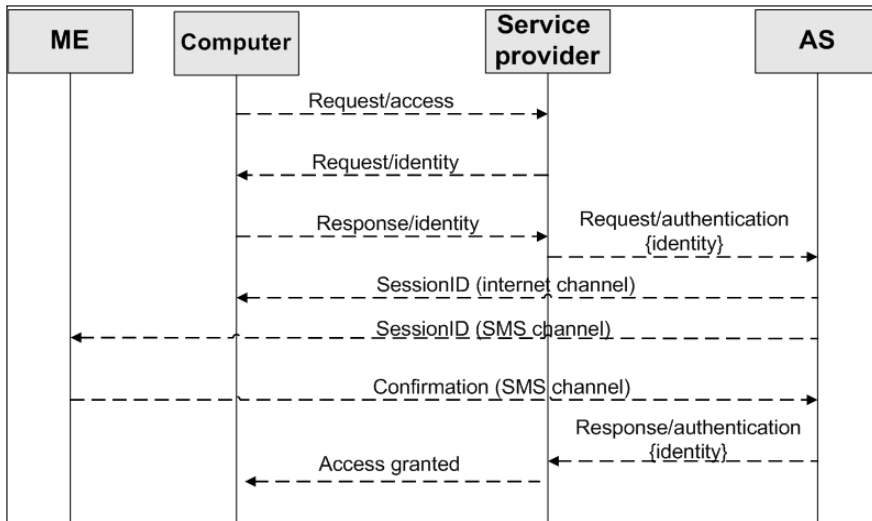
#### 3.1 General mobile phone authentication schemes

It is important to make sure that the same user controls both the devices in authentication scheme that use two devices (a computer and a mobile phone) and multiple channels for authentication protocol exchange (internet connection from the computer to the authentication server and GSM connection from the mobile equipment (ME) to the authentication server) [3].

##### 3.1.1 SMS authentication with session-ID check

In this mechanism the mobile network operator (MNO) has a role of an IDM provider. The basis of this authentication scheme is on the fact that a user is already authenticated in the GSM/UMTS network. Therefore the authentication process consists of the steps that make certain that the owner of the ME and the computer are the same. For verification a sessionID is send to both devices, to computer over the Internet and to the ME over GSM/UMTS network. The ME owner or the ME itself (automatically) matches the received sessionIDs with computers and sends a confirmation message via SMS to the authentication server. The Authentication procedure completes when the AS receives the confirmation SMS. Figure 20 shows the procedure which can be seen as following.

A service provider is client of the Authentication Server (AS), it outsources authentication to the AS.



**Figure 20:** SMS authentication with sessionID check

For automatic verification of sessionIDs by the ME a Bluetooth connection is required between the computer and the ME.

A mutual authentication is not provided by the described authentication scheme as it only authenticates the user to the service provider and the security of this scheme relies on the security of the underlying GSM/UMTS network which is much stronger in case of the UMTS network. This scheme does not have any explicit integrity protection as well. SMS forgery threat can be avoided by the GSM/UMTS security mechanisms. Still it is possible to spoof SMS sender address [3].

In case if the attacker controls one of the intermediate nodes between the user's computer and the service provider the following attack can be easily executed by an attacker. When the user starts the authentication process to the server provider, at the same time attacker also starts the authentication (supplying the identity of the victim) to the same or the other service provider that uses services of the same AS. The attacker blocks the original request of user so that it does not reach the intended service provider. It responds with a forged Request/identity to the user. When the attacker receives the sessionID through the Internet channel from the AS, he sends it to the victim. Now the user will receive two similar sessionIDs as one send by the attacker through Internet channel and the second one sent by the AS via SMS. The user (the ME in case of automatic verification) does not have any method to check that the received sessionIDs were actually issued by the AS to provide authentication service for the computer with different internet address and possibly for the different service provider. So the user will issue a confirmation SMS and the AS will reply to the service provider that the authentication was successful.

In case of the automatic sessionID verification by the Mobile Equipment (ME) using the Bluetooth connection for communication with the computer, it is possible for the attacker to authenticate to any service provider registered at the AS even without the user noticing it (unless the PIN is required to issue confirmation message), in case if the Bluetooth security is compromised. Still, the attacker would need to be in the close proximity to launch this attack.

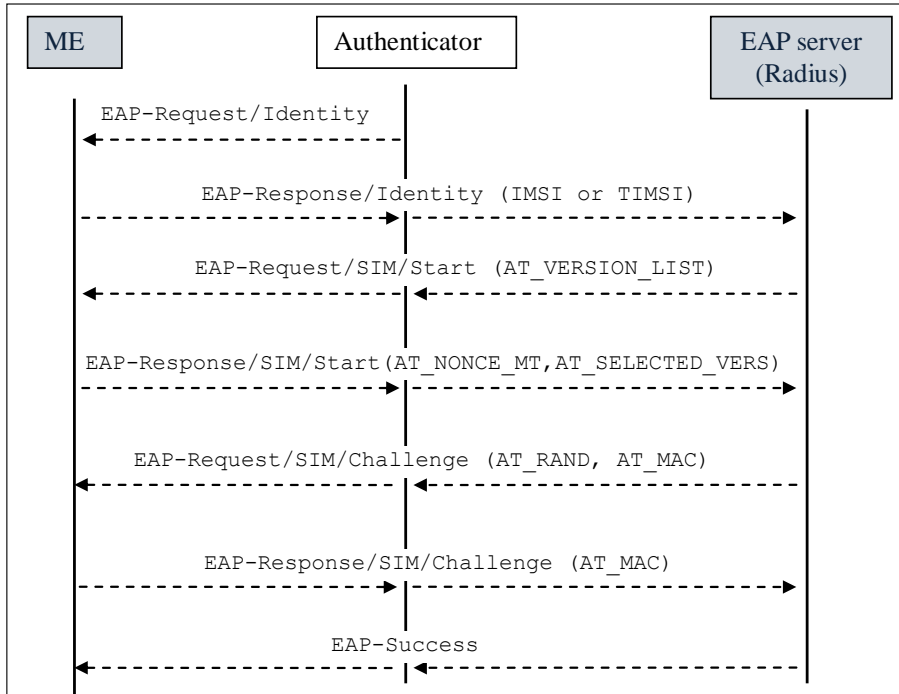
### 3.1.2 SIM Strong authentication

The SIM Strong makes use of the EAP-SIM protocol to provide mutual authentication [4]. In this method the MNO has a role of the IDM provider. Therefore the service provider depends on the MNO to perform authentication.

The EAP-SIM authentication scheme provides higher level of security as compared to the GSM authentication scheme. The EAP-SIM provides enhancements to the GSM Authentication and Key agreement (AKA) procedure which are discussed as following [80]:

- For deriving the Master key a 64-bit long GSM encryption key Kc is used which is not used directly.
- To create authentication responses and session keys of greater strength than the individual GSM triplets, the multiple authentication triplets combined for this purpose.
- The Master key is used to derive the Transient EAP Keys for protecting EAP-SIM packets, a Master Session Key for link layer security and Extended Master Session Key.

The EAP-SIM provides mutual authentication, integrity, confidentiality and replay protection.



**Figure 21:** EAP-SIM authentication

SIM Strong authentication can be run either through Internet and Bluetooth channel or via SMS channel by using the sessionIDs [4]. In case of Internet and Bluetooth SIM Strong authentication, the ME has Bluetooth connection with the computer and the EAP-SIM authentication communication is performed through the Bluetooth channel and Internet channel, not utilizing the GSM radio channel. The complexity and the number of the messages in the EAP-SIM exchange makes it impractical for the user to manually perform this exchange without using Bluetooth. Bluetooth security should be considered while using this scheme. By compromising the Bluetooth security the attacker has a chance to trick the ME to communicate with attacker's computer and perform the authentication for the attacker.

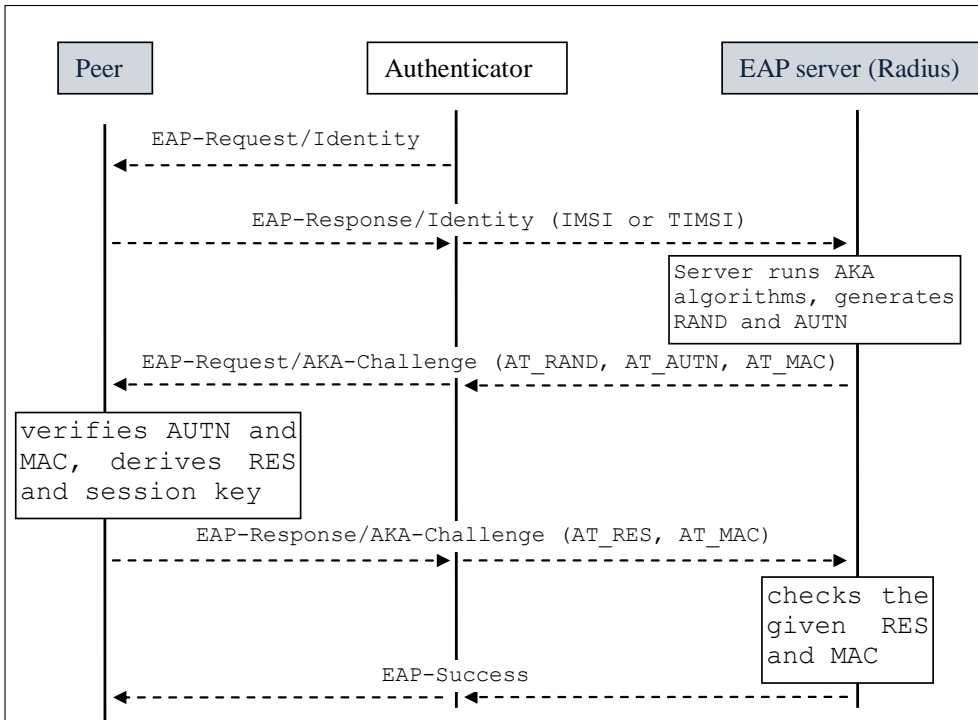
SIM Strong authentication that uses the SMS channel must also have the mechanism to make sure that the same user controls both the computer and the ME. This mechanism is done with sessionIDs. A sessionID is generated by the devoted applet that resides on the SIM card [3]. The generated sessionID can be entered by the user in the computer and transferred through Internet channel to the AS. After that the applet performs the mutual EAP-SIM authentication with the AS over the SMS

channel. The advantage of this scheme is that it does not require Bluetooth connection [4].

Both types of SIM Strong authentication need specialized applet on the SIM card to perform EAP-SIM [4].

### 3.1.3 EAP-AKA

As defined in RFC 5448, EAP-AKA is a mechanism for authentication and session key distribution that is used in UMTS Authentication and Key Agreement (AKA) mechanism. Unlike EAP-SIM which is based on the GSM AKA, this authentication protocol is designed to work with third generation networks. EAP-SIM and EAP-AKA have many common ideas as they were developed in parallel [97].



**Figure 22:** EAP-AKA authentication

As EAP-SIM, the EAP-AKA also provides mutual authentication, key derivation, confidentiality protection, integrity protection and optional identity privacy protection. EAP-AKA can also be used for SIM Strong authentication in the same manner as EAP-SIM.

## 3.2 One Time Password schemes

One of the secure methods of authentication is one time passwords (OTP). As the name suggests, OTP is used for authentication only once and after expiration of a session is useless. Therefore OTP is not susceptible to replay attacks. We can see the use of one time password (OTP) device in banks. Banks issue OTP tokens to their customers for online transaction authentication.

OTP use the concept of randomness. Randomness is an important factor because if passwords are not random then the attacker can guess the new passwords from previous observations. There are various methods used for OTP generation:

**Mathematical Algorithms:** We can use mathematical algorithms which can generate passwords



from the previous data. One method proposed by Leslie Lamport [81] uses one-way hash functions. It works by starting with generating an initial seed value. Then a set of one-time passwords is generated by applying a hash function multiple times to the initial seed. Every next password is generated by taking the previous passwords as seed and hashing it. The security of this scheme is based on the non-reversibility of the hash function. In this technique we have to change the seed when the set of password generated from one seed exhausts.

**Counter-based:** the scheme is based on the ever increasing counter and a secret shared by the token and the AS. RFC 4226 describes a counter-based scheme that uses HMAC-SHA-1 function to generate OTP value.  $HOTP(K, C) = \text{Truncate}(HMAC\text{-SHA-1}(K, C))$ , where K is a shared secret key and C is a counter value [82]. Truncation is used to enable the user to easily enter the resulting value in computer. It is important for the counter to be synchronized between a token used by the user and the authentication server.

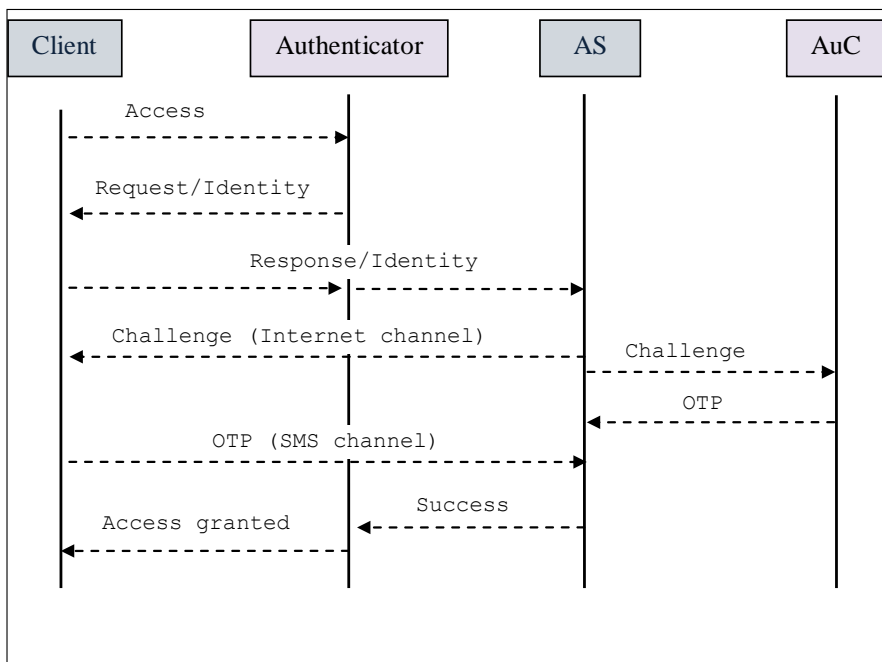
**Time-based:** This approach is based on the time synchronization between Authentication Server and the user who provides password for his authentication. The user is given some device which needs to be synchronized with the server to generate time-based OTP (TOTP). Usage of time as an input parameter ensures that OTP values do not repeat. As described in [83], the TOTP value can be generated by hashing the shared secret and a time parameter. Time synchronization is crucial for the TOTP scheme.

**Challenge Approach:** this scheme is based on randomly challenge values and shared secrets. In this method a verifier gives a randomly generated challenge to a party that wants to be authenticated. Based on the shared secret and the received challenge value the OTP password is generated. OATH Challenge-Response Algorithms, described in [84], provides one-way or mutual authentication based on the challenge-response concept. The advantage of this scheme is that parties do not need to maintain synchronization of any values.

### 3.2.1 One Time Password from PC to SMS

This scheme is a multi-channel challenge-based OTP authentication system. It is a one way authentication scheme. Only the user is authenticated to the service provider. The following exchange take place between the client and the service provider as can be seen in the Figure 23.

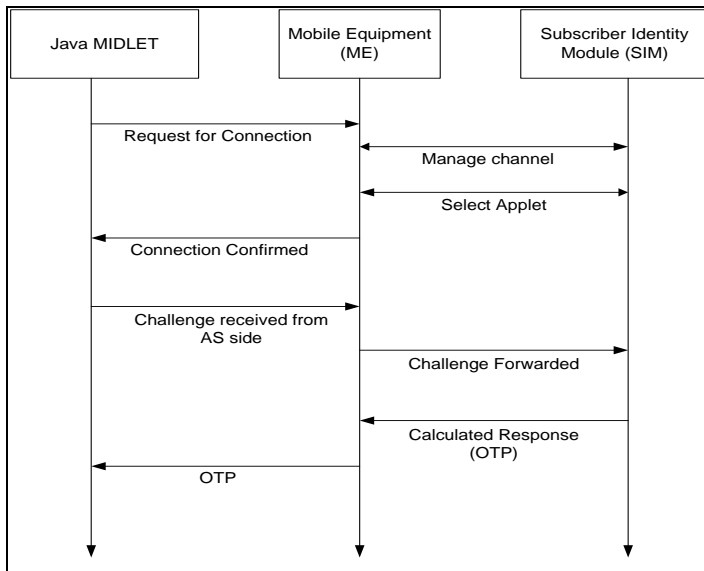
- Client requests for access to the service provider (SP).
- The service provider asks to prove his identity.
- The client writes his username in the browser and sends it to the service provider which forwards to authentication server (AS) for authentication process.
- When the Authentication server receives the credentials of the client it generates a challenge which can be a random number taken from the client data stored on the server side.
- It send the challenge to the client through Internet channel
- When the client receives the challenge he enters the challenge on the mobile phone.
- There is an OTP applet in the SIM. When the challenge is entered the OTP applet calculates the OTP.
- The OTP is send to the Authentication server through SMS.
- When authenticator receives the OTP it compares it with the one it generated itself.
- If both are same the client is authenticated and now the service provider can provide the required services.



**Figure 23:** OTP from PC to phone authentication [3]

When the user receives the challenge from the AS via Internet channel he can either enter this value to the ME manually, or a Bluetooth connection between the computer and the ME can be used to transfer challenge. The process of generating OTP on mobile consists of the following steps which are also summarized in the Figure 24.

- There is Java MIDlet installed on the ME.
- The user can start the MIDlet manually or, if we want the MIDlet to interact automatically then we can use Wireless Manager API for this purpose. The MIDlet communicates with the SIM using SATSA-APDU as can be seen in the figure given below.
- The challenge that is entered is communicated to the OTP Applet which generates OTP from the challenge.
- The generated OTP value is sent through SMS to the AS either manually or automatically using Wireless Manager API.



**Figure 24: OTP Applet [3]**

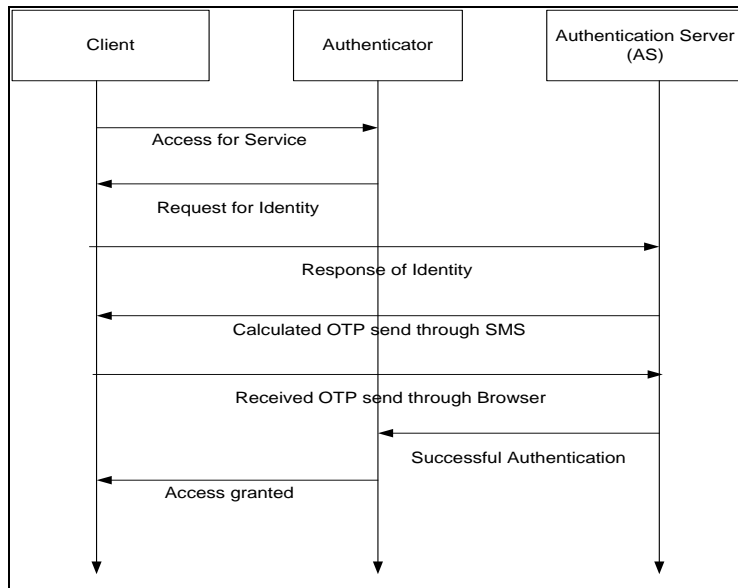
The OTP PC to SMS scheme requires a dedicated OTP generation application on the SIM card. By sending the OTP value via GSM/UMTS SMS channel the user assures AS that he controls both the computer and the ME. The security is based on the assumption that only the legitimate subscriber's SIM can generate OTP and on the fact that the radio channel is encrypted, making it almost impossible to intercept SMS and extract OTP value.

This scheme neither provides confidentiality protection nor temporal key derivation. Besides, since this scheme does not provide mutual authentication and integrity protection it is susceptible to many attacks. As in SMS authentication with sessionID check scheme, it is also possible to launch session hijacking attack. This attack can be easily executed if the attacker controls one of the intermediate nodes between the user's computer and the service provider. When the user starts authentication to the server provider the attacker also starts authentication (supplying the identity of the victim) to the same or the other service provider that uses services of the same AS. The attacker blocks the original request from the user so that it does not reach intended service provider, and responds with a forged Request/identity to the user. When the attacker receives a challenge via Internet channel from the AS, he "forwards" it to the victim. The user will receive the challenge from the attacker. The computer/user/ME has no way to check that the received challenge was actually issued by the AS to provide authentication for the user's session. Believing that this challenge was intended for him the user will unblock OTP generation function by supplying the PIN to the ME, the OTP value will be generated and sent to AS. AS will check whether the received and self-generated OTP values match and will authenticate attacker.

If Bluetooth connection is used between the ME and the computer, it should be well protected. The attacker can authenticate to any service provider registered at the AS even without the user noticing it (unless the PIN is required to issue confirmation message), if the Bluetooth security is compromised.

### 3.2.2 One Time Password from SMS to PC

In this architecture we consider that the user with mobile phone is authenticated by GSM/UMTS network. The main difference between this architecture and the previous architecture is the processing done at the server side, making user free from processing tasks. The steps involved in this architecture can be seen in the figure below



**Figure 25:** OTP SMS to PC authentication [3]

The authentication exchange consists of the following steps:

- User requests Access to some service provided by the service provider (SP)
- The Service provider asks for user's credentials for authentication
- The user provides the username which is forwarded to the AS
- The AS generates the OTP value
- The generated OTP is forwarded to the client mobile through the SMS channel
- The Client inserts the OTP received through SMS either manually or automatically using Bluetooth technology
- If the OTP generated at the server side and received from the user are same than the user is authenticated.

In the automatic version of this architecture we use java applet to communicate with SIM using SAP. When the mobile will receive the SMS it can notify the Java applet running on client's computer and the SMS can be retrieved from the mobile.

The OTP SMS to PC scheme utilizes the fact that the user is already authenticated in the GSM/UMTS network. Thus the AS needs to confirm that the owner of the ME actually controls the computer. This is done with OTP exchange. Although the OTP value is sent to the user's ME via SMS channel, it is not used to provide mutual authentication. The only difference between the randomly generated sessionID that could be used and the OTP value is that the latest is actually associated with the HTTP session created by the user [3]. To do session hijacking the attacker needs to send OTP value from his computer. However, the attacker cannot intercept this value on the radio channel since this link is protected by GSM/UMTS security mechanisms. Only if Bluetooth is used between the ME and the computer can the attacker obtain OTP value by compromising the Bluetooth security. Though there is no need for the attacker to do this. By intercepting the OTP value, sent back to the AS via Internet channel, and sending it from his computer attacker can hijack the session. Since the OTP value is bound to the HTTP session, the attacker cannot authenticate to arbitrary service provider if the user's computer performs OTP check.

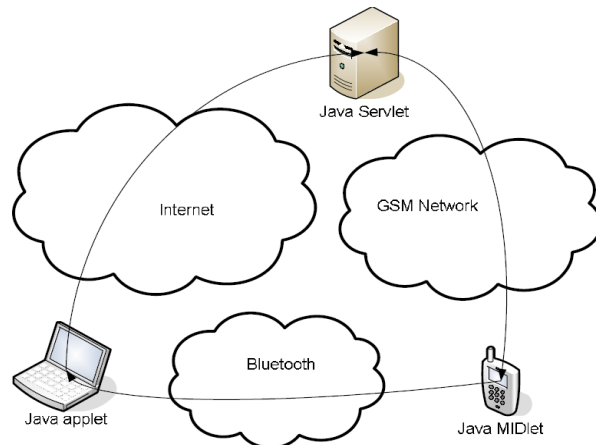
If Bluetooth connection is used between the ME and the computer, it should be well protected. The

attacker can authenticate to any service provider registered at the AS even without the user noticing it (unless the PIN is not asked to authorize the transfer of OTP value to the computer), if the Bluetooth security is compromised.

Session hijacking is possible since this scheme does not provide integrity protection. It also does not provide confidentiality protection and temporal key derivation mechanism.

### 3.2.3 Enhanced OTP from PC to SMS authentication

This architecture, described in [5], is an enhanced version of the multi-channel challenge-based OTP from PC to SMS solution, which provides integrity protection. The components architecture is shown in the diagram given below:



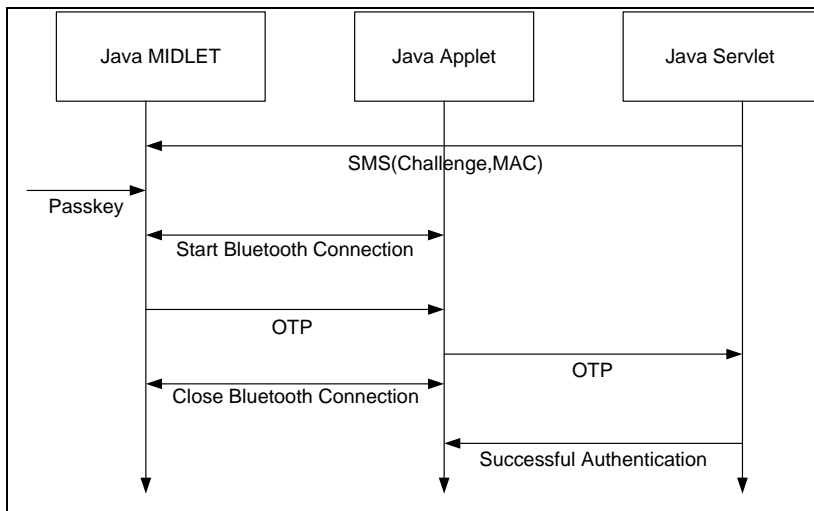
**Figure 26:** Component Architecture [5]

The steps involved in the authentication process, shown in Figure 27, are as follows:

- When the user wants to get some service he accesses the server, presents his user name to the Java Servlet present at server side.
- The server generates a challenge and it computes the OTP value in the following way:  

$$\text{OTP} = \text{hash}(\text{challenge} \parallel \text{secret key})$$
- The message authentication code (MAC) is generated on OTP.
- The challenge and the MAC are sent to the MIDlet through the SMS channel.
- Upon receipt of the SMS, the MIDlet in the ME is activated and asks for user password. When the user enters the password it is hashed and then compared with the stored value.
- After checking the password, the password is used to decrypt the secret key.
- The challenge received is concatenated with the secret key and the result is hashed to generate OTP at the MIDlet side.
- Message Authentication code (MAC) is generated on OTP and matched with the MAC received through SMS message. If it is different the procedure is aborted.
- On successful match the MIDlet start communicating with the Java applet running on the computer through Bluetooth. The OTP value can be entered in the computer manually without the usage of Bluetooth, though it should be truncated for usability.
- The Java applet sends the OTP value to the Java servlet via Internet channel, and the connection between Java MIDlet and Java Applet is aborted.

- The server compares the self computed OTP with received OTP. If they match the user is authenticated.



**Figure 27:** Authentication process

**Java MIDlet:** It is one of the important components of the ME as it gives the capability of OTP generation and communication with the Java Applet and Authentication server. Java MIDlet can be downloaded by registering it on the website. After registration a push message is send to the user which enables him to download and install it. When the MIDlet is installed, keys are exchanged between authentication server and the MIDlet at the ME through SMS. After the MIDlet completes key exchange, it starts working.

There are 2 passwords used for authentication. A PIN code selected by the user (used to provide two factor authentications) and the other one is the OTP, which is generated by the Server and shown on the web page when the Java MIDlet is downloaded. This initial OTP is used to authenticate the key exchange.

The wireless Messaging API (WMA) enable the MIDlet in the ME to send and receive SMS messages. The SMS communication can be explained in the following steps:

- The MIDlet is registered with a port and protocol in such manner that if the SMS message is received at the port, the Application Management Software (AMS) forwards it to the MIDlet. The registration with the port can be done statically by using the Java ME application descriptor (JAD) file.
- At the server side the SMS message is sent at the specified port of the mobile using the desired protocol.
- When the AMS receives the SMS message at specified port it forwards it to the MIDlet as MIDlet is listening at the specified port.

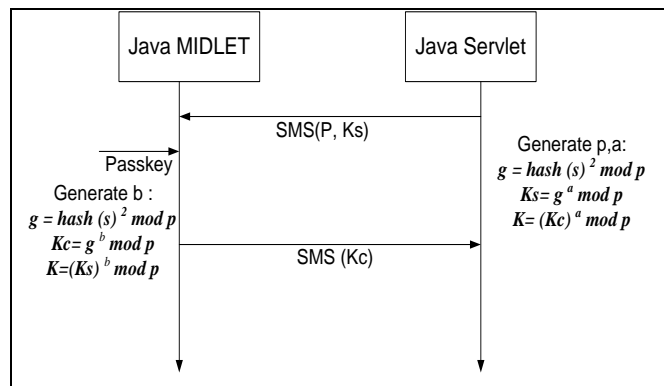
**Java Applet:** It serves as server for MIDlet and helps MIDlet to connect with the computer through Bluetooth. For communication between Java MIDlet and servlet, established Bluetooth connection should be present. A 16 byte pass phrase is used while connecting through Bluetooth to keep the Bluetooth connection protected. When the applet receives the OTP it sends it to the servlet at the server where it is matched with the computed OTP.

**Java Servlet:** It acts as an authentication server to clients. When the user downloads the MIDlet it

is registered with the Servlet. At the registration time the key exchange takes place to compute shared secret key  $K$  at the MIDlet and the servlet side. The key is saved with the user profile in the database. The servlet can be configured to send SMS at a specific port so that MIDlet can receive it on that port. The servlet asks for the user name when the user is authenticated for service. It creates a challenge and sends it to the MIDlet through SMS. The MIDlet computes the OTP and sends it to the servlet with the help of java applet that is connected with the MIDlet using Bluetooth.

At the beginning (after the MIDlet was downloaded) the Java MIDlet on the ME and the AS do not have shared secret to generate and check OTP values. The shared secret is derived via key exchange procedure. However, before the shared key is derived, the authentication procedure should be performed. When downloading a MIDlet the user is shown the OTP value on the server's web page. The user enters this OTP value to the MIDlet only once. This OTP value is used to authenticate the key exchange [5]. For the key exchange the Simple Password Exponential Key Exchange (SPEKE) protocol, which is an improved version of Diffie-Hellman, is used [5].

The key exchange procedure can be seen in the figure below



**Figure 28:** Key Exchange procedure [5]

The key exchange steps are the following [5]:

- At the server side a large random prime  $p$  is generated. Then value  $g$  is calculated as  $g = \text{hash}(s)^2 \bmod p$ , where  $s$  is short OTP which is displayed in the browser after registration. The server also computes  $Ks$  using the following equation:
- $Ks = g^a \bmod p$ , where  $a$  is the secret random number.  $Ks$  and  $p$  are sent to the MIDlet via SMS channel.
- At the MIDlet side the following values are computed:  $g = \text{hash}(s)^2 \bmod p$ ,  $Kc = g^b \bmod p$ , where  $s$  is the short OTP as in first step and entered by user;  $b$  is the secret random number. The computed  $Kc$  value is send to the server through the SMS message. Then the MIDlet computes the shared secret key  $K$  as:  $K = (Ks)^b \bmod p$
- After receiving  $Kc$  the server computes the shared key  $K$  as:  $K = (Kc)^a \bmod p$ .

Now both the servlet and the MIDlet share the same secret key  $K$ , which will be used to generate OTP. The key  $K$  is further encrypted using user selected password, so that if mobile is lost no one can take benefit of it unless the password is known. The key is then stored in the encrypted form in the Record store in J2ME. In the MIDlet a hash of user password is stored so that it can verify the correctness of the password entered by the user at the start of the procedure.

It is important to note that the user does not need to prove his identity, though he needs to prove that he controls both the computer and the ME. The security relies on the fact that the ME with

subscriber's UICC card is already authenticated by the mobile network operator. The OTP value that is used as the input to the hash function ensures that the user, with whom the key exchange via SMS is performed, is the same user that requested the service from the server (it means that the user controls both the computer and the ME). Only the legitimate subscriber could receive SMS messages, decrypt them, and compute shared secret K based on the initial OTP value and the values sent in SMS. The GSM/UMTS network provides encryption of the radio channel and ensures that SMS is forwarded to the destined receiver.

In the subsequent exchanges the user proves that he controls the computer by sending the OTP value, generated by the ME based on the shared secret and the challenge sent via SMS, via the Internet channel to the AS. Along with the challenge the MAC value (computed on OTP) is sent to the user via the SMS channel. After computing the OTP value, the user can check whether the challenge was sent by the party that knows the shared secret key derived during the key exchange phase. Only the party that knows shared secret could compute OTP value and calculate the MAC value of the OTP. Thus the user is ensured that the challenge is sent by the same server that displayed the initial OTP value on the web page. However, it is still not the mutual authentication, since the user does not check the identity of the server when making initial HTTP request to the server, thus masquerade attack is possible. The attacker can intercept the initial HTTP request, respond with the initial OTP value in the HTTP response to the user, go through the key exchange phase and derive the shared secret for OTP generation. This scheme provides only integrity protection. The initial OTP value cannot be used to authenticate the server (even if it were based on the secret shared by the server and the user) since it is transmitted openly to the user, and the attacker can intercept it.

Since the shared secret for the OTP generation is dynamically derived by the Java MIDlet, it is not mandatory for the MNO to act as IDM provider. The server can execute SMS exchange with the user directly. The advantage of this scheme is that the dedicated applet on the UICC card is not required, however the shared secret key used for OTP generation has to be stored in the encrypted form in the Java Record Store.

### **3.3 Proposed Solution for Mobile Based Linux Workstation Logon Service**

To design a solution for secure mobile based authentication for Linux workstation logon, it is required to make the solution more secure, user friendly and supported by the existing technologies. At the client (user) side a Pluggable Authentication module (PAM) is used to make the solution technology specific free. By using the PAM API and the functions we can access the user information and authenticate user at the server without taking care of the technology at the server side like LDAP, Kerberos etc. The PAM API converts the required messages to the technology.

The user information is saved at the authentication server. We require the user name and the mobile number so that we can send the One Time Password (OTP) to the user as SMS. The user information is accessed using the *passwd* structure. The *passwd* structure has the following information accessed from the server.

- User name
- User password
- User id
- Group id
- Real name
- Home directory
- Shell program

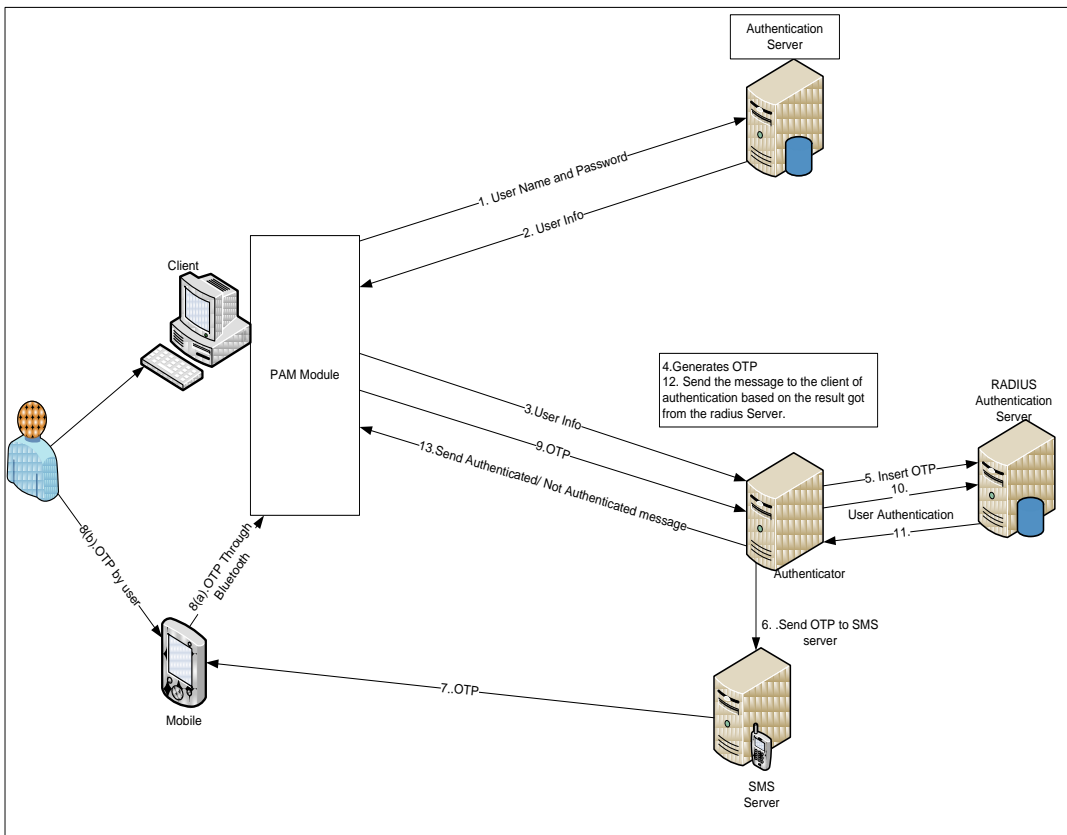


For getting the mobile number of the user we have suggested the use Real Name field to be used as Real Name, mobile number e.g. George, 004712345678. We will access the Real name field and get the mobile number from it as well.

After having the user authenticated from the existing technology and having the user information, we can pass it to the Mobile Based Authenticator that generates the OTP, send it to the user as SMS and insert the OTP in the RADIUS server against the user information. The communication between the client computer and Authenticator will be done by using socket communication. The Authenticator serves as the connection point for the client and the RADIUS Authentication server.

When the user will receive the OTP as SMS, they can insert it in the client computer manually or by using Bluetooth technology. The password will be passed to the Authenticator which will forward it to the RADIUS Authentication Server for authentication process. The Authenticator will send the message authenticated if the RADIUS server passwords matches with the OTP entered by user. Otherwise user will not be authenticated.

The process of the solution can be seen in the figure given below:



**Figure 29:** Mobile Based Authentication System for Linux Workstation Logon

The steps involved in the process are as following

1. The user enters name and static password which will be forwarded to the Authentication Server like LDAP, Kerberos Server etc.

2. The user will be authenticated by the static password which the user will have. After authentication the user information will be accessed and returned to the client.
3. At the client side the PAM module will check the user information. If it finds the mobile number in the Real Name field of *passwd* structure then it will continue with the second authentication process based on mobile. Other wise it will authenticate user based on the first authentication process. This option will give administrator the choice of allowing some users to use only the existing authentication technology and others using mobile based authentication technology.
4. The Authenticator serves as the connection point for the PAM module and the RADIUS Authentication server. It will generate OTP for the user.
5. The OTP will be inserted in the database against the user information in the RADIUS Authentication Server.
6. The OTP will be sent to the SMS server with the mobile number, so that it can be send to the user.
7. The SMS server will send the OTP to mobile phone.
8. Upon receiving the OTP the user will have the option of transferring the OTP from mobile phone to the client computer in following ways:
  - a. Using Bluetooth.
  - b. By user reading from mobile and entering on the Mobile OTP page on client
 The manual option will help for those user who do not have the Bluetooth capability in their mobiles.
9. The client will send the OTP to Authenticator for Authentication.
10. The Authenticator will forward the OTP and user name to the RADIUS Authentication server for Authentication process.
11. The RADIUS Authentication Server will send the result of Authentication process by comparing the OTP and the password of the user in the data base.
12. The Authenticator will forward the Authentication process result to the PAM module.
13. The PAM module will authenticate the user if user is authenticated by the RADIUS Authentication Server, otherwise the user will not be authenticated.

The solution which is proposed for the mobile based authentication for Linux workstation logon process will make the logon process more secure giving the administrator the option to select the users with their authentication method. The fact that the solution can be used with the existing technology and the infrastructure is already available, makes the solution less costly. Thus this makes the solution more suitable.

## 4. Analysis

This chapter consists of system requirements, Use cases, interaction diagrams and a proposed solution for secure mobile authentication for Linux log on, to specify the required functionality, components and the interfaces of the system.

### 4.1 Requirements

The requirements define the characteristics or features of the desired system. Those features which the system must perform are called *Functional requirements* and those features which are relevant to the system performance are called *Non functional requirements*.

#### 4.1.1 Functional Requirements:

The functional requirements of the system are as following

- Secure authentication of user for Linux workstation Log on in an enterprise, using mobile phone.
- A support for mutual authentication should be added.
- The system should support two factor authentications.
- A support for existing authentication schemes should be added.
- The password used during mobile phone based authentication should be strong enough to protect against dictionary attack.
- Sensitive data should be protected.
- The passwords used with mobile phone authentication should be used for once. (One Time Passwords)

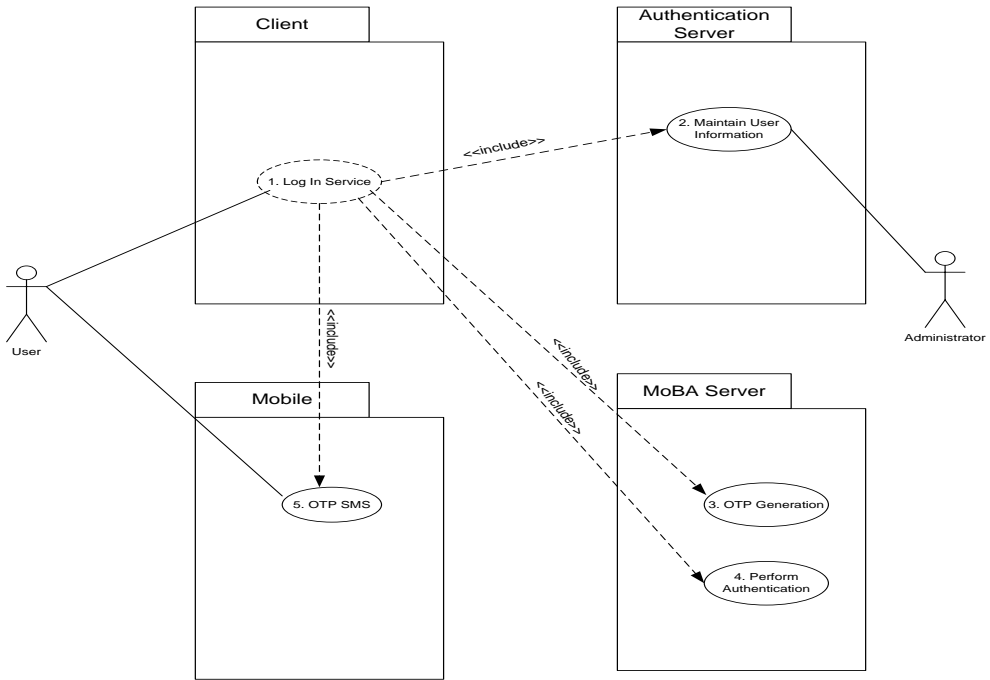
#### 4.1.2 Non Functional Requirements:

The non functional requirements of the system are as following:

- **User Friendly:** The proposed solution should be user friendly.
- **Cost:** The cost of deployment should not be more than the existing solutions. Cost of the solution should be less.
- **Scalability:** With increasing number of users the system should be easily scalable.
- **Performance:** The performance of the proposed solution should be good. The delay in the authentication should not be more than the existing systems.
- **Availability:** The system should be available all the time when the user requires it.
- **Reliability:** The system should be reliable. The percentage of error occurring should not be more than the expected value or existing systems.

### 4.2 Use Case Diagrams:

The use case diagrams are used for capturing the basic functional requirements of the system. Following are the general use case diagram in which functional requirements are defined



**Figure 30:** General Use Case Diagram

Figure 30 shows the overall overview of the system. The tables given below explain the main scenarios

**Table 12:** Use Case-Login Service

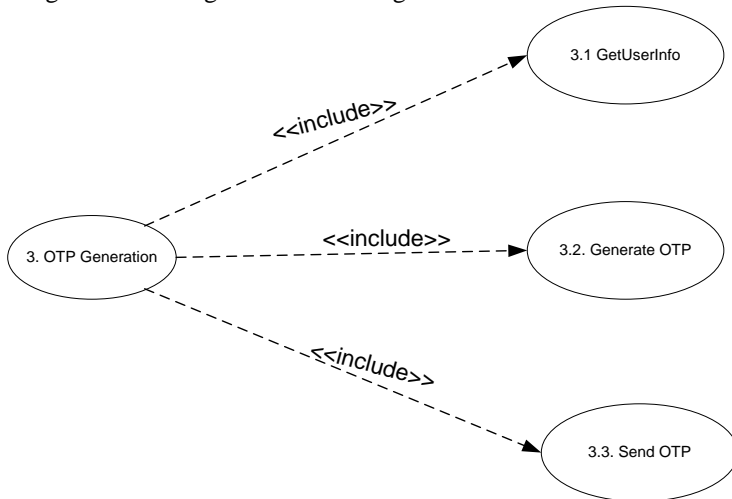
Use Case	Login service
<b>Description</b>	The user wants to login securely using mobile based Authentication (MoBA).
<b>Actors</b>	User Client Authentication Server MoBA Server Mobile
<b>Assumptions</b>	1. The client is connected with Authentication Server and MoBA Server. 2. The MoBA Server is connected to mobile phone using some wireless technology like GSM or UMTS. 3. The user provides the correct credentials.
<b>Steps</b>	1. The user enters the username. 2. The user gets the user information from Authentication server (Starts UC2). 3. The user information is then forwarded to MoBA Server (Starts UC3) 4. The OTP is generated and sent to user mobile phone. (Starts UC5). 5. The OTP is sent to MoBA Server by user or through Bluetooth from mobile to client and then to MoBA Server (Starts UC4). 6. And after that on the bases of OTP the MoBA server decides for authentication of user.
<b>Variations</b>	1. The user enters incorrect username. 2. The user enters the incorrect password.
<b>Issues</b>	

**Table 13:** Use Case-Maintain User Information

Use Case	2. Maintain User Information
<b>Description</b>	The user information is saved at the Authentication Server and the administrator has the right to assign rights to other users and can manipulate the users information
<b>Actors</b>	Authentication Server Administrator
<b>Assumptions</b>	1. The user can access his information. 2. The administrator can add new users and assign rights to users.
<b>Steps</b>	1. The Authentication Server receives the user name from client. 2. The Authentication Server fetches the data of the user in the database. 3. The data of the user is send back to the client
<b>Variations</b>	The information of user not found in the database.
<b>Issues</b>	

**Use Case 3: OTP Generation:**

The details of OTP generation are given below in Figure 31:



**Figure 31:** OTP Generation Use Case

The details of the above use case are explained in the table 14, 15, 16, 17, 18 and 19.

**Table 14:** Use Case-OTP Generation

<b>Use Case</b>	<b>3. OTP Generation</b>
<b>Description</b>	The client requests for password so that user can be authenticated.
<b>Actors</b>	Client MoBA Server
<b>Assumptions</b>	1. The user has entered valid username. 2. Data for user is found in the database. 3. The data is forwarded to the MoBA Server.
<b>Steps</b>	1. The client enters the username. 2. The data from database is retrieved for the username and sent to the MoBA Server. 3. The MoBA Server generates the OTP. 4. The OTP is forwarded to the user mobile phone using SMS service.
<b>Variations</b>	User enters invalid username.
<b>Issues</b>	

**Table 15:** Use Case-Get User Information

<b>Use Case</b>	<b>3.1. Get User Information</b>
<b>Description</b>	The MoBA receives the user information from the database in the Authentication server through client.
<b>Actors</b>	Authentication Server Client MoBA Server.
<b>Assumptions</b>	1. The Authentication server is connected to the client 2. The Client is connected to the MoBA Server.
<b>Steps</b>	1. The client receives the user information from the database in the Authentication server. 2. The client forwards it to the MoBA Server 3. The MoBA retrieves mobile number and other information from the data.
<b>Variations</b>	Invalid username entered and data cannot be found in the database.
<b>Issues</b>	

**Table 16:** Use Case-Generate OTP

Use Case	<b>3.2. Generate OTP</b>
<b>Description</b>	A One Time Password (OTP) is generated by MoBA Server.
<b>Actors</b>	MoBA Server
<b>Assumptions</b>	The valid username is provided to the MoBA Server
<b>Steps</b>	The MoBA Server generates the OTP.
<b>Variations</b>	
<b>Issues</b>	

**Table 17:** Use Case-Send OTP

Use Case	<b>3.3. Send OTP</b>
<b>Description</b>	The MoBA Server sends the One Time Password (OTP) to the user mobile
<b>Actors</b>	MoBA Server Mobile
<b>Assumptions</b>	1. The MoBA Server is connected to the mobile through some wireless technology like GSM or UMTS. 2. The user information is valid.
<b>Steps</b>	The MoBA sends the OTP to the user mobile.
<b>Variations</b>	If the user information is invalid
<b>Issues</b>	

**Table 18:** Use Case-Perform Authentication

Use Case	<b>4. Perform Authentication</b>
<b>Description</b>	The user can be authenticated by the valid OTP received through mobile and sending it to the MoBA Server.
<b>Actors</b>	MoBA Server
<b>Assumptions</b>	The user provides the valid password.
<b>Steps</b>	1. The user inputs the password received on mobile phone. 2. The MoBA server compares the OTP's and makes the decision.
<b>Variations</b>	The user enters invalid password.
<b>Issues</b>	

**Table 19:** Use Case-OTP SMS

Use Case	5. OTP SMS
<b>Description</b>	The mobile receives the OTP by SMS and it is forwarded to the client by user or through Bluetooth.
<b>Actors</b>	Mobile. Client. User.
<b>Assumptions</b>	In case of Bluetooth the connection should be established between the client and the mobile phone.
<b>Steps</b>	1. The mobile phone receives the OTP by SMS. 2. The OTP is forwarded to the client through Bluetooth or by user.
<b>Variations</b>	
<b>Issues</b>	

### 4.3 Interaction Diagrams:

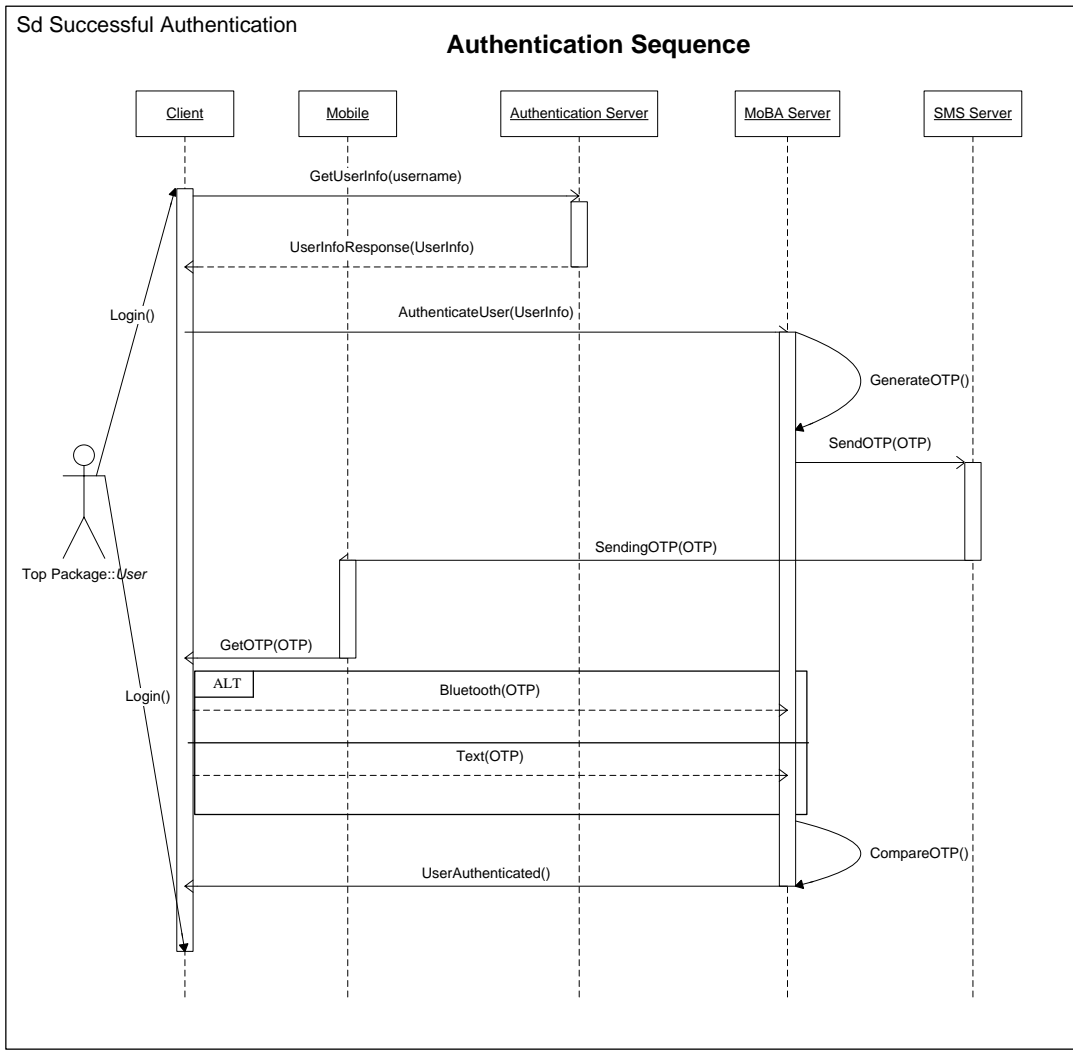
The interaction diagram show how the different objects will interact with each other. The messages it will pass to each other and the information. One of the commonly used interaction diagram is sequence diagram which is very useful in showing the passing of messages. The sequence diagram of the successful authentication is shown in the below given diagram.

#### 4.3.1 Successful Authentication

The steps involved in the successful authentication are discussed as following:

- i. The User enters the username for login in the computer system (Client) in an enterprise.
- ii. The user information is forwarded to an authentication server to fetch the user information from the database in the Authentication Server
- iii. The user information is returned to the client, which is forwarded to the Mobile based Authentication server (MoBA). The MoBA server serves as Authenticator as well as the Authentication server.
- iv. The MoBA Server generates an OTP and forwards it to the SMS Server.
- v. The SMS server sends the OTP to the mobile through SMS message.
- vi. Upon receiving the OTP message at the mobile phone either the user can type the OTP or it can be forwarded to the client using Bluetooth technology.
- vii. The client forwards the OTP to the MoBA Server.
- viii. The MoBA server compares the OTP received from the client with the generated one.
- ix. If the OTP's match the user will be authenticated and will be given right to login in the system.





**Figure 32:** Sequence Diagram- Authentication

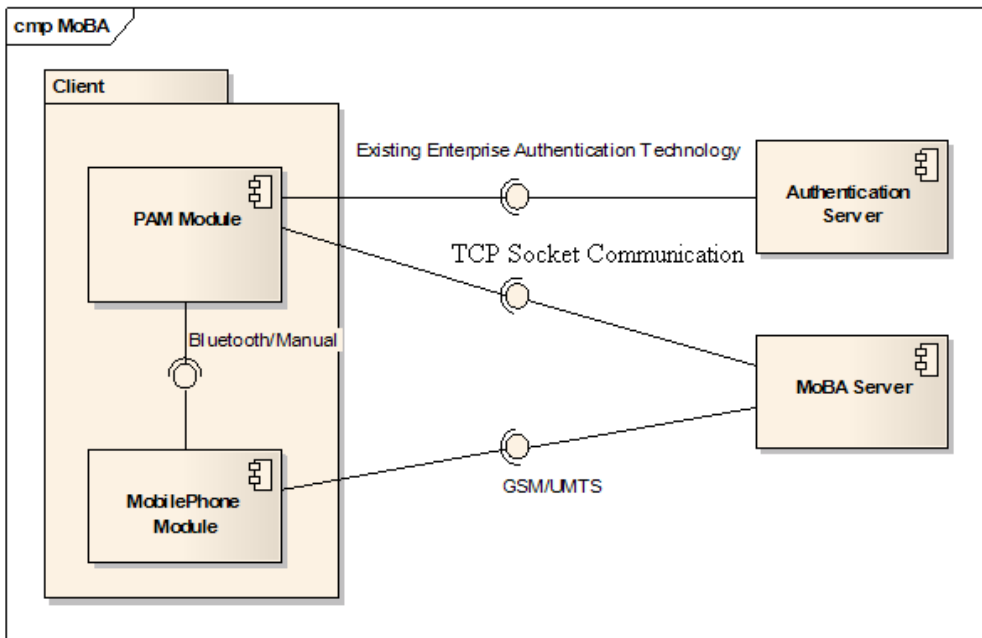
## 5. Design

The design chapter will identify the components, packages and classes of the system that the system will be composed of.

### 5.1 Components:

Components are modular parts of the system [85]. They are the individual pieces which are independent from each other. The behaviour of the components is defined by the provided and required interfaces.

The Figure 33 shows the main components of the system. The client package is added for showing the clarity in the Figure 33. There are four main components as can be seen in the given figure. The client side consists of the PAM module and the Mobile phone component, while the server side has been divided in to two portions. The Authentication Server has the user database and is controlling the existing technologies for authentication like LDAP, Kerberos etc. The other part Mobile Based Authentication (MoBA) Server handles the One Time password (OTP) and doing authentication on the basis of the OTP. The detail of each component is explained in this chapter.



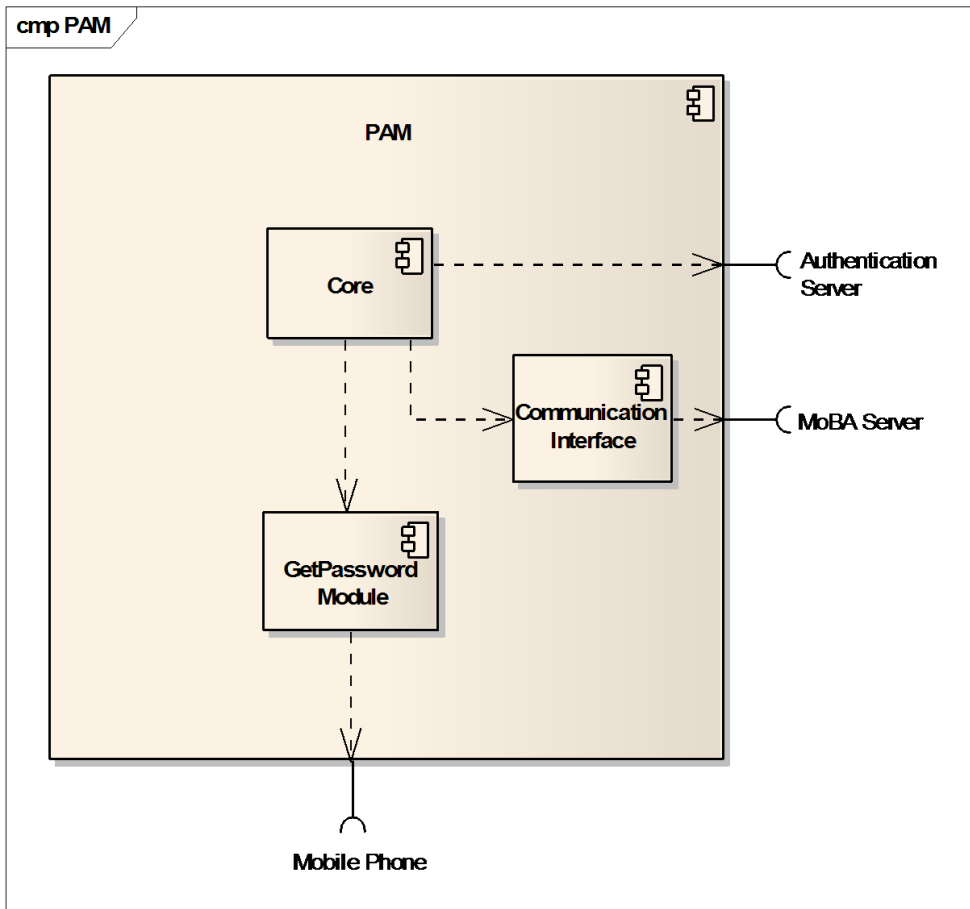
**Figure 33:** Main Component Diagram.

The PAM module at the client side communicates with the Authentication Server with the existing technology messages. It communicates with the Mobile Phone component with Bluetooth or manually as chosen by the user. The Mobile phone component has interface with MoBA Server for communication of OTP through SMS. The PAM module has also communication interface with MoBA Server for communication.

#### 5.1.1 PAM Module:

The PAM module handles the login service at the client side. It communicates with the existing

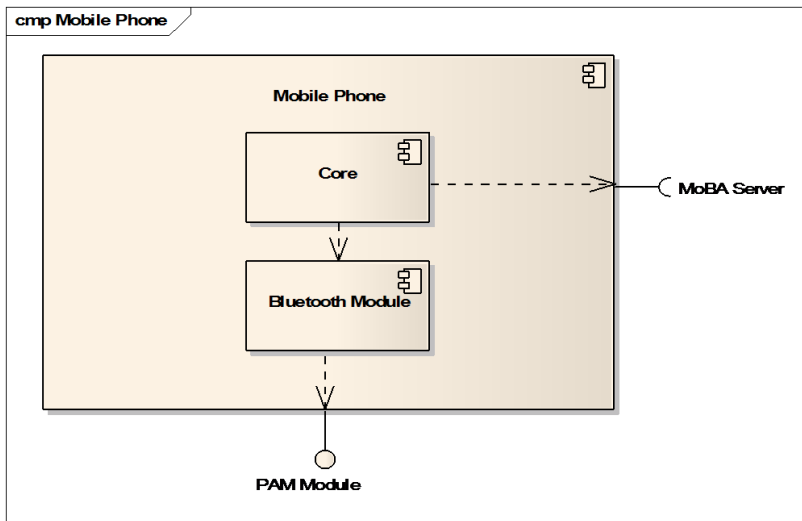
Authentication server to get the user information. For Authentication of user and for OTP it communicates with MoBA server. The PAM Module is divided in to 2 components; the GetPassword module accepts the password from user either through Bluetooth or manually depends on user choice. The communication interface controls the communication with the MoBA server for OTP and authentication.



**Figure 34: PAM Module Component Diagram.**

### 5.1.2 Mobile Phone Module:

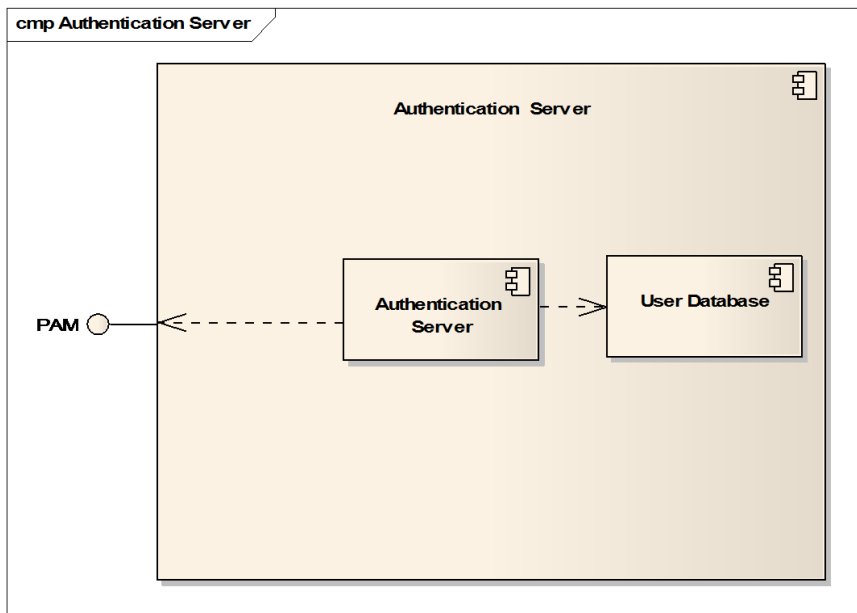
The main task of this component is to receive the OTP through SMS. The detail of the components can be seen in the Figure 35. It communicates with the MoBA server for OTP. It has Bluetooth Module which task is to forward the OTP through the Bluetooth to the PAM Module as it has the interface available for this task. After receiving the OTP it will be the user choice to forward the OTP either manually or through the Bluetooth.



**Figure 35:** Mobile Phone Component Diagram.

### 5.1.3 Authentication Server (AS):

The Authentication Server (AS) performs the authentication of users with the existing Authentication technology like LDAP, Kerberos etc. The detail of the Authentication server can be seen in Figure 36. It has user database which has all the information required for the authentication. It has an interface with the PAM module for communication.

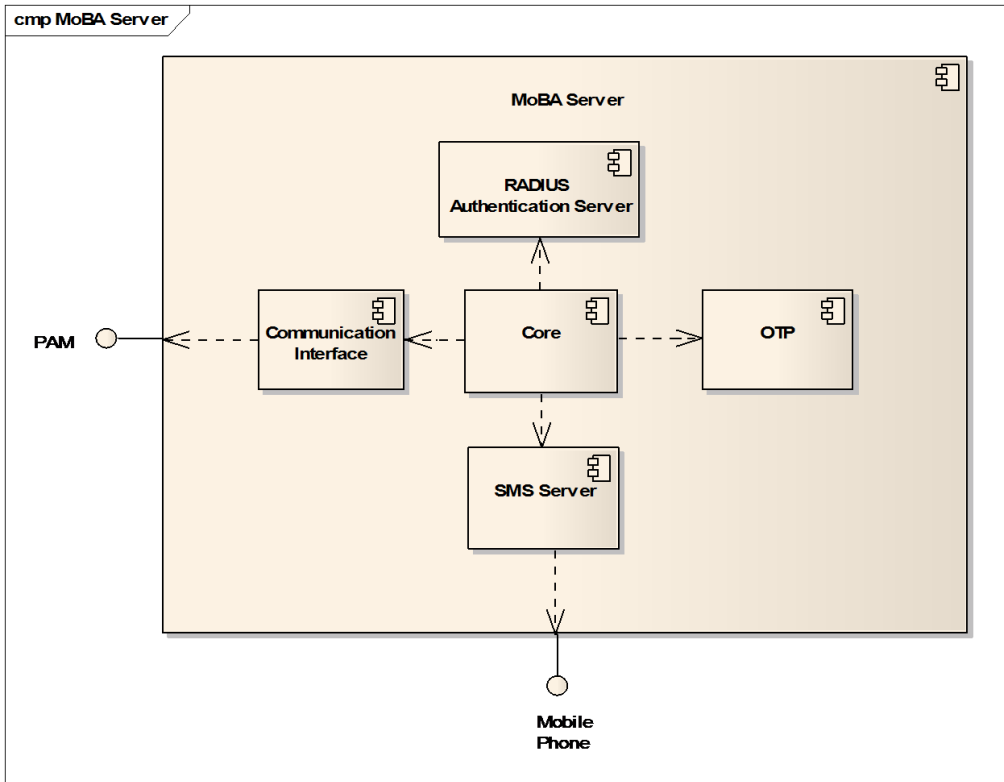


**Figure 36:** Authentication Server Component Diagram.

### 5.1.4 MoBA Server:

This is the main component of the system. It handles the generation of OTP and authentication of

user based on that OTP. There are four main components which can be seen in the figure given below. The communication interface handles the communication with the PAM module. The OTP handles the generation of the OTP. The RADIUS Authentication Server handles the Authentication of user based on the OTP. The SMS server handles the communication with the SMS server and SMS communication with the Mobile Phone module as it has an interface attached with it.

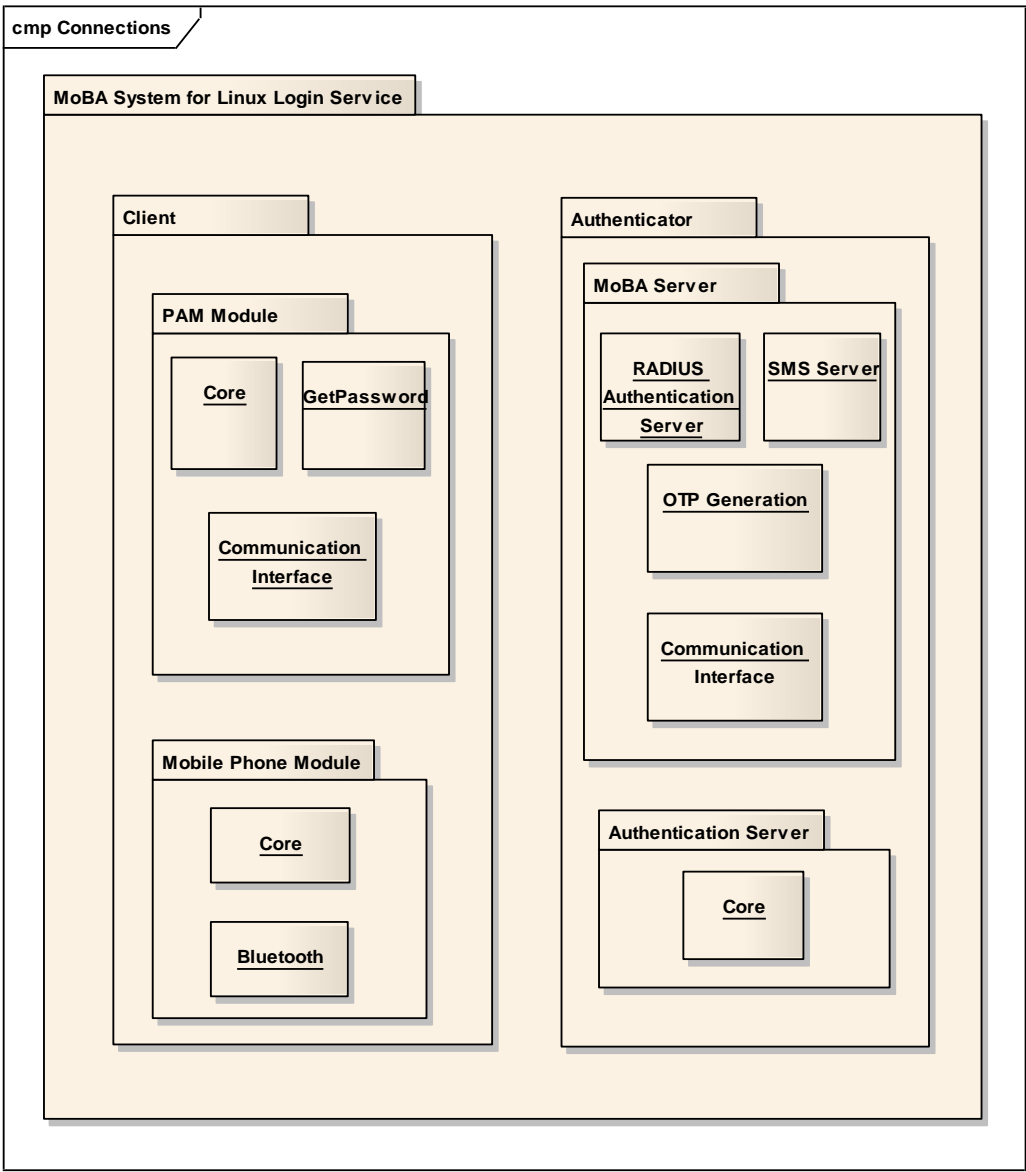


**Figure 37: MoBA Server Component Diagram.**

## 5.2 Package and Class Diagrams

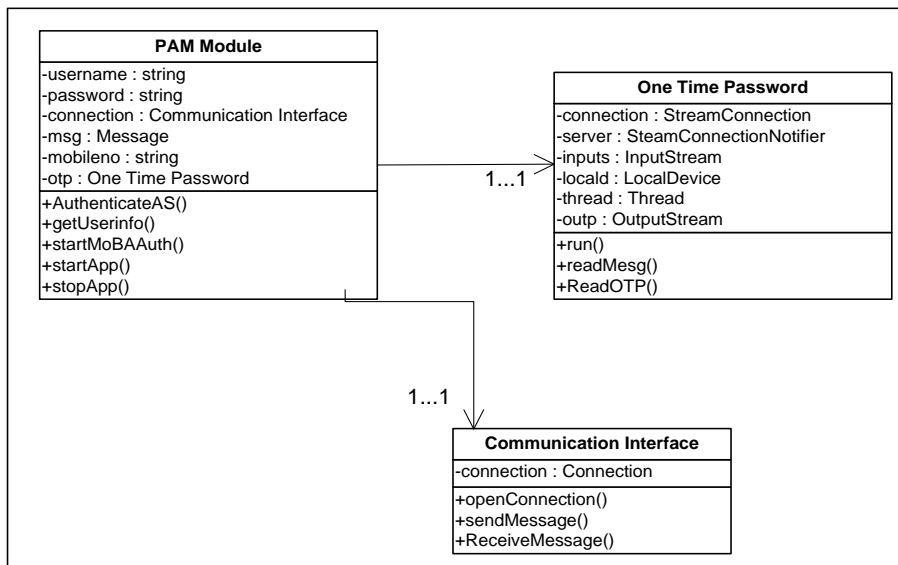
The class diagrams show the different objects in the system and their relationship. It also shows the attributes, operations and the constraints with the objects connections [86].

Package diagrams are used to see the overview of the system as the classes from the same part are grouped together to form a simple view of the system. The figure given below shows the package diagram of the system.



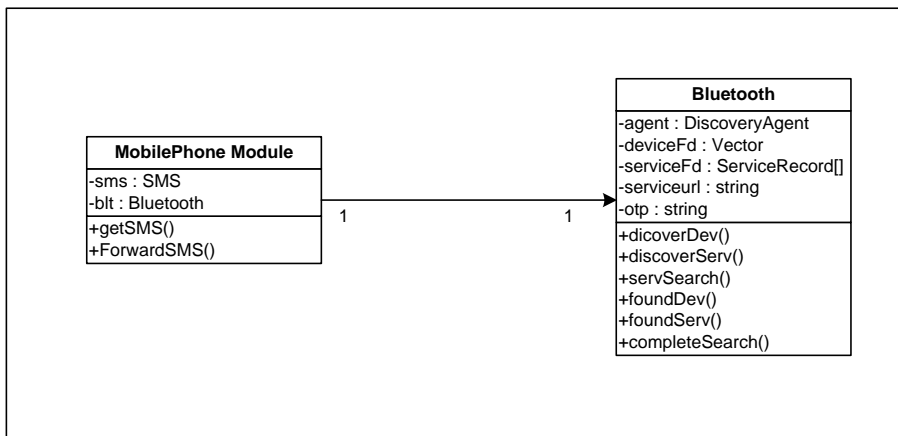
**Figure 38:** MoBA System for Linux Login Service Package Diagram.

The diagram given below show the PAM Module class diagram. The PAM module handles the user login and the functionality is implemented by PAM Module class. The One Time Password class implements the functionality that it will extract the OTP received by the user through SMS to forward it to PAM Module either manually or by bluetooth. The communication interface class implements the socket communication with the MoBA server for authentication. It controls all the communication with the MoBA Server.



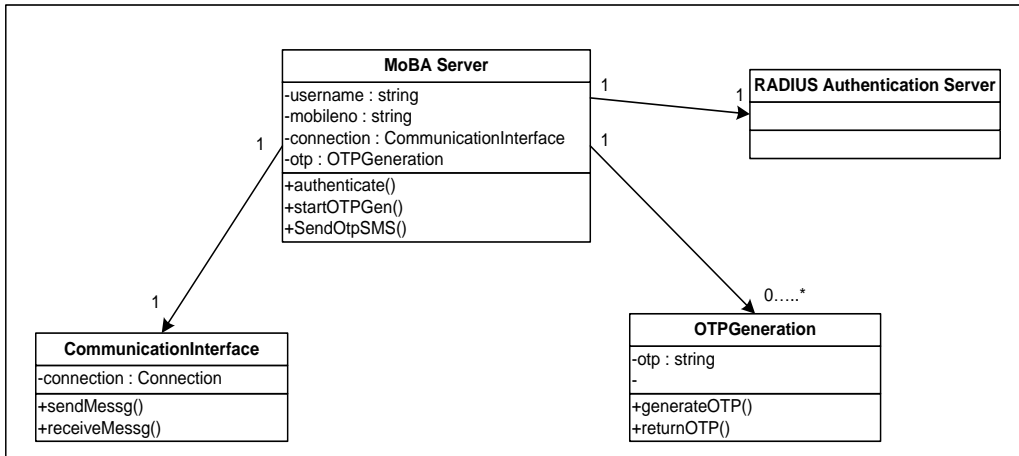
**Figure 39:** PAM Module Class Diagram.

The following diagram shows the class diagram of Mobile Phone Module. The Mobile phone module implements the functionality of receiving the OTP as SMS and then depends on the user choice to forward the OTP through Bluetooth or manually. The Bluetooth class implements the functionality of forwarding the OTP as SMS to the PAM module.



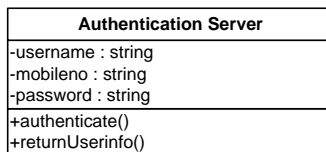
**Figure 40:** Mobile Phone Module Class Diagram.

The following diagram implements the functionality of the Mobile Based Authentication (MoBA) server. The MoBA Server class authenticates the users based on the OTP generated. It also handles the functionality of sending the OTP as SMS to the user so that he can authenticate himself by using it. The OTPGeneration class implements the functionality of generating a new OTP for every session. The CommunicationInterface class implements the functionality of handling the communication with the clients for authentication.



**Figure 41:** MoBA Server Class Diagram.

The following diagram shows the Authentication Server class diagram. Its functionality is to authenticate the users on the bases of stored static passwords. They also store the data of the user like name and their mobile numbers which can be used for the mobile based authentication for Linux workstation logon process in second stage.



**Figure 42:** Authentication Server Class Diagram.



## 6. Implementation

This chapter describes the implementation of the concepts of the proposed solution. It includes the deployment diagram of the system and the details of the components implementation are also discussed.

### 6.1 Deployment:

The deployment diagram shows that how the different software pieces of the system will run on different hardware and their relationship. Figure 43 shows the deployment diagram for the secure mobile authentication for Linux workstation logon process.

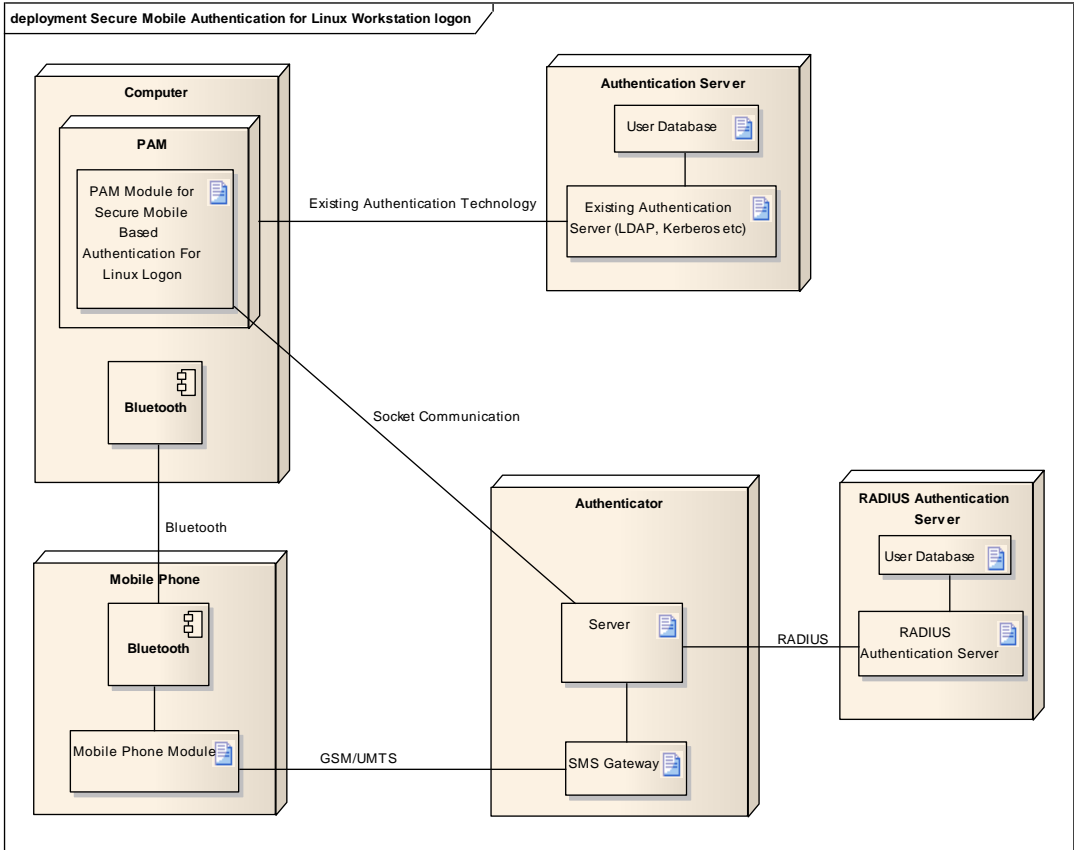


Figure 43: Secure MoBA for Linux Workstation Logon Deployment Diagram

### 6.2 Implementation of the components of the solution

The implementation of the components is as following:

#### 6.2.1 PAM Module

The PAM module has been installed on the client as PAM is used for authentication purpose in Linux environment. The PAM module has been implemented using C language. PAM makes the user free from the technology specific. It works as an intermediate API between the PAM functions and the underlying technology.

A PAM module has been implemented that calls the required functions for authentication and accessing data for the mobile based authentication from the existing enterprise authentication technology like LDAP, Kerberos etc. In section 2.5 it is explained how to write a PAM module. A sample code for writing a PAM module has been given in Appendix A as well.

The data of user is accessed with the *passwd* structure which has the following data members.

```
struct passwd {
    char *pw_name; /* user name */
    char *pw_passwd; /* user password */
    uid_t pw_uid; /* user id */
    gid_t pw_gid; /* group id */
    char *pw_gecos; /* real name */
    char *pw_dir; /* home directory */
    char *pw_shell; /* shell program */
};
```

The *pw\_gecos* data member is used for the mobile number of the user. The field is used as

*USER REAL NAME, MOBILE NUMBER*

E.g. george, 004746252374.

If the PAM module finds the mobile number it continues with the mobile based authentication as well otherwise it authenticates the user on the basis of the existing technology authentication method.

The other task that PAM Module performs is to interact with the Authenticator for the mobile based authentication. It used TCP socket communication for the communicating with the Authenticator. The PAM module also interacts with the user for getting the username and password. For mobile based Authentication it can accept the OTP either manually or by Bluetooth.

### 6.2.2 Authentication Server

To make the solution work with the existing technology like LDAP, Kerberos etc, the existing enterprise authentication server has been used that will be used for user data like user name, mobile number, the static password and can be used for the static password authentication as now a day's used for authentication in an enterprise. This will help in security as one authentication of user will be done on the static password and other on the basis of mobile based authentication. The PAM API helps us to be free of any specific technology and there are functions available for the required functionalities.

### 6.2.3 Authenticator

The Authenticator is implemented in C language. A multi process server has been implemented in C language to handle multiple users at a time. It works as a server and a connection point for PAM module and the RADIUS Server. It communicates with the PAM Module using TCP sockets. The other task that Authenticator performs is to generate a one time password (OTP). It generates OTP sends it to the SMS gateway so that OTP can be sent to the user as SMS. At the same time it saves the OTP in RADIUS server which can be used for authentication purpose. After the user receives the OTP and sends it back to the Authenticator the process waits. Upon receiving the OTP the Authenticator sends it to the RADIUS Server for authentication of user. If the user is authenticated by the RADIUS Server then user is allowed for login otherwise rejected and he has to do the authentication process from start.

#### **6.2.4 One Time Password (OTP)**

The OTP is used once. There are two methods implemented for the generation of the OTP.

In the first method SHA-384 is used. The following data is given as input to the algorithm

USERNAME | RANDOM NUMBER | TIME

The time part consists of day, month, year, hour, minute, second. It generates a hash of 48 bytes which is converted to hexadecimal values that can be human readable. When we change the hash value in hexadecimal values, we have then 96 hexadecimal characters.

The second method that we have implemented is using *dev/random* package of Linux for generating a random number which can be used as OTP. This package generates a random number based on the environment noise received through device drivers and other things [88].

We have used the second option as it gives us the freedom of selecting the length of OTP and it uses different characters like numbers, small and capital letters, special characters. We have used an OTP of 13 characters long, as it will help if the user has to enter the OTP manually.

#### **6.2.5 RADIUS Authentication Server**

The RADIUS Authentication Server is used for mobile based authentication of the user. It saves the OTP generated and do the authentication process.

#### **6.2.6 Mobile Phone Module**

A MIDlet is required to enable a mobile to send and receive the message through Bluetooth. We can configure the mobile so that when it receives the message from a certain number it will send it to required place using Bluetooth technology by using the PUSH to Register Service.

A MIDlet consists of a Java Archive (JAR) file containing the source code and a java application descriptor (JAD) file describing the MIDlet properties.

#### **6.2.7 SMS Gateway**

A web service is provided for SMS gateway. We have accessed the SMS gateway using the web service which has been provided.

### **6.3 Testing the Prototype**

The implementation has been tested to check the functionality of the system against the functional requirement that have been discussed in section 4.1.1.

#### **Environment of Testing**

There are 2 desktop, 1 laptop used for the testing purpose. Ubuntu 9.10 is installed on 2 computers and Ubuntu 8.10 is installed on the third computer. One computer is used as the server and the 2 are used as client who depends on the server for their authentication. Open LDAP has been installed so to have the Authentication Server. The PAM modules have been installed on client side and the server code of Authenticator is being run on the server side with RADIUS Server. Nokia 6230i has been used for the testing purpose.

#### **Testing Scenarios**

There are 2 users added to LDAP server for testing. The mobile number has been added to the *gecos* field of the database as discussed in section 6.2.1. In the given Figure 44 the users in the LDAP server have been displayed. The 2 users are George and Babar.

```

Applications Places System
usman@usman-desktop: ~
File Edit View Terminal Tabs Help
usman@usman-desktop:~$ ldapsearch -xLLL -b 'ou=people,dc=example,dc=com' cn
dn: ou=people,dc=example,dc=com

dn: uid=george,ou=people,dc=example,dc=com
cn: george

dn: uid=babar,ou=people,dc=example,dc=com
cn: babar

usman@usman-desktop:~$ █

```

**Figure 44:** Users in the LDAP Authentication Server

In the figure 45 the data of Babar user have been displayed. The gecos field is changed according to the requirement.

```

Applications Places System
usman@usman-desktop: ~
File Edit View Terminal Tabs Help
usman@usman-desktop:~$ sudo ldapfinger babar
dn: uid=babar,ou=people,dc=example,dc=com
objectClass: account
objectClass: posixAccount
cn: babar
uid: babar
uidNumber: 10002
gidNumber: 10000
homeDirectory: /home/babar
loginShell: /bin/sh
description: User account
userPassword:: e1NTSEF9SnlXS1hYRHRKYUI2SFpmWlhKSLRGSXFjTXhNQ25FVVg=
gecos: babar,4746252374

usman@usman-desktop:~$ █

```

**Figure 45:** User Information in LDAP Server

The server code at the Authenticator has been started so that it should be in waiting state for the client to be authenticated. Figure 46 shows the wait state.

```

Applications Places System
usman@usman-desktop: ~/Servercode
File Edit View Terminal Tabs Help
usman@usman-desktop:~$ cd Servercode
usman@usman-desktop:~/Servercode$ ./server
server: waiting for connections...
█

```

**Figure 46:** Server Waiting for Clients.

At the client side the client code has been executed. It asks for the username and the static password for the LDAP server authentication. After authentication of user with the static password, the data is send to the server for mobile based authentication. If user is not authenticated or mobile number is not found than the mobile based authentication is not done. Figure 47 shows the process at the client side

```
Applications Places System [Firefox] [Email] [Help]
usman@usman-desktop: ~/Project
File Edit View Terminal Help
usman@usman-desktop:~$ cd Project
usman@usman-desktop:~/Project$ ./pam_moba_login
login:george
Password:
Authentication Successful          retval=0
IP=129.241.208.66
PORT=3490
USERNAME=george
Mobile#=4746252374
client: connecting to 129.241.208.66
MOBA Password:█
```

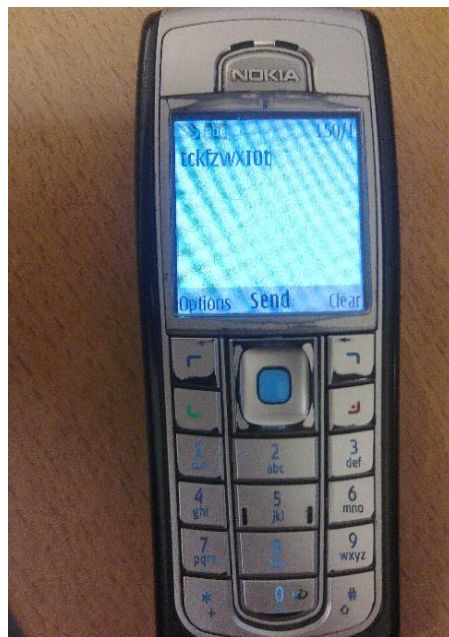
**Figure 47: LDAP Authentication**

With this the data is sent to the Authenticator for mobile based authentication. At the Authenticator the OTP is generated. At the same time the OTP is sent to the user through SMS. The following figure 48 shows the process.

```
Applications Places System [Firefox] [Email] [Help]
usman@usman-desktop: ~/Servercode
File Edit View Terminal Tabs Help
usman@usman-desktop:~$ cd Servercode
usman@usman-desktop:~/Servercode$ ./server
server: waiting for connections...
server got connection from ::ffff:129.241.208.219:3490
Mobile#=4746252374
User Name Connected to Server: george
OTP=tckfzwXI0t
█
```

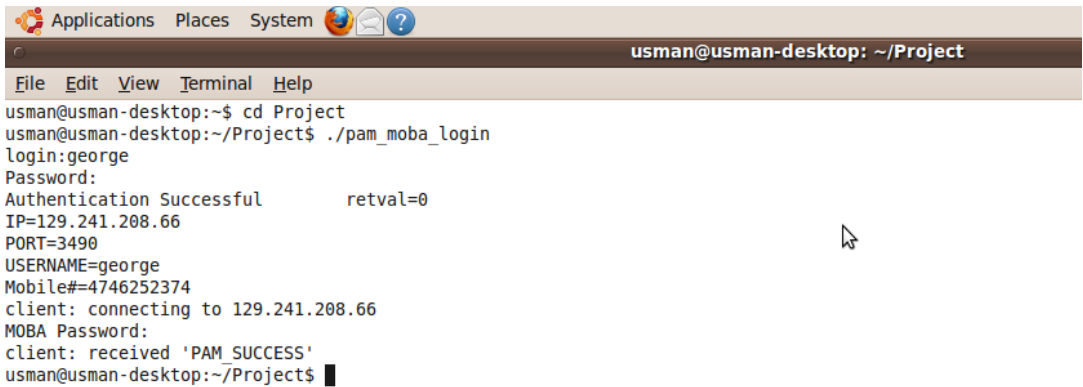
**Figure 48: OTP Generation Snapshot**

The OTP is received at the user mobile number as can be seen in the figure 49 below.



**Figure 49: OTP received at User Mobile**

The OTP received at the user mobile is typed by the user at the client console waiting for the MoBA password. If the password typed is correct the user will be authenticated otherwise the process will start again. The figure 50 given below shows the process

A terminal window titled 'usman@usman-desktop: ~/Project' with a menu bar (File, Edit, View, Terminal, Help). The terminal output shows the execution of './pam\_moba\_login' and the following details: login:george, Password:, Authentication Successful, retval=0, IP=129.241.208.66, PORT=3490, USERNAME=george, Mobile#=4746252374, client: connecting to 129.241.208.66, MOBA Password:, client: received 'PAM\_SUCCESS', and usman@usman-desktop:~/Project\$.

```
usman@usman-desktop:~$ cd Project
usman@usman-desktop:~/Project$ ./pam_moba_login
login:george
Password:
Authentication Successful          retval=0
IP=129.241.208.66
PORT=3490
USERNAME=george
Mobile#=4746252374
client: connecting to 129.241.208.66
MOBA Password:
client: received 'PAM_SUCCESS'
usman@usman-desktop:~/Project$
```

**Figure 50:** User Authentication Snapshot.

## 7. Validation and Evaluation

In this chapter the functionality of the proposed solution is validated against the functional requirements. The security evaluation of the solution is also done which defines the different types of attacks and the countermeasures for the attacks.

### 7.1 Validating the Functionality against Requirements

The functional requirements of the system are

#### Secure Mobile Based Authentication

The solution provides secure mobile based authentication of users for Linux workstation logon in an enterprise. As can be seen in the proposed solution in section 3.3, the PAM module at client communicates with the Authenticator for the mobile based authentication. And the Authenticator then communicates with the RADIUS Authentication server and user through his mobile phone using SMS.

#### Mutual Authentication

The client identity can be validated as the clients are in an enterprise environment where administrator has the knowledge of the network. The server validity can be confirmed by the validation of the OTP generated by the Authenticator received at the user mobile. Thus this will confirm the mutual authentication.

#### Two Factor Authentication

The user is authenticated on the basis of two factors. One is the static password that the user selects and the other factor is the mobile based authentication using a 13 character long OTP. Thus the two factor authentication will make the solution more secure

#### Working with the Existing technology

The solution will work with the existing static password systems. The solution uses PAM module at the client side which allows us free of the existing authentication server technology for enterprises like LDAP, Kerberos etc. Thus we can use the solution with the current technologies without changes.

#### Strong Password (OTP) For Mobile Based Authentication

The OTP is generated with the use of *dev/random* package which generates random numbers (OTP) using the noises from the environment obtained through the device drivers and other things. The password length is 13 characters consisting of letters, numbers and special characters, thus making it a strong password. This solution will be suitable for those cases where most of the users don't have the facility of using the Bluetooth feature of the solution and has to enter password manually.

The other solution for OTP that has been implemented is using SHA-384 which produces a 48 byte hash which is converted in 96 hexadecimal characters. This solution of OTP can be used when the all the users have the facility of using the Bluetooth feature.

#### Usability

Besides creating a secure mobile based authentication scheme the important issue is usability. As by using the mobile based authentication method no extra hardware is required on the client side and as most people always carry their mobile phone with them, thus making the solution easy for users to use.

## Cost and Deployment

As the solution will use the already functional mobile and SIM with the user which will dramatically reduce the deployment time and which will also has great impact on cost as well. The deployment will be easy as the solution will work with the existing architecture making the solution less costly. A new user in this solution will be authenticated almost immediately as the administrator adds the record of the user.

## Reliability, Availability and Scalability

If due to some problem, the mobile based authentication system is down then the user can be authenticated with the existing static password based authentication method with making the required changes by the administrator. Thus, makes the availability and reliability of the solution much higher. On the other hand the GSM system is considered to be very reliable and it is the most deployed mobile network in the world. The other important factor is that the system must scale well with the increase in the number of users. The proposed solution can be easily distributed both physically and geographically to avoid the possible bottlenecks during the authentication of users.

## 7.2 Different Types of Attack

There are different types of attacks which are important to be considered for the security of the solution. Following are some attacks

**Eavesdropping:** In this type of attack, the attacker will monitor the transmission between the client and authenticator for getting the sensitive information.

**Dictionary attack:** This is a kind of brute force attack where the attacker tries to find the password from the list of passwords called dictionary. As most of the time the password that people use are single words or common things that can be found in the dictionary.

**Masquerade:** In this type of attack the attacker pretends to be someone else. E.g. the attacker may masquerade himself as client and authenticator as well. The main goal is to get some sensitive information from the user.

**Replay attack:** In Replay attack the attacker reuses the old authentication sequence to have an unauthorized access to the system.

**Modification:** In this type of attack the attacker make some changes in the message or the contents of a message to generate an unauthorized effect

**Session hijacking:** In Session hijacking attack the attacker takes over a valid session that is already established to get an unauthorized access to the system.

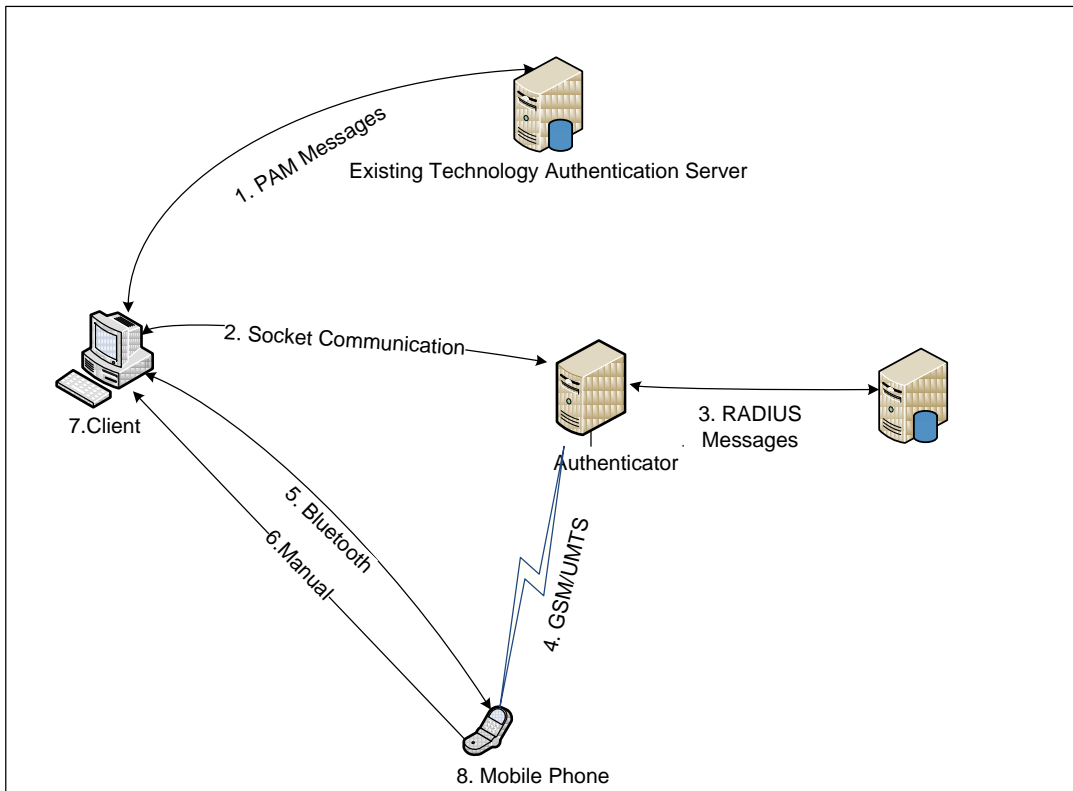
**Phishing:** In this type of attack the attacker uses social engineering techniques to get the user secret information.

**Man-in-the-middle (MITM) attack:** In this kind of attack the attacker can read and modify all messages that are going between the client and authenticator without the knowledge of the user.



### 7.3 Security Evaluation of the Solution

The following diagram shows the components of the security environment of the solution.



**Figure 51:** Component Threat Model

#### 1. PAM Messages

The PAM messages are secured by the existing server technology. So this component will be difficult for the attacker to attack as the system already provides the security.

#### 2. Socket Communication

This component is very important part of the solution to be protected as the user information is communicated through this channel. The security can be provided by using Transport layer security (TLS), so the communication channel will be secured from modification and the session hijacking attack. Secondly as the proposed solution requires 2 factors, one is the static password which is authenticated with a secure channel and the second is the OTP which is communicated through the mobile. If the OTP has been somehow released to the attacker he will still need the static password of the user and the OTP can only be used for once, making the task difficult for the attacker.

#### 3. RADIUS Messages

The RADIUS messages contain the security as the RADIUS technology already have.

#### **4. GSM/UMTS**

As UMTS has the security mechanisms available as it provides mutual authentication, so we can depend on it. For GSM, it is vulnerable to the man in the middle attack (MITM) as the network is not authenticated which is weak link. This problem is solved by having the static password authentication. As if the attacker gets the OTP he will still need the static password for the authentication. Secondly the OTP is used for once so he cannot misuse it.

#### **5. Bluetooth**

Bluetooth is another weak link which can be used by an attacker. To secure link we can use the encryption so that the data can be made secure. We can use the pass key made during the pairing for encryption as discussed in section 2.9. And the static password is the other data that the attacker has to find out to use the OTP.

#### **6. Manual typing of OTP**

When the user is typing the OTP there may be a chance that he writes it or release the OTP. If even the attacker finds the OTP, he will still require the static password for authentication. And the OTP is useful for authentication once, thus makes the OTP useless for the attacker.

#### **7. Client Computer**

The client computer can be attacked by the attacker for getting some useful information. As the client computer is in the enterprise network where the environment can be controlled by the network administrator to not allow such malware like Trojan horse or worms which can cause leakage of information. Secondly the OTP used once is useless for the second time.

#### **8. Mobile Phone**

Mobile phone is another important component for the security of the system. If the attacker steals the mobile with the SIM of the user, it will still be useless for the attacker as he will require the static password of the user as well.

#### **9. One Time Password**

It is necessary to keep the OTP strong enough so that the attacker cannot guess it or cannot find it with dictionary attack. As in our solution we have used 13 random characters consisting of numbers, letters and special characters. The NIST considers a password of 80 bits entropy to be strong enough against the brute force attack. As we have used 13 random characters so its entropy is equal to 85 bits thus making it strong enough against the brute force attack. The OTP is generated by using the *dev/random* package that generates the random numbers from the environment noise obtained from the device drivers and other things [88], thus creating random numbers which will make the guessing of OTP difficult.

Secondly the attacker needs the static password as well for authentication, so the attacker will have many things to coup with.

## 8. Conclusion

In order to come up with a proposed solution for secure mobile based authentication system for Linux workstation logon, a study of different technologies that are used in mobile phone based authentication schemes is done. The overview of identity management has been discussed. A study of smart card security, smart card operating systems, identity management UMTS networks and how security is provided in UMTS networks has been carried out.

An overview of the different enterprise identity management solutions used currently is made. The RADIUS technology has been explored to get the idea of an enterprise authentication server. A detailed study of Bluetooth security is done. An analysis of different available mobile phone based authentication schemes is done to get some sense how the mobile based systems work. Then the mechanism of authentication and authorization in Windows, Linux, and MAC OS X has been analyzed for adding the required functionality to the logon process. In Linux operating system the Pluggable authentication module technology provides extendibility by allowing different applications to have different modules for authentication. The logon process in Linux is treated as the same way as any other process, has its own PAM configuration file and can use any appropriate module.

While designing a solution it is required to make the Linux workstation logon more secure, user friendly and can work with the existing technologies. At the client (user) side a Pluggable Authentication module (PAM) module has been implemented which makes us free from being technology specific. By using the PAM API and the functions we can access the user information and authenticate user at the server without taking care of the technology at the server side like LDAP, Kerberos etc. The PAM API converts the required messages to the technology, making us free from technology. An Authenticator has been implemented which communicates with the PAM module using socket communication. The Authenticator serves as the connection point for the client and the RADIUS Authentication server. The Authenticator is responsible for generating OTP of 13 characters and communicating with the SMS gateway for sending the OTP as SMS to the user.

When the user will receive the OTP as SMS, he can insert it in the client computer as manually or by using Bluetooth technology. By using the option of manually entering the password, the user who does not have mobile phone with Bluetooth facility or the mobile of the user uses software which our solution does not support will still be able to use the solution. The mobile authentication is done by the RADIUS Authentication Server through the Authenticator.

The Solution uses two factors for authentication, one is the static password and the second factor is the mobile based authentication using a 13 character long OTP. With adding the existing technology and giving the administrator the option of selecting the authentication method for user makes the solution more suitable for an enterprise. The administrator can select either the authentication based on static password or both static password and mobile based authentication method. By adding this facility we add more reliability and availability to the system as if there is some problem with the mobile based authentication the users can still be authenticated with the existing static password technology.

The detailed analysis and design of the proposed solution has been carried out in this thesis so that to come up with an implementation of the different parts of the solution. The validation and the security evaluation is also discussed as to check the different security threats and how to tackle with the different security attacks

The solution which is proposed for the mobile based authentication for Linux workstation logon process will make the logon process more secure. One of the big advantages of the proposed

solution is that it uses the infrastructure that is already available which will make the cost less and the deployment very easy. Thus makes the solution more suitable.

## **Future Work**

In this thesis the prototype implemented shows the main concept of the solution and has been tested with very few users and exhausting testing has not been carried out. For commercial use it is required to have fully developed implementation that should be tested with all the security features and real time environment to get a secure and reliable system.

New techniques should be explored to find a common mobile based authentication architecture for multiple platforms like windows, Mac OS, Solaris and Linux etc in an enterprise environment.

As in case of using Bluetooth for automatic transfer of data from mobile to computer ,different mobile phone companies have different operating system technologies like symbian, Andriod, iPhone OS, palm OS etc. The method and support for accessing the SMS in all technologies is different. Therefore it is required to have a solution which can access the SMS without being technolgy specific. This will allow the enterprise that will use this solution to use the automatic version without taking care of the user mobile technolgy. And companies have to bear the cost of mobile phone for its users and that is one of the reason that we have added the manual version of entering the password.

It is required to explore a secure method of using the secrets or identities in the Mobile Network operator for the mobile based authentication of users.

## References

- [1] Do Van Thanh, Ivar Jorstad, “The Ambiguity of Identity”, teletronikk 3/4.07, 2007, available at: [http://www.telenor.com/teletronikk/volumes/pdf/3\\_4.2007/Page\\_003-010.pdf](http://www.telenor.com/teletronikk/volumes/pdf/3_4.2007/Page_003-010.pdf)
- [2] Do Van Thuan, “Identity Management Demystified”, teletronikk 3/4.07, 2007, available at: <http://www.telenor.com/teletronikk/volumes/index.php?page=ing&id1=73&id2=200&id3=976&select=05-09>
- [3] I. Jorstad, Do Van Thanh, “The Mobile Phone as Authentication Token”, teletronikk 3/4.07, 2007.
- [4] “Offering SIM Strong Authentication to Internet Services”, available at: [http://www.ongx.org/SIM\\_STRONG\\_WHITEPAPER2.3\\_SMS.Screen.pdf](http://www.ongx.org/SIM_STRONG_WHITEPAPER2.3_SMS.Screen.pdf)
- [5] S. Hallsteinsen, I. Jorstad, Do Van Thanh, “Using the mobile phone as a security token for unified authentication”, International Conference on Signal Processing, Communication and Networking (ICSCN), August 2007
- [6] “Accenture Leadership in Customer Service: Building the Trust”, 2006, available at: [http://www.accenture.com/Global/Services/By\\_Industry/Government\\_and\\_Public\\_Service/PS\\_Global/R\\_and\\_I/BuildingtheTrustES.htm](http://www.accenture.com/Global/Services/By_Industry/Government_and_Public_Service/PS_Global/R_and_I/BuildingtheTrustES.htm)
- [7] A. Miriam B. Lips, J. A. Taylor, J. Organ, “IDENTITY MANAGEMENT, ADMINISTRATIVE SORTING AND CITIZENSHIP IN NEW MODES OF GOVERNMENT”, Information, Communication & Society, 12:5,715 — 734, 2009, available at: <http://dx.doi.org/10.1080/13691180802549508>
- [8] M. Crompton, “Proof of ID required? Getting Identity Management Right”, Australian IT Security Forum, 2004, available at: <http://www.privacy.gov.au/materials/types/speeches/view/6339>
- [9] “Identity”, Wikipedia, [cited] September 2009, available at: [http://en.wikipedia.org/wiki/Identity\\_\(philosophy\)](http://en.wikipedia.org/wiki/Identity_(philosophy))
- [10] “Identity Management Design Guide with IBM Tivoli Identity Manager”, 3<sup>rd</sup> edition, IBM Corp., April 2009, available at: <http://www.redbooks.ibm.com/redbooks/pdfs/sg246996.pdf>
- [11] “What is the data store”, available at: <http://technet.microsoft.com/en-us/library/cc787905%28WS.10%29.aspx>
- [12] D. Birch, “Digital Identity Management: Perspectives On The Technological, Business and Social Implications”, Gower Pub Co, 2007
- [13] “Sun Java System Directory Server Enterprise Edition 6.3 Evaluation Guide”, SunMicrosystems, 2008, available at: <http://dlc.sun.com/pdf/820-2766/820-2766.pdf>
- [14] M. Benantar, “Access Control Systems: Security, Identity Management and Trust Models”, Springer Science+Business Media, 2006.
- [15] “The Role of Kerberos in Modern Information Systems”, MIT Kerberos Consortium, 2008, available at: <http://www.kerberos.org/software/rolekerberos.pdf>
- [16] “The MIT Kerberos Administrator’s How-to Guide”, MIT Kerberos Consortium, 2008, available at: <http://www.kerberos.org/software/adminkerberos.pdf>
- [17] C. Neuman, T. Yu, S. Hartman, K. Raeburn, “The Kerberos Network Authentication Service (V5)”, RFC 4120, July 2005, available at: <http://tools.ietf.org/html/rfc4120>

- [18] “Recommended Practices for Deploying & Using Kerberos in Mixed Environments”, MIT Kerberos Consortium, 2008, available at: <http://www.kerberos.org/software/mixenvkerberos.pdf>
- [19] D. Todorov, ”Mechanics of user identification and authentication: Fundamentals of Identity Management”, Auerbach Publications, 2007
- [20] W. R. Stanek, “Windows Server 2008 insideout”, Microsoft Press, 2008
- [21] W. Soyinka, “Linux administration Beginner’s guide”, 5<sup>th</sup> edition, McGraw-Hill, 2008
- [22] T. Bialaski, Michael Haines, “LDAP in the Solaris Operating Environment: Deploying Secure Directory”, Prentice Hall, 2003
- [23] R. Harrison, “Lightweight Directory Access Protocol (LDAP): Authentication Methods and Security Mechanisms”, RFC 4513, June 2006, available at: <http://tools.ietf.org/html/rfc4513>
- [24] S. Huque “LDAP Weaknesses as a Central Authentication System”, University of Pennsylvania, 2007, available at: <http://www.huque.com/~shuque/doc/2007-10-LDAP-Authn.html>
- [25] “Investigating Single Sign-on”, Novell white paper, available at: [http://www.novell.com/rc/docrepository/public/37/basedocument.2007-08-07.2321076507/4622014\\_en.pdf](http://www.novell.com/rc/docrepository/public/37/basedocument.2007-08-07.2321076507/4622014_en.pdf)
- [26] “SunOpenSSO Enterprise 8.0 TechnicalOverview”, SunMicrosystems, March 2009, available at: <http://dlc.sun.com/pdf/820-3740/820-3740.pdf>
- [27] Jan De Clercq, “Single Sign-On Architectures”, Springer Berlin, 2002, available at: <http://www.springerlink.com/content/806c0atpq9ab0nx4/>
- [28] Dao Van Tran, Pål Løkstad, Do Van Thanh, ”Identity Federation in a Multi Circle-of-Trust Constellation”, Telektronikk, Telenor, 2007, Volume 103, pp. 103-117
- [29] “Security Assertion Markup Language (SAML) V2.0 Technical Overview”, OASIS Committee Draft, 2008, available at: <http://www.oasis-open.org/committees/download.php/27819/sstc-saml-tech-overview-2.0-cd-02.pdf>
- [30] “Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) V2.0”, OASIS standard, 2005, available at: <http://docs.oasis-open.org/security/saml/v2.0/saml-sec-consider-2.0-os.pdf>
- [31] K. Mayes, K. Markantonakis, “Smart cards, Tokens, Security and Applications”, Springer Science+Business Media, 2008.
- [32] B. Holcombe, “Government smart card handbook”, available at: <http://www.smartcard.gov/information/smartcardhandbook.pdf>.
- [33] W. Rankl, W. Effing, “Smart Card Handbook”, 3rd Edition, Wiley, November 2003.
- [34] “What Makes a Smart Card Secure?”, Smart Card Alliance Contactless and Mobile Payments Council White Paper, October 2008, available at: <http://www.smartcardalliance.org/pages/download>.
- [35] “MF1ICS70 Functional specification”, Rev. 4.1, January 2008, available at: [http://www.nxp.com/acrobat\\_download/other/identification/M043541\\_MF1ICS70\\_Fspec\\_rev4\\_1.pdf](http://www.nxp.com/acrobat_download/other/identification/M043541_MF1ICS70_Fspec_rev4_1.pdf)

- [36] J. Ausssel, "Smart Cards and Digital Identity", Telektronikk, Telenor, 2007, Volume 103, pp. 66-79.
- [37] F. Vater, S. Peter, P. Langendorfer, "Combinatorial Logic Circuitry as Means to Protect Low Cost Devices Against Side Channel Attacks", Springer Berlin, 2007, available at: <http://www.springerlink.com/content/78k77t2m75731737/fulltext.pdf>.
- [38] "Common Criteria for Information Technology Security Evaluation", v.3.1, July 2009, available at: <http://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R3.pdf>
- [39] K.Boman, G.Horn, P.Howard, V.Niemi, "UMTS security", Electronics & Communication Engineering Journal, Volume 14, Issue 5, Oct. 2002 Page(s):191 - 204 October 2002, available at: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1088436&isnumber=23651>
- [40] M. Barbeau, Jean-Marc Robert, "Perfect identity concealment in UMTS over radio access links", Wireless And Mobile Computing, Networking And Communications, 2005. (WiMob'2005), IEEE International Conference on, Volume 2, 22-24 Aug. 2005 Page(s): 72 - 77 Vol. 2, available at: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1512853&isnumber=32399>
- [41] Abdul Bais, W.T. Penzhorn, P. Palensky, "Evaluation of UMTS security architecture and services" Industrial Informatics, 2006 IEEE International Conference on, 16-18 Aug. 2006 Page(s):570 – 575, available at: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4053451&isnumber=4053336>
- [42] J. A. Audestad, "Technologies and systems for access and transport networks", Artech House, 2008.
- [43] "UMTS", Wikipedia, available at: <http://en.wikipedia.org/wiki/Umts>
- [44] P. Hernberg, "User Authentication HOWTO", 02-05- 2000, available at: <http://www.faqs.org/docs/Linux-HOWTO/User-Authentication-HOWTO.html#AEN59>
- [45] "AIX: root group and system group ", [Cited] November 2009, available at: <http://www.linuxquestions.org/questions/aix-43/aix-root-group-and-system-group-751399/>
- [46] "The wheel Group", [cited] November 2009, available at: <http://administratosphere.wordpress.com/2007/07/19/the-wheel-group/>
- [47] "Hardening Linux authentication and user identity", September 23, 2004, available at: <http://www.linux.com/archive/articles/39114>
- [48] "The Linux Login Process", [cited] November 2009, available at: [http://www.comptechdoc.org/os/linux/howlinuxworks/linux\\_hllogin.html](http://www.comptechdoc.org/os/linux/howlinuxworks/linux_hllogin.html)
- [49] "Access Control lists in Linux", Administration guide, 18<sup>th</sup> July 2003, available at: [http://www.suse.de/~agruen/acl/chapter/fs\\_acl-en.pdf](http://www.suse.de/~agruen/acl/chapter/fs_acl-en.pdf)
- [50] A. Grunbacher, "POSIX Access control lists on Linux", USENIX Annual Technical Conference, San Antonio, Texas, June 2003, available at: <http://www.suse.de/~agruen/acl/linux-acls/linux-acls-final.pdf>
- [51] J. A. Martinez, M. Strasser,A. Tapaninen,T. Sirainen, L. Rousseau, "PAM-PKCS11 User Manual", [cited] December 2009, available at: [http://www.opensc-project.org/doc/pam\\_pkcs11/pam\\_pkcs11.html#introduction](http://www.opensc-project.org/doc/pam_pkcs11/pam_pkcs11.html#introduction)
- [52] Kenneth Geisshirt, "Pluggable Authentication Modules: The Definitive Guide to PAM for Linux SysAdmins and C Developers", Packt Publishing, 2007.

- [53] J. M. Johansson, "Windows Server 2008 Security Resource Kit", Microsoft Press, 2008
- [54] M. E. Russinovich, D. A. Solomon, "Microsoft Windows Internals: Microsoft Windows Server 2003, Windows XP, and Windows 2000", 4<sup>th</sup> ed., Microsoft Press, 2005
- [55] D. Todorov, "Mechanics of user identification and authentication: Fundamentals of Identity Management", Auerbach Publications, 2007
- [56] "Access Control Model", MSDN Library, 2009, available at: <http://msdn.microsoft.com/en-us/library/aa374876%28VS.85%29.aspx>
- [57] "Access Control Lists", MSDN Library, 2009, available at: <http://msdn.microsoft.com/en-us/library/aa374872%28VS.85%29.aspx>
- [58] "Access Mask Format", MSDN Library, 2009, available at: <http://msdn.microsoft.com/en-us/library/aa374896%28VS.85%29.aspx>
- [59] "How DACLs Control Access to an Object", MSDN Library, 2009, available at: <http://msdn.microsoft.com/en-us/library/aa446683%28VS.85%29.aspx>
- [60] "Privileges", MSDN Library, 2009, available at: <http://msdn.microsoft.com/en-us/library/aa379306%28VS.85%29.aspx>
- [61] "How Interactive Logon Works", Windows TechNet Library, 2009, available at: [http://technet.microsoft.com/en-us/library/cc780332%28WS.10%29.aspx#w2k3tr\\_intlg\\_how\\_tpxs](http://technet.microsoft.com/en-us/library/cc780332%28WS.10%29.aspx#w2k3tr_intlg_how_tpxs)
- [62] M. E. Russinovich, D. A. Solomon, A. Ionescu, "Windows Internals: Including Windows Server 2008 and Windows Vista", 5<sup>th</sup> ed., Microsoft Press, 2009
- [63] R. Morimoto, M. Noel, O. Droubi, R. Mistry, C. Amaris, "Windows Server 2008 Unleashed", Sams, 2008
- [64] "Application Compatibility: Session 0 Isolation", MSDN Library, 2009 available at : <http://msdn.microsoft.com/en-us/library/bb756986.aspx>
- [65] D. Griffin, "Create Custom Login Experiences With Credential Providers For Windows Vista", MSDN magazine, 2007, available at: <http://msdn.microsoft.com/en-us/magazine/cc163489.aspx>
- [66] "Understanding Logon and Authentication", 2005, available at: <http://technet.microsoft.com/en-us/library/bb457114.aspx>
- [67] L. Zhu, B. Tung, "Public Key Cryptography for Initial Authentication in Kerberos (PKINIT)", RFC 4556, 2006, available at: <http://tools.ietf.org/html/rfc4556>
- [68] A. Nirmalanathan, "The Smart Card Cryptographic Service Provider Cookbook ", Microsoft Corporation, 2002, available at: <http://msdn.microsoft.com/en-us/library/ms953432.aspx>
- [69] "Windows Vista Smart Card Infrastructure ", Microsoft Corporation, available at: <http://msdn.microsoft.com/en-us/library/bb905527.aspx>
- [70] "What's New in Smart Cards", 2009, available at: <http://technet.microsoft.com/en-us/library/dd367851%28WS.10%29.aspx>
- [71] Dave," How Mac OS X Implements Password Authentication, Part 2", and April 2006, available at: [http://www.dribin.org/dave/blog/archives/2006/04/28/os\\_x\\_passwords\\_2/](http://www.dribin.org/dave/blog/archives/2006/04/28/os_x_passwords_2/)



- [72] “Understanding and Using NetInfo”, [cited] November 2009, available at: [http://download.info.apple.com/Apple\\_Support\\_Area/Manuals/software/UnderstandingUsingNetInfo.PDF](http://download.info.apple.com/Apple_Support_Area/Manuals/software/UnderstandingUsingNetInfo.PDF)
- [73] “Authorization”, MAC OS X reference Library, [cited] November 2009, available at: [http://developer.apple.com/mac/library/documentation/Security/Conceptual/authorization\\_concepts/02authconcepts/authconcepts.html#//apple\\_ref/doc/uid/TP30000995-CH205-TPXREF9](http://developer.apple.com/mac/library/documentation/Security/Conceptual/authorization_concepts/02authconcepts/authconcepts.html#//apple_ref/doc/uid/TP30000995-CH205-TPXREF9)
- [74] “Mac OS X 10.4: Enabling smart card login”, MAC OS X reference Library, [cited] December 2009, available at: [http://support.apple.com/kb/TA24244?viewlocale=en\\_US](http://support.apple.com/kb/TA24244?viewlocale=en_US)
- [75] “RADIUS”, Wikipedia, [cited] 18<sup>th</sup> May, 2010, available at: <http://en.wikipedia.org/wiki/RADIUS>
- [76] Jon Edney, Willam A. Arbaugh, "Real 802.11 Security: WiFi Protected Access and 802.11i", Addison Wesley, 2003.
- [77] “Simple Pairing Whitepaper”, v.10r00, Core Specification Working Group, 2006, available at: [http://www.bluetooth.com/NR/rdonlyres/0A0B3F36-D15F-4470-85A6-F2CCFA26F70F/0/SimplePairing\\_WP\\_V10r00.pdf](http://www.bluetooth.com/NR/rdonlyres/0A0B3F36-D15F-4470-85A6-F2CCFA26F70F/0/SimplePairing_WP_V10r00.pdf)
- [78] K. Scarfone, J. Padgette, “Guide to Bluetooth Security”, National Institute of Standards and Technology (NIST), NIST Special Publication 800-121, September 2008, available at: <http://csrc.nist.gov/publications/nistpubs/800-121/SP800-121.pdf>
- [79] “National Information Assurance Glossary”, available at: [http://www.cnss.gov/Assets/pdf/cnssi\\_4009.pdf](http://www.cnss.gov/Assets/pdf/cnssi_4009.pdf)
- [80] “Extensible Authentication Protocol Method for Global System for Mobile Communications (GSM) Subscriber Identity Modules (EAP-SIM)”, RFC 4186, available at: <http://tools.ietf.org/html/rfc4186>
- [81] “S/Key”, Wikipedia, [cited] November 2009, available at: <http://en.wikipedia.org/wiki/S/KEY>
- [82] “HOTP: An HMAC-Based One-Time Password Algorithm”, RFC 4226, available at: <http://tools.ietf.org/html/rfc4226>
- [83] “TOTP: Time-based One-time Password Algorithm”, Internet-Draft, available at: <http://tools.ietf.org/html/draft-mraihi-totp-timebased-03>
- [84] “OCRA: OATH Challenge-Response Algorithms”, Internet-Draft, available at: <http://tools.ietf.org/html/draft-mraihi-mutual-oath-hotp-variants-09>
- [85] “Component (UML)”, [Cited] 21<sup>st</sup> April, 2010, available at: [http://en.wikipedia.org/wiki/Component\\_UML](http://en.wikipedia.org/wiki/Component_UML)
- [86] Donald Bell, “UML basics: The class diagram”, [cited] 3<sup>rd</sup> May, 2010, available at: <http://www.ibm.com/developerworks/rational/library/content/RationalEdge/sep04/bell/>
- [87] “UML 2 Package Diagrams”, [Cited] 3<sup>rd</sup> May, 2010, available at: <http://www.agilemodeling.com/artifacts/packageDiagram.htm>
- [88] “Random”, [Cited] 15<sup>th</sup> May, 2010, available at: <http://linux.die.net/man/4/random>

## Appendix A

### Sample Code for PAM Module

```
#include <security/pam_appl.h>
#include <security/pam_misc.h>
#include <stdio.h>

//Conversation function
static struct pam_conv conver = {
    misc_conv,
    NULL
};

// main function
int main(int argc, char *argv[])
{
    pam_handle_t *pamhandler=NULL;
    int retvalue;
    const char *user="username";

    // starting PAM session and initializing the PAM data structure
    retvalue = pam_start("check", user, &conver, &pamhandler);

    //check if the PAM session is created
    if (retvalue == PAM_SUCCESS)
        retvalue = pam_authenticate(pamhandler, 0); /*Authenticating user */

    //check if user is authenticated
    if (retvalue == PAM_SUCCESS)
        retvalue = pam_acct_mgmt(pamhandler, 0); /* permitted access? */

    /* This is where we have been authorized or not. */

    if (retvalue == PAM_SUCCESS) {
        fprintf(stdout, "User is Authenticated\n");
    } else {
        fprintf(stdout, "User is not Authenticated\n");
    }

    if (pam_end(pamhandler,retvalue) != PAM_SUCCESS) { /* close Linux-PAM */
        pamhandler = NULL;
        fprintf(stderr, "check_user: failed to release authenticator\n");
        exit(1);
    }
    // indicate success or failure

    return ( retvalue == PAM_SUCCESS ? 0:1 );
}
////////////////////Code END////////////////////
```

## Compilation of PAM Module

For compiling the PAM Module the following libraries files should be added. The command will be like

```
gcc filename.c -lpam -lpam_misc -o outputfile
```

For creating .so files which can be used than in PAM configuration files for accessing the service we should do the following steps

```
$ gcc -fPIC -c filename.c  
$ ld -x --shared -o filename.so filename.o  
$ sudo cp filename.o /lib/security
```

## **Appendix B**

The source code for the secure mobile based authentication prototype is attached as a ZIP file. The attachment has the following data

1. Source code for the PAM module for the client side
2. Source code for the Authenticator
3. A “read me” file explaining how to use the prototype



# Problem Description

The Master thesis work will study and propose how workstation identity management can be made more user-friendly and secure by using the mobile phone in the Linux workstation log on process. The current log on process is neither user friendly nor sufficiently secure. This work consists of the following tasks:

1. Study of existing identity management and authentication for enterprise network and workstations.
2. Study of Platform security architecture of Linux.
3. Study of Authentication and identity management standards and protocols (e.g. EAP, RADIUS/ DIAMETER, SAML etc.)
4. Design and implementations of solutions for integration of the mobile phone with the workstation logon processes in different operating systems
5. Threat and vulnerability analysis of proposed solution(s).

Assignment given: 21. January 2010

Supervisor: Van Thanh Do, ITEM



# Preface

This thesis is done as Master thesis in the Master of Telematics: Communication Networks and Networked Services program at the Norwegian University of Science and Technology (NTNU). The thesis has been performed in the spring semester 2010 at the department of Telematics in collaboration with Telenor R&I.

The supervisor for this thesis has been Ivar Jørstad at UbiSafe AS and the academic responsible has been professor Do van Thanh. I would like to thank them both for their help and support during the work with the thesis and especially Do Van Thanh for the valuable suggestions and comments.

Trondheim, June 17, 2010

Usman Habib.





# Abstract

Password based logon schemes have many security weaknesses. For secure environments smart card and biometric based authentication solutions are available as replacement for standard password based systems. Nevertheless, the cost of deployment and maintenance of these systems is quite high. On the other hand, mobile network operators have a huge base of deployed smart cards that can be reused to provide authentication in other spheres significantly reducing costs. In this project we present a study of how mobile phones can be used to provide a secure low-cost two-factor workstation logon solution.

To find and study the available mobile phone based authentication architectures and come up with workstation logon architecture the study of relevant technologies utilized in these solutions: UMTS networks, Bluetooth communication, Remote Authentication Dial in User Service (RADIUS), authentication and authorization in Windows, Linux, and MAC OS X. The analysis of available mobile phone based authentication schemes like SIM Strong schemes based on EAP-SIM, Session-ID based schemes, and OTP based schemes are also added.

A solution for Linux workstation logon process has been proposed in the thesis using the Pluggable Authentication Module (PAM). The Solution uses 2 factors for authentication, one is the static password and the second factor is the mobile based authentication using a 13 character long OTP. With adding the existing technology and giving the administrator the option of selecting the authentication method for user makes the solution more suitable for an enterprise.



# Table of Contents

<b>1. Introduction.....</b>	<b>1</b>
1.1 Motivation.....	1
1.2 Problem definition.....	2
1.3 Objectives .....	2
1.4 Organization of the thesis .....	3
<b>2. Background.....</b>	<b>4</b>
2.1 Identity and identity management.....	4
2.2 Identity management in enterprise systems.....	8
2.3 Smart card technology .....	19
2.4 Identity management in UMTS system.....	23
2.5 Authentication and authorization in Linux.....	27
2.6 Authentication and authorization in Windows .....	36
2.7 Authentication and authorization in Mac OS X .....	42
2.8 Remote Authentication Dial In User Service (RADIUS) .....	46
2.9 Bluetooth security .....	48
<b>3. Mobile Phone Based Authentication Systems .....</b>	<b>53</b>
3.1 General Mobile Phone Authentication Schemes.....	53
3.1.1 SMS Authentication with Session-ID check .....	53
3.1.2 SIM Strong Authentication .....	54
3.1.3 EAP-AKA .....	56
3.2 One Time Password Schemes .....	56
3.2.1 One Time Password From PC to SMS .....	57
3.2.2 One Time Password From SMS to PC .....	59
3.2.3 Enhanced OTP From PC to SMS authentication.....	61
3.3 Proposed Solution for Mobile Based Linux Workstation Logon Service.....	64
<b>4. Analysis .....</b>	<b>67</b>
4.1 Requirements .....	67
4.1.1 Functional Requirements .....	67
4.1.2 Non Functional Requirements .....	67

4.2 Use Case Diagrams .....	67
4.3 Interaction Diagrams .....	72
4.3.1 Successful Authentication.....	72
<b>5. Design .....</b>	<b>74</b>
5.1 Components .....	74
5.1.1 PAM Module .....	74
5.1.2 Mobile Phone Module .....	75
5.1.3 Authentication Server .....	76
5.1.4 MoBA Server .....	76
5.2 Package and Class Diagrams .....	77
<b>6. Implementation .....</b>	<b>81</b>
6.1 Deployment.....	81
6.2 Implementation of the Components of the solution .....	81
6.2.1 PAM Module .....	81
6.2.2 Authentication Server .....	82
6.2.3 Authenticator .....	82
6.2.4 One Time Password (OTP).....	83
6.2.5 RADIUS Authentication Server.....	83
6.2.6 Mobile Phone Module .....	83
6.2.7 SMS Gateway.....	83
6.3 Testing the Prototype .....	83
<b>7. Validation and Evaluation .....</b>	<b>87</b>
7.1 Validating the Functionality against Requirements .....	87
7.2 Different Types of Attack .....	88
7.3 Security Evaluation of the solution.....	89
<b>8. Conclusion.....</b>	<b>91</b>
<b>References.....</b>	<b>93</b>
<b>Appendix A.....</b>	<b>98</b>
<b>Appendix B .....</b>	<b>100</b>

# List of Figures

<b>Figure 1:</b> Partnership Identity Management [2].....	8
<b>Figure 2:</b> Identity and access manager architecture .....	11
<b>Figure 3:</b> Kerberos authentication model .....	13
<b>Figure 4:</b> ALL IP UMTS.....	24
<b>Figure 5:</b> UMTS architecture [40] .....	25
<b>Figure 6:</b> Mutual Authentication and Key Agreement [40].....	26
<b>Figure 7:</b> Login Process in Linux.....	31
<b>Figure 8:</b> Mapping between ACL entries & File mode Permission Bits [50] .....	33
<b>Figure 9:</b> PAM Application Structure .....	34
<b>Figure 10:</b> Interaction of NetInfo and Users [72] .....	42
<b>Figure 11:</b> Administrator Management with NetInfo [72] .....	43
<b>Figure 12:</b> Shared Domain in NetInfo [72] .....	43
<b>Figure 13:</b> NetInfo Hierarchies [72].....	43
<b>Figure 14:</b> RADIUS Architecture [75].....	46
<b>Figure 15:</b> RADIUS Different Authentication Operations.....	47
<b>Figure 16:</b> RADIUS Message Format.....	47
<b>Figure 17:</b> SSP Link Key Establishment for pairing [78] .....	49
<b>Figure 18:</b> Bluetooth Authentication [42].....	51
<b>Figure 19:</b> Bluetooth Encryption [78] .....	52
<b>Figure 20:</b> SMS authentication with sessionID check .....	54
<b>Figure 21:</b> EAP-SIM authentication.....	55
<b>Figure 22:</b> EAP-AKA authentication .....	56
<b>Figure 23:</b> OTP from PC to phone authentication [3] .....	58
<b>Figure 24:</b> OTP Applet [3].....	59
<b>Figure 25:</b> OTP SMS to PC authentication [3].....	60
<b>Figure 26:</b> Component Architecture [5] .....	61
<b>Figure 27:</b> Authentication process.....	62
<b>Figure 28:</b> Key Exchange procedure [5] .....	63
<b>Figure 29:</b> Mobile Based Authentication System for Linux Workstation Logon .....	65
<b>Figure 30:</b> General Use Case Diagram.....	68
<b>Figure 31:</b> OTP Generation Use Case .....	69
<b>Figure 32:</b> Sequence Diagram- Authentication.....	73
<b>Figure 33:</b> Main Component Diagram. ....	74
<b>Figure 34:</b> PAM Module Component Diagram. ....	75
<b>Figure 35:</b> Mobile Phone Component Diagram.....	76

<b>Figure 36: Authentication Server Component Diagram.</b> .....	76
<b>Figure 37: MoBA Server Component Diagram.</b> .....	77
<b>Figure 38: MoBA System for Linux Login Service Package Diagram.</b> .....	78
<b>Figure 39: PAM Module Class Diagram.</b> .....	79
<b>Figure 40: Mobile Phone Module Class Diagram.</b> .....	79
<b>Figure 41: MoBA Server Class Diagram.</b> .....	80
<b>Figure 42: Authentication Server Class Diagram.</b> .....	80
<b>Figure 43: Secure MoBA for Linux Workstation Logon Deployment Diagram</b> .....	81
<b>Figure 44: Users in the LDAP Authentication Server</b> .....	84
<b>Figure 45: User Information in LDAP Server</b> .....	84
<b>Figure 46: Server Waiting for Clients.</b> .....	84
<b>Figure 47: LDAP Authentication</b> .....	85
<b>Figure 48: OTP Generation Snapshot</b> .....	85
<b>Figure 49: OTP received at User Mobile</b> .....	85
<b>Figure 50: User Authentication Snapshot</b> .....	86
<b>Figure 51: Component Threat Model</b> .....	89

# List of Tables

<b>Table 1:</b> Identities in UMTS components [40].....	25
<b>Table 2:</b> Linux built-in groups.....	28
<b>Table 3:</b> Types of ACL entries [50] .....	32
<b>Table 4:</b> Masking of permissions [50] .....	33
<b>Table 5:</b> Return Codes .....	35
<b>Table 6:</b> Built-in accounts in Windows.....	36
<b>Table 7:</b> Well-known system groups in Windows .....	37
<b>Table 8:</b> Logon rights in Windows .....	38
<b>Table 9:</b> Authorization in MAC OS.....	45
<b>Table 10:</b> Rule Attributes and Description [73] .....	45
<b>Table 11:</b> Tokens installed with MAC OS .....	46
<b>Table 12:</b> Use Case-Login Service .....	68
<b>Table 13:</b> Use Case-Maintain User Information .....	69
<b>Table 14:</b> Use Case-OTP Generation.....	70
<b>Table 15:</b> Use Case-Get User Information.....	70
<b>Table 16:</b> Use Case-Generate OTP.....	71
<b>Table 17:</b> Use Case-Send OTP.....	71
<b>Table 18:</b> Use Case-Perform Authentication.....	71
<b>Table 19:</b> Use Case-OTP SMS .....	72





# Abbreviations

AAA	Authentication, Authorization and Accounting
ACL	Access Control List
AES	Advanced Encryption Standard
AK	Authentication Key
AKA	Authentication and Key Agreement
API	Application Programming Interface
AS	Authentication Server
AuC	Authentication Center
BSC	Base Station Controller
BSS	Base Station Subsystem
BTS	Base Transceiver Station
CK	Cipher Key
DAC	Discretionary Access Control
EAL	Evaluation assurance level
EAP	Extensible Authentication Protocol
ECDH	Elliptic Curve Diffie Hellman
EIR	Equipment Identity Register
FAST	Flexible authentication secure tunneling
GGSN	GPRS support node
GID	Group Identity
GINA	Graphical Identification and Authentication
GSM	Global System for Mobile Communications
HLR	Home Location Register
HMAC	Hash-based Message Authentication Code
HSS	Home subscription server
HTTP	Hypertext Transfer Protocol
IAM	Identity and access control management
ID	Identity
ID-FF	Identity Federation Framework
IdM	Identity Management
IK	Integrity Key
IMEI	International Mobile Equipment Identity
IMEISV	International Mobile Station Equipment Identity and Software Number
IMSI	International Mobile Station Identity
ISIM	IP multimedia Services Identity Module
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IMSI	International Mobile Subscriber Identity
JAD	Java Application Descriptor File
JME	Java Micro Edition
KDC	Key Distribution Center
LDAP	Light weight directory Access protocol
LFSR	Linear Feedback Shift Register
LMP	Link Management Protocol
MAC	Message Authentication Code
ME	Mobile Equipment
MGW	Media Gateway
MITM	Man -In-The-Middle
MNO	Mobile Network Operator

MoBA	Mobile Based Authentication
MS	Mobile Station
MSC	Mobile Switching Center
MSISDN	Mobile Subscriber Integrated Services Digital Network
NIS	Network Information Service
NIST	National Institute of Standards and Technology
Node B	Base Station Transceiver
NONCE	Number used once
OTP	One-Time Password
OS	Operating system
PAM	Pluggable Authentication Module
PC	Personal Computer
PDA	Personal Digital Assistant
PIN	Personal Identification Number
PLMN	Public Land Mobile Network
RADIUS	Remote Authentication Dial In User Service
RAM	Random Access Memory
RAND	Random number
RES	Response
RNC	Radio Network Controller
ROM	Read only Memory
RFC	Request for comments
RPC	Remote Procedure Call
SAML	Security Assertion Markup Language
SAP	SIM access protocol
SASL	Simple Authentication and Security Layer
SGSN	Serving GPRS Support Node
SHA	Secure Hash Algorithm
SID	Security identifier
SIM	Subscriber Identity Module
SIP	Session Initiation Protocol
SMS	Short Messaging System
SP	Service Provider
SQL	Structured Query Language
SQN	Sequence Number
SSO	Single sign-on
SSP	Secure Simple Pairing
TGS	Ticket Granting Service
TGT	Ticket Granting Ticket
TLS	Transport Layer Security
TMSI	Temporary Mobile Subscriber Identity
TOTP	Time-based One-time Password Algorithm
UE	User Equipment
UI	User Interface
UID	Unique Numerical User Identity
UICC	Universal Integrated Circuit Card
UMTS	Universal Mobile Telecommunications System
USIM	Universal Subscription Identity Module
UTRAN	UMTS terrestrial radio access network.
VLR	Visitor Location Register
XML	Extended Markup language
XRES	Expected Response

# 1. Introduction

This chapter gives the overview of the Thesis. It includes the motivation, problem definition, objectives and how the thesis is organized.

## 1.1 Motivation

Every person in our society has one or several identities. A common person normally has a national identity, an employee or student identity, an alumni identity, a driver license and a set of other digital identities. People are constantly required to prove their identity. While in the real world proving your identity usually means showing a document that identifies you but in the digital world this process can be much more sophisticated. There are digital identity attributes like login, name, etc. that can be easily copied, now to prevent the identity theft some secret credentials known only to a person and an authentication authority have to be used [1] for secure identification of a person. Therefore Identity management is one of the important concepts of the modern society.

Identity management is a nonstop process that encompasses on identity lifecycle management, authentication, and access control [2] among other things. In the perspective of the modern enterprise information systems the main idea of the identity management is to manage access and control to enterprise resources and information. Enterprise information systems provide a range of services to support business processes. With the development of e-business the functionality of such systems grows along with increase in complexity as well, which makes it difficult to provide full protection from unauthorized access. The Authentication process plays an important role, as the decision whether to give access to services or provide resources is based on identity. Therefore, it is important to ensure that an attacker (fraudster) should not be able to steal the identity or masquerade as a user to a system in some other way.

The authentication party (user) presents a proof of identity to the authenticating authority for getting access to the system. The authentication schemes are mostly based on one or the combination of the following factors: something you know (password), something you have (ID card), and something that you are (fingerprints). One of the most common authentication schemes nowadays is a static password authentication. The password authentication schemes have been in use for a very long period of time. The reason for using static password scheme is that they are easy to use and people are used to them. Though, from the security point of view these schemes have many weaknesses.

The constantly increasing computational power of modern computers is making it possible to launch a brute force extensive search attack on password based authentication schemes if the passwords used are weak. As with study of cryptographic protocols and people (hackers) new vulnerabilities of security are coming in observation quite often, thus dramatically decrease the theoretical security level in cryptographic protocols which allows attackers to launch much more efficient attacks than the extensive search. So in strong designed cryptographic schemes the use of longer keys which have higher entropy provides higher security level. Therefore, it is required to use randomly generated passwords with adequate length that consist of a combination of letters, numbers, and special symbols to overcome this problem. Though, such passwords are difficult to remember for humans and this is one of the reasons that usually people choose passwords that are easy for them to learn but can be easily compromised with a simple dictionary

attack. Additionally, the user-name/password systems that the user has to use can be large in number which may result in reusing the credentials for all systems. It is also important to notice that by using strict administrative measures may not strengthen the security of a system as a too strict password policy (difficult to remember passwords or frequent change of password, etc.) can actually weaken it, since users can end up writing down passwords. Use of default passwords or careless users who reveal their passwords either accidentally or as a consequence of social engineering attacks can further decrease the security level of a system.

## **1.2 Problem definition**

There are systems where the ordinary static password based authentication may be sufficient as depending on the sensitivity of activity or data it has, but systems that processes sensitive data requires stronger authentication schemes e.g. Biometric systems, smart card based systems or one-time password based authentication schemes as these are considered to be much stronger than the ordinary user-name/password authentication scheme. However the cost of deployment and maintenance makes these systems less common.

In smart card development, security is a keystone to be considered. The blend of logical and physical security mechanisms makes a high level of security. The ability of storing information (e.g. identity information) and executing cryptographic protocols resulted in a huge success for smart cards to be used in security sensitive areas. The other benefit is that the Mobile network operators have already deployed smart cards to authenticate subscribers, hence having the infrastructure which can be reused to provide authentication in other services.

Thus a mobile phone authentication can provide strong authentication scheme based on the possession of the Universal Integrated Circuit Card (UICC) with a 3GPP application and based on the authentication of the card owner to the card. This solution which uses a mobile phone with a UICC card as a security token provides a much stronger scheme of security than the ordinary user-name/password authentication and at the same time decreases operational costs [3].

As currently there is no workstation logon architecture based on the mobile phones for Linux, thus we need to come up with the idea of using mobile phone based authentication scheme to integrate it in Linux operating system logon process. Thus, a study of related technologies and a detailed analysis of Linux logon process and its internal mechanisms responsible for authentication and logon are done in this thesis.

## **1.3 Objectives**

There is a lot of research done and several solutions have been proposed, using the mobile phone based authentication [3, 4, 5]. The main objective of the thesis is to study available mobile phone based authentication solutions and to come up with the most appropriate architecture that uses mobile phones for Linux workstation logon in an enterprise environment. As in my knowledge there is no available system or a published work that deals with the mobile phone based workstation logon using Linux operating system.

The other tasks that are required for the thesis are as following:

- Study of existing identity management and authentication for enterprise network and workstations.
- Study of Platform security architecture of Linux.

- Study of Authentication and identity management standards and protocols (e.g. EAP, RADIUS/DIAMETER, SAML etc.)
- Design and implementations of solutions for integration of the mobile phone with the workstation logon processes in different operating systems
- Threat and vulnerability analysis of proposed solution(s).

## 1.4 Organization of the thesis

The document has been organized in the following way:

Chapter	Description
Chapter 1: Introduction	This chapter gives the overview of the thesis and consists of motivation, problem definition, objectives and how the thesis document is organized.
Chapter 2: Background	This chapter gives the background study of the technologies required for the implementation of thesis. It starts with identity and identity management then identity management in enterprises and UMTS, SIM technology, Authentication and Authorization in different operating systems and at the end covering Bluetooth security and the Radius technology. Most of the work in this chapter has been covered in the project and the thesis is the continuation of work done in the project.
Chapter 3: Mobile Based Authentication Systems	This chapter presents the different architectures proposed for the mobile based authentication system. At the end a solution for mobile based authentication system for Linux workstation has been proposed.
Chapter 4: Analysis	In this chapter the functional requirements, non functional requirements, use cases and the activity diagram of the system is discussed.
Chapter 5: Design	In this chapter the component diagrams, package diagram and the class diagrams are discussed.
Chapter 6: Implementation	In this chapter the deployment diagram has been discussed along with the test cases of the system implemented.
Chapter 7: Security Evaluation	This chapter discusses the security aspect of the proposed architecture.
Chapter 8: Conclusion	This chapter discusses the results of the knowledge obtained from the thesis with the future work.

## 2. Background

To come up with the solution for the mobile based Authentication for Linux workstation logon process, a study of relevant technologies utilized in these solutions have been made. First the concept of identity has been explored; thereafter a study of different identity management systems in enterprises has been performed. As the mobile phone contains a UICC smart card, it is important to study how smart cards can be helpful to provide a secure storage and execution environment. Nowadays, different authentication schemes use Bluetooth to connect the mobile phone with the computer. Therefore, it is required to study whether Bluetooth can provide a sufficient level of security for using it in the workstation logon solution.

The study of identity management in UMTS is done as many mobile phone based authentication schemes uses GSM/UMTS secrets and identities and they rely on a mobile network encryption. The authentication and authorization in different operating systems is analyzed, as it was important for the thesis. At the end Remote Authentication Dial in User service (RADIUS) has been explored.

### 2.1 Identity and identity management

#### Identity

Every object around us has some characteristics which make easy for us to recognize them e.g. shape, weight, size, height etc. Identity can be defined as: *“The similarities and differences between objects that make them unique to identify.”*

The word identity is easily understood by everyone but at times it is difficult and confusing to understand when we consider the same characteristics of objects at different time and place.

Identity in Logic: It is defined as the relationship between a thing and itself i.e. it can be defined with a predicate “=” such if “ $x=y$ ” is true when  $x$  is the same thing as  $y$ . Identity is transitive (if we have  $A, B, C$  where  $A=B$  and  $B=C$  than  $A=C$ ), symmetric (For  $A$  and  $B$  if  $A=B$  then only  $B=A$ ), and reflexive (For every  $A$  is  $A$ ).

The **Identity of Indiscernibles** states that: *“No two distinct substances exactly resemble each other”*. This means that: *“There does not exist two objects that can have exactly the same properties or characteristics”*.

There are some laws that explain the above rule as if two entities are numerical identical then they must be qualitatively equal (same properties) and similarly if the two entities are qualitatively equal then they must be numerical identical. There are problems with these laws like the symmetric of the world, the infinity problem, the impact of quantum mechanics as discussed in [1].

#### Personal Identity

Identity of human beings is more complex. The personal identity is about the characteristics that make them unique and recognizable. Personal identity is also about ones knowing himself as well. There are certain issues that are relevant to personal identity are as follows:

- What should be the characteristics that are necessary to be a person?

- What are the conditions that are necessary to make sure that same person exists in different times?
- How a person can be recognized?
- How a person is unique from others?
- How people should perceive a person?
- Can a person change himself?
- Who is the model of a person whom he wish to be?

As there are some human characteristics that change with time like weight, height, so then how we can make sure that the same person exists in different times? This question can be answered with the following approaches:

- **The Psychological Approach:** It is about the psychological things like memories, beliefs. So according to this approach it is necessary to for a person to have psychological continuity to exist in different time.
- **The Somatic Approach:** This approach is about the physical relation. So identity of a person in different time consists in the identity of his body.
- **The Simple View Approach:** The approach says that the person in different can only exist in different times if they are identical.

## Citizen Identity

Governments throughout the world are introducing digitized personal identification and authentication systems into their service relationships with citizens [6].

This has improved the efficiency of and effectiveness of public service provision as discussed with case studies in [7].

The citizen identity depends on physical characteristics and needs physical continuity for identification. There can be infinite set of characteristics so we need to restrict to a few characteristics to make the process easy. The characteristic properties are called Attributes [1].

- An **attribute** is a characteristic attached with an individual. An attribute can be **intrinsic** (by nature) like finger prints, eye etc. or **extrinsic** (acquired from outside) like name, address etc. Attribute can be persistent or temporary.
- An **identifier** is an attribute that is most representative for an entity with in a context. E.g. a bank account and a person have identifiers. The person can be associated with account with extra information like social security number.
- **Personal Identifiers** are the unique attributes associated with a human being that are impossible to change like biometric attributes.
- **Biometrics** is systems used to recognize a person on physical attributes like fingerprints, iris, face, retinal, veins.
- An **identity** is a set of attributes that are permanent and associated with an individual which are unique and always make possible to recognize a person.
- **Identification** is a process of associating personal identification to an individual who have some attributes.



- **Authentication** is a process of proving the association between identifier and the individual.
- **Authorization** is a process of deciding to give access to which action on the basis of identifiers or attributes.
- An **identification document** or **identity card** is a credential used to verify the person identity. It may include name, age, profession, sex, rank, religion and new technologies are adding the biometrics as well. Passport is one of the example of identification document.
- The ID cards are expensive but one of the serious concerns is privacy with electronic ID. Electronic ID cards can be used to track movements of a person.
- **Digital Identity** is representing the identity of an individual in computers. Like we do online banking transactions the bank needs to confirm the identity of the user, so for that reason we need digital identity that may consist of name, password etc.

One of the problems that are arising is to have many login names and passwords that each user is getting. To solve the problem single sign on solutions are introduced to sign in once and have access based on user credentials to all processes without entering the credential again and again.

## Identity Management

Identity management is one of the hot areas for growth as it is expected to rise to more than USD 950 million by 2009 [2].

Identity management can be defined as: *“a set of data management systems and practices designed to increase confidence in the identity of individuals where appropriate”* [8].

Identity management can also be defined as: *“a discipline that consists of processes, policies and technologies to manage the complete life cycle of user identities across the system and to control the user access to the system resources by associating user rights and restrictions, these resources include information, services, process capability, buildings and physical asset”* [2].

The main benefits of Identity management systems are as following.

- It allows having control over user-to-application interaction which makes easy for auditing and reporting.
- It helps in lower operational costs.
- We have enhanced security of assets with the use of Identity management systems.
- We have good productivity. Users can access from outside the enterprise securely.
- Different companies can give access to specific services securely without having to expose the whole system to outsiders.

There are some functions that are performed by an Identity management system are discussed as follows:

- **Identity Administration:** This function includes the management of the digital identities in the system which can be creation of digital identity (fingerprints, name, address etc) then maintaining it and termination if required.

- **Access control:** This functionality consists of providing the necessary access to the resources when required and is enforced by using access rights. It can be giving deletion, writing access to the user for some data.
- **Authorization:** This process shows what resources the user is allowed to access. The permission to access is granted by the authority.
- **User Self Service:** This functionality includes having such process to enable users in the administration of his identity like resetting password etc.
- **Auditing:** This is one of the important functionalities of Identity management systems. It includes analysis of all the process which helps in making the system secure e.g. it may include digital forensics for determining the access of and illegal user.
- **Identification:** This is a process of recognizing the users at the login.
- **Authentication:** This is the process to make sure that the person who claims to be user is the real one.
- **Single Sign-on:** There are certain processes that requires authentication at different stages and many times, so single sign-on helps to access all resources at once and can help in increasing productivity.
- **Federation:** The achievement of this process is to enable users of one domain to access the services, data of another domain securely which will help in removing redundancy and at low cost.
- **Reporting:** This process consists of reports that are generated by auditing. The report should be meaningful for the users that may be the owner.
- **Directory:** It stores the identities. It can include Light weight directory Access protocol (LDAP), relational databases, flat files and other types of data stores.
- **Meta Directory:** It provides the organizational view of the information in the heterogeneous directories and other storages.

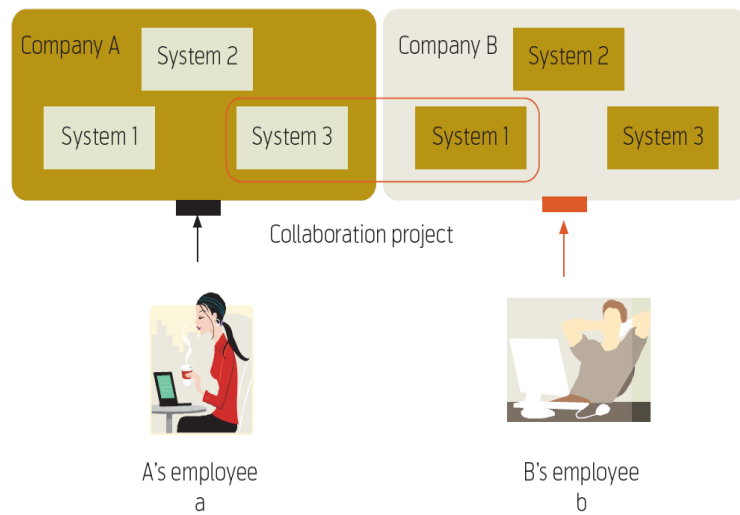
### **Enterprise Identity Management**

The organizations usually have different departments. Users in each department are given identity like user name and password which is valid for that department. If the user is given some task in another department for some time, he has to be registered in that department which can cause serious problems if he is not deleted from there than and difficult for user as well to remember many passwords and usernames. So to solve this problem it is recommend having a centralized Identity management system and the user is granted access to the resources according to the company policies, thus reducing cost, time and making it more secure.

### **Partnership Identity Management**

Nowadays many companies have to make collaborations with other companies and terminate them as soon as possible for survival. Collaboration between companies requires sharing of data and resources, so it would be better to have control over the access given to the users of other company to avoid any serious damage. So it would be to have an identity management system

that allows the flexibility of adding the users of other company and easily assigning the required services securely without damaging the ongoing system. Figure 1 explains that system 1 can access the data of system of another company.



**Figure 1: Partnership Identity Management [2]**

### Customer Relation Identity Management

In this kind of Identity management it is encouraged to give the management task to the user as maximum as possible, which will help in reducing management cost. It needs careful evaluation that which system the user is allowed to access and which not to for security reasons.

### Identity Management as Commercial Service

Identity management is a complex process which requires resources and knowledge. Companies don't have that much resources so they go for outsource the identity management.

Three parties are involved in this process. One is an **Identity provider** that manages the identities of the users. They issue and validate the identity credentials.

Second is the **Service provider** that provides the services for the end user. They are the relying party. They provide the services to the user on the identity credentials validated by the identity provider.

Third is the **User** who is using the services and proves his identity by his credential like user name and password.

So for a company who provides services have not to worry about Identity management. He works as a relying party and the Identity management is provided by the Identity provider, reducing his management cost.

## 2.2 Identity management in enterprise systems

To provide a range of services for supporting business processes, the enterprise information systems are made. The Business objectives are fulfilled by the development of e-business, enterprise IT systems and that is the reason behind the growth of functionality provided by IT systems. Yet, the complexity and vast functionality of these systems makes it harder to provide

protection against the unauthorized access to the enterprise information. The Access control management is required for outsiders, semi-trusted parties like partners, customers and insiders. For managing the access to the enterprise resources, access control system strongly depends on the identity management system. Identity and access control management (IAM) has to meet the requirement of business to have external links with partners, suppliers, customers and other people. Thus, managing the identities of corporate users is not enough; identity management goes beyond the boundaries of single enterprise.

The other fact that makes it more complicated is that many organizations have heterogeneous IT platforms, and each has its own identity database. The centralization of identity management will simplify the management tasks, will reduce the management costs, and decreases the risk that something will not be taken into account during data flows planning and privileges setup [10].

Additionally, IAM systems must fulfill the government legislation in order to provide specific services like processing credit card payments [10].

### **Centralized identity and access control management**

With Centralized identity and access control management approach the identities and access controls for many heterogeneous systems with different repositories in an enterprise can be managed with a single interface. A directory service forms the basis of an identity and access control management system as it stores the information about the state of an enterprise system. This state information consists of identities, account-related information, policies, roles, groups, workflows etc. A directory is a unique type of database which is optimized for read operation, whereas the other databases are optimized for the write/modify operation [11]. In addition, directories are mainly accessed through Lightweight Directory Access Protocol (LDAP) and databases utilize Structured Query Language (SQL) for this purpose [11]. If we use only one server that stores all data in one location, then we can say that the directory is centralized. The directory can be distributed as well, if there is more than one server and information has been replicated between servers in order to have all the same data or it is divided between the servers so that each can hold certain part of the data.

It is difficult to manage different repositories of identities and access control rules which can also cause mistakes that can lead to weakness in security. Synchronization and consolidation of data will decrease the management costs and will make the whole system much more easily manageable. There are different methods which can provide an integrated enterprise directory service. The options can be a single directory, a meta directory or a virtual directory [10].

**Single directory:** In this method there is only one directory of identity information for the whole system. A single directory will simplify the management tasks as compared to a bunch of different repositories; however there have to be some modification in some applications to be able to work with a single directory [10]. Uniqueness is not in physical terms but in logical. In order to not create a single point of failure and share the load a single directory can be distributed in to several servers having the same data. Legal, political or security issues may create a barrier preventing the creation of a single directory [12]. In this case a single interface will be provided by the IAM to manage all these systems.

**Meta directory:** In this method all the information from different repositories is copied into a single directory with a unified namespace. The synchronization between the meta directory and satellite/original repositories is controlled through bidirectional synchronization mechanism if

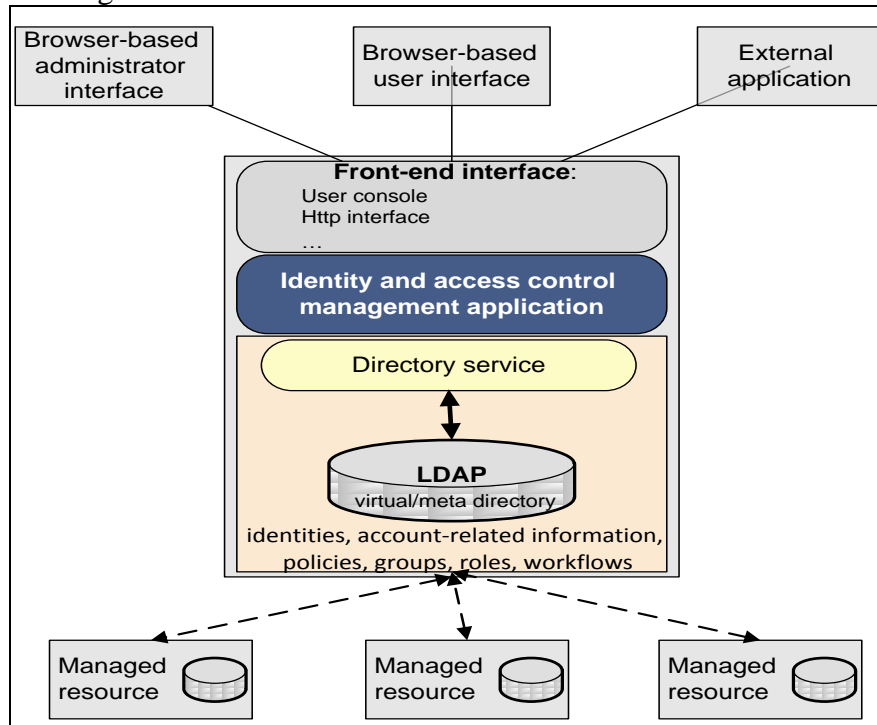
there is some change occurred in the data [12]. With using this approach the requirement of modification of applications will not be needed, that will work with the particular repositories as these repositories will remain in the system [10].

**Virtual directory:** In this method the virtual directory will serve as an abstraction layer between various repositories and applications by providing a single logical directory which will gather the information from all the repositories in real-time. The logical presentation of data can be modified for each application. By using the same input data different application-specific views of data, optimized for application needs can be derived [13]. There is no central physical directory which will contain a copy of all data.

The main functions of IAM are:

- One of the main tasks is managing users and accounts which include creating, editing, deactivating, and deleting users, setting/changing user passwords, etc.
- The other task is policy and workflow based management which helps to automate management process. The Policies which can be used in an enterprise consists of account provisioning, password, authentication policy, etc. The workflow can be defined as a predefined sequence of automated processes that automates some time-consuming actions like gathering approvals. Workflows also enforce consistency (it is the sequence and the set of involved actions to be constant) and completeness (it is to not start some action until all previous are completed successfully). The Request for approvals always takes the same predefined path and is automatically delivered to the person in charge.
- The functions of IAM include Privileges and Access control management for setting permissions to make sure that only those entities that have permission can access the resources. The common models used for the access control are Discretionary Access Control (DAC), Mandatory Access Control (MAC) and Role-based access control model.
- The IAM functions include Enterprise single sign-on mechanism that enables a user to authenticate once to a system and can allow user to reuse this authentication for many enterprise applications during the current session. In this case the requirement to have different accounts for different service providers and remember corresponding passwords is not needed. A user gets enterprise wide access to data at once.
- The IAM also provides strong authentication which imply the usage of a two factor authentication. The Technologies such as smartcards and biometrics can be used to provide two-factor authentication. According to [12] SIM-based strong authentication that uses mobile phones can be quite cost-efficient as compared to the other two-factor authentication solutions, and it is comparable in price with password-based authentication.
- The IAM provides Monitoring and audit services in which system events are automatically checked against policies and rules for violations. If the violation is detected then some action is triggered like account blocking.
- The IAM provides Federated identity management in which user authenticated identity information is communicated across security domains to trusted partners that exist in the same Circle of trust. Thus, a user is not required to authenticate when he accesses the resources of a collaborating company.

Identity and access manager architecture:



**Figure 2:** Identity and access manager architecture

A standard administration and management toolkit for a single directory can be used without using IAM software. But all the functions that are automated in an IAM suite have to be done manually. IAM suites can provide a benefit that they can define policies that automatically create user accounts, mail boxes, and group memberships in real-time. Moreover, creating a single directory is not always possible.

### Centralized authentication and trust

At the early phase of the development of authentication solutions, the use of decentralized autonomous approach was common. In a system every station performs the authentication and authorization autonomously by maintaining its own user's file or user's database, and the access to the services and resources was provided to that station depends on that file. To allow users to use services provided by some server the administrator had to add information about those users to the users file of the server. A user is required to have an account on every station for the services and resources he uses. This kind of management approach had serious scalability issues. In case where different passwords were used for different accounts so the remembering of all passwords and the synchronizing password changes between many accounts in case one password was used for many accounts makes both these approaches a big problem in management and security. All identity and access control management procedures requires a lot of work and were prone to errors.

The solution for the problem was to use a trusted third-party authentication approach. In this method the third-party authentication authority maintains a centralized repository with identities and account relevant information. Each user is given only one account which is maintained by the third-party authority. The authentication is responsibility of trusted third party and is solely

done by it. The rest of the systems trust it. The third-party should be highly secure thus can be trusted; otherwise the compromise of the third-party would lead to the compromise of a whole system. The third-party authority can be a single point of failure; if it crashes the whole system will stop working. There are two third-party authentication based schemas [14]:

- Implicit authentication schema: In this schema an authenticating entity (ex. Service provider) does not request authentication service from a centralized authentication authority. The authentication is cryptographically produced from the encrypted message given by the third-party to the entity that is being authenticated. E.g. Kerberos v5 protocol uses this kind of approach.
- Explicit authentication schema: In this schema an authenticating entity explicitly requests third-party to make authentication.

There are many technologies which can provide authentication and access control, but among all the preferred are Kerberos security service and LDAP directories [15]. Kerberos prime task is to provide authentication but it can also provide some simple authorization services. Alternatively, LDAP directories are primarily used for storing and managing authorization data, but they can also provide some authentication services [15] as well. And this is the reason that these technologies can be used separately or can be integrated in one system.

### **Kerberos protocol**

The Kerberos protocol deploys implicit authentication schema. Kerberos version 5 which is defined in RFC 4120 is the present version of the protocol. Kerberos performs mutual secure authentication in the network which is not secure, however it does not provide accounting and offers very basic authorization [16]. To provide secure authentication service Kerberos uses secret key cryptography. The three types of entities in the Kerberos architecture are as following:

- Clients: They are the one that wants to use the services provided by the service providers.
- Service providers: They provide the services for clients to use.
- Kerberos servers: They manage Kerberos authentication. They are known as Key Distribution Centers (KDC). A secret key is shared by the KDC with every principal (a client or a service provider) in a network. The service providers and clients trust KDC. KDC is composed of three different elements. Authentication server that serves the client authentication requests. The Second element is the Ticket granting server whose job is to issue Ticket Granting Service (TGS) tickets to clients, and the third element is the database that stores identities, secret keys, policies, etc.

The security in the Kerberos schema is based on tickets and the corresponding authenticators. A ticket is an encrypted message that contains the client name, session key and the life time of the ticket. The encryption of the ticket is done with a secret key shared by a server and KDC. The authenticator consists of an encrypted client's name, client's realm, timestamp. The encryption of authenticator is done with the session key that is in the corresponding ticket. A client can authenticate by providing the ticket and the authenticator to a server.

The KDC is trusted by all principals in a realm which is an authentication domain with some administrative boundaries. The management of a realm is done by only one KDC (to share the load a KDC can be distributed between several servers, but logically it is one). Each realm has its

own KDC database. The principal identity is formed with the Name and the Realm parts.

### Kerberos authentication model

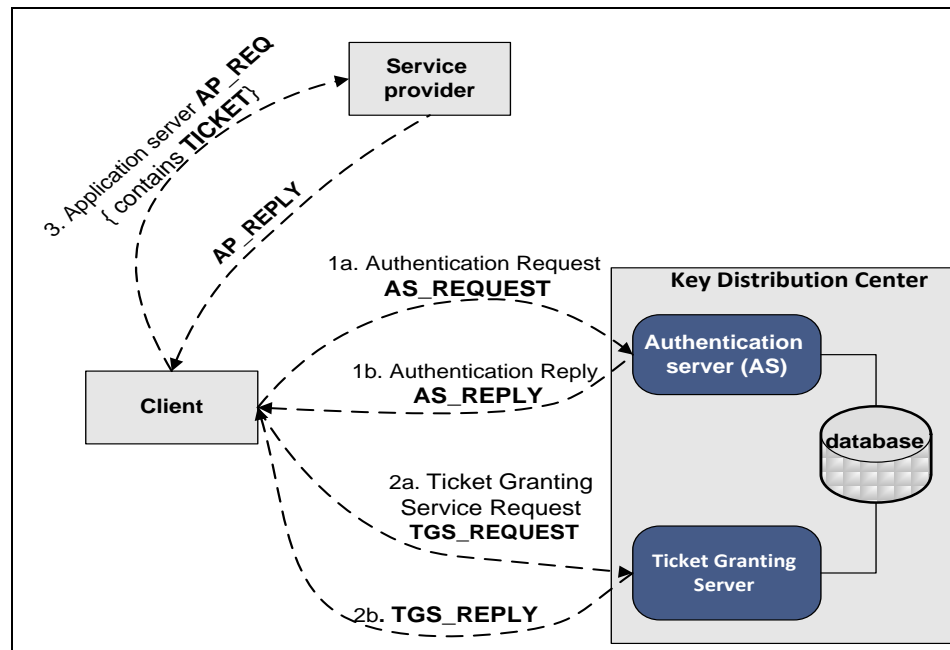


Figure 3: Kerberos authentication model

The Kerberos authentication procedure consists of three distinct exchanges which are discussed as following:

- **The Client & Authentication server exchange:** The Client authenticates the KDC and obtains a Ticket Granting Ticket (TGT) which will be used to get the credentials for authenticating to a Service provider. The client can be authenticated by the Authentication server before issuing the TGT (pre-authentication), or it can wait until the Client – Ticket granting server exchange [15]. From the security prospective it is better to make pre-authentication, because this will provide a protection against the attacks on the principal’s secret key [17] that is used by KDC to encrypt the response.
- **The Client & Ticket granting server exchange:** The Server authenticates the Client (this authentication is done in every case) and grants a Ticket Granting Service (TGS) to the Service provider which is specified in the request by the client. The TGS granted will be used in the Client – Service provider authentication.
- **The Client & Service provider authentication exchange:** In this procedure the Client sends a request to the Service provider which normally contains the authentication information and the initial request [17]. The Client at all times authenticates to the Service provider, but the client must requests for mutual authentication explicitly [17]. The TGS and the authenticator that are in the Client’s request facilitate the Service provider to authenticate the Client and obtain a shared session key which can be used for further protection of communication with the client [17]. If mutual authentication is requested by the Client then the reply of the Service provider contains the authentication information that enables the Client to authenticate the Service provider.



The main services that are provided by the Kerberos are as following:

- Mutual Client & Server authentication: In this process a client is able to authenticate both KDC and various Service providers which will help by not allowing a malicious server to deceive a client and obtain confidential information.
- Centralized management of authentication information: The Kerberos uses a centralized database for storing identity and account information.
- Delegation: If a principal wants to allow a service provider to perform operations on its behalf e.g. a client can assign rights to a printing server to access client's files on a file server to print them out [17]. A principal can ask the KDC for a new TGT with different network address that will allow a service with that network address can act on behalf of the principal. The principal is required to transmit the new TGT and the corresponding session key to the service provider to facilitate delegation.
- Single sign-on: The Kerberos supports single sign-on with caching tickets and the corresponding session keys. Thus the next time the authentication is needed, a user is not required to type in the password again. The cached credentials are used on user's behalf.
- Cross-realm authentication: In this process a client in one realm can authenticate to a service provider from another realm. For establishing an inter-realm communication the service provider's KDC should be registered as a principal in the client's KDC. After that the client asks its KDC for a ticket to the service provider's KDC. It presents this ticket to the service provider's KDC and requests for a ticket to the service provider. It is possible to pass through several realms to authenticate to the remote service provider. The Inter-realm communication can be controlled hierarchically. So the authentication path through multiple realms can be easily created [17].
- Multi-factor authentication: the traditional Kerberos authentication is done with the secret keys (for a user the secret key is derived from a password). Though, the public key cryptography for initial client – KDC authentication (PKINIT which is defined in RFC 4556) can also be used which makes it possible to use smart cards and other cryptographic tokens when authenticating through Kerberos [18]. PKINIT needs the usage of trusted by KDC and its principals Certification Authority. We can use One-Time Password mechanism and biometric scanners for the initial authentication.

There exist several different implementations of the Kerberos Protocol e.g. MIT version, Heimdal version, Windows Active Directory version etc [19]. Active Directory does not simply use Kerberos as its default authentication protocol, leaving the NTLM for compatibility [20]. But it tightly integrates it in its framework which results in some issues in environments with Windows and non-windows systems [18].

### **Explicit authentication schema**

The authentication in this schema is entrusted to a third-party that has all the identity information. The typical authentication procedure is discussed as following:

- a client request a service from the application server and it forwards its credentials (identity and password) to the application server
- the application server requests the trusted third-party server to carry out the authentication

It is required to cryptographically protect the exchanges between the client and the application server and between the application server and the third party.

To provide a centralized third-party directory service for storing identities and account-related information, the technologies like Network Information Service (NIS), NIS+, and directory services with Lightweight Directory Access Protocol (LDAP) interface are used. Normally the NIS is used to centralize the storage of */etc/passwd*, */etc/shadow*, */etc/group* files from all stations in the Unix-based domain, and after that this information can be accessed by clients [21]. The changes in the centralized directory are also propagated to all source stations. From administration point of view the NIS is easy to manage, but it lacks the security mechanisms. It does not provide authentication and authorization for directory access and the communication is unencrypted [16].

Both NIS and NIS+ protocols are based on Remote Procedure Calls (RPC). NIS+ is advance version of the NIS protocol. Besides providing hierarchical namespace, NIS+ offers a stronger security mechanism. Though NIS and NIS+ are legacy technologies and it is recommended to transfer to LDAP [22].

### **Authentication against LDAP server**

The Lightweight Directory Access Protocol (LDAP), defined in RFC 4511, is a protocol that accesses the directory services which comply with X.500 standard. There are various directory servers that provide LDAP support e.g. Active Directory, IBM Tivoli Directory Server, OpenLDAP, Novell eDirectory, Sun Java System Directory Server etc.

LDAP is a client-server protocol which runs on the top of TCP. It is recommended that the servers use port 389 for an incoming request. A client requests to a server for performing some operation in the directory. The server responds to the request, performs the operation and returns a response. Some of the operations used in LDAP are bind, unbind, add, delete, search, modify, compare and startTLS.

The Bind operation is the same like authentication. The Bind request specifies the authentication identity. The Bind operation uses different authentication methods that are simple authentication method and SASL authentication method [23]. The simple authentication method is further decomposed into the following methods that are anonymous authentication, unauthenticated authentication, and name/password authentication. In the name/password authentication, for validation the client sends both the name and the password to the server. This method of authentication should only be used in environments where confidentiality protection is provided [23]. A TLS can be established for the LDAP session by sending a request with StartTLS operation by the client. The TLS will provide confidentiality and integrity protection for LDAP session, so a simple name/password authentication can be performed in a secure manner.

The Simple Authentication and Security Layer (SASL) defined in RFC 4422, is a framework which enables the use of various security mechanisms in protocols. SASL provides the abstraction layer which allows any protocol to use any mechanism. By using SASL

authentication method LDAP allows authentication through any SASL mechanism [23].

An indication of a success/failure of the authentication request is communicated by the Bind response message. LDAP has become one of the key elements in enterprise identity and access control systems [19]. It can provide a centralized storage for identity and account-related information and can be utilized to authenticate principals. The procedure of authentication starts by the client sending its identity and password to the application server over the protected channel. E.g. protection can be provided by TLS/SSL. If the LDAP server is configured in such a way that the principals need to authenticate to it, in that case the application server sends the Bind request to the LDAP server using the credentials of the client. If this authentication succeeds (in that case the LDAP server sends a Bind response message with success status), then the client is considered to be authenticated by the application server. The other simple solution for the application server is to simply retrieve the client's identity and password from the directory and compare it with those received from the client [24]. For using this solution it is required that the application server should authenticate to the LDAP server prior to the information retrieval and the application server should be approved by the LDAP server to perform those actions.

### **Enterprise Single sign-on**

The mechanism of Single sign-on (SSO) enables a user to authenticate once to a system and can reuse this authentication during the current session. The technologies that can use to provide a single sign-on can be divided into three main categories [25] that are ticket-based, cookie-based, and PKI-based.

In the Ticket-based SSO the user authenticates to the authentication service and in response receives a cryptographic ticket. This ticket can later be used to authenticate to service providers. Kerberos is a one of the examples of ticket-based SSO system.

In Web-based environments the Enterprise SSO can be achieved through cryptographically protected HTTP cookies. The Sun OpenSSO Enterprise 8.0 uses this method for providing a SSO solution. The process used by OpenSSO involves the following main steps [26]:

- The user request to a service provider by sending a HTTP request. The policy engine intercepts the request that protects the resource. After exploring the request and if it does not find an HTTP cookie then the policy engine redirects the user to another URL for authentication.
- The browser follows the URL and sends a new HTTP request to OpenSSO Enterprise authentication service. The authentication service using one of its authentication modules, e.g. LDAP authentication, validates the user's credentials. The HTTP response which contains a cookie that carries an encrypted session token is sent to the client. The HTTP response redirects it to the original location.
- Again the browser sends an HTTP request to the service provider for one more time. As this time the request contains the cookie with the session token. The policy engine receives the request and it checks it for the session token. The check is done by contacting the OpenSSO Enterprise service. OpenSSO Enterprise decrypts the token and it inquires whether the session data related with the session token exists. The policy engine receives a response defining whether the token is valid. After the

validation of session token the policy agent makes the decision whether the user should be granted access or not.

- When the user contacts some other service provider next time, the cookie with the session token is incorporated in the request. Thus the policy engine that receives the request needs only to certify the token. So the second authentication procedure is not required from a user in this case.

The Public key based SSO needs to use the public key infrastructure. The trusted third party certification authority is in charge for examination of user's credentials and issuing certificates. The certificate and the user's private key can be stored on a user's computer or on some cryptographic token [27]. The user is authenticated by the service provider itself, the certification authority checks the identity of the user only while issuing certificate. The PKI infrastructure also provides the service of non-repudiation which is considered quite important for business [25].

### **Identity federation**

Now days the business-to-business communication is carried out by the extensive use of internet technologies. The Business processes in a company require external connections with partners, suppliers, contractors, clients, etc. Though, creating and managing the accounts for external users locally is not an efficient solution in the perspective of security, management and operational cost [28]. Identity federation enables the inter-organization identity and management sharing and secures the external access to the required set of company's resources.

Identity federation depends on the trust relationship between collaborating organizations. So, one organization trusts in authentication made by another organization for users. The trust relationship is the basic requirement that makes the Single Sign-On service for cooperating organizations possible. The organization that is responsible for maintaining and managing the user's identity information is called the identity service provider. The functionality of service provider is to provide some service. A circle of trust is formed by at least one identity provider and a group of service providers that trust in this identity provider. There are many organizations that play the role of service provider and identity provider at the same time [4], but for some companies it is favorable to outsource identity management to an identity provider as this can help in focusing on the services they provide rather than the identity management.

Identity Federation enables flawless interaction between different organizations with completely different and independent environments. It is not required for Collaborating organizations to have similar security systems or have in depth knowledge of systems used by the partner [12]. There are many frameworks which define federated identity management; some of the examples are Liberty Alliance Identity Federation Framework (ID-FF), Web Services Federation (WS-Federation) and Security Assertion Markup Language (SAML) framework. The SAML v2.0 is considered as a favorite solution [26] as the Liberty ID-FF and SAML v1.x were contributed to OASIS consortium and are the foundation of SAML v.2.0. WS-Federation is an alternative solution for interacting with the Active Directory Federation Services [26].

SAML [29] is an XML-based framework which is used for exchanging identity, corresponding attributes and authentication information between collaborating organizations. The Information exchanged is expressed in the form of SAML assertions. Assertion consists of a set of statements about a principal (which can be a user, computer or company etc.). The types of statements that a

SAML assertions can contain are as following:

- **Authentication statements:** It is created by the party that successfully authenticates a user. It describes the authentication mechanism and the time of authentication, etc.
- **Attribute statements:** It contains a specific principal's attributes like e-mail, telephone number etc.
- **Authorization decision statements:** It describes the authorized actions

The SAML assertions are sent in a SAML protocol messages (both assertions and protocol messages consist of XML documents). The SAML bindings explain how protocol messages are carried by the underlying transport protocols. The SAML defines SOAP-based bindings and HTTP-based bindings. The security protection for message exchange is not provided in SAML protocol [30]. It depends on other protocols like TLS/SSL or IPSec for providing the functionality of confidentiality and integrity protection. The security can also be provided by XML encryption and digital signatures.

The foremost use cases of SAML are Single Sign-on and Federated identity. A common SSO scenario is when the client uses a browser application and sends a request to a service provider. For authentication the service provider redirects the user to the identity provider. An assertion is issued to the user after the authentication by the identity provider which will be used by the service provider to authenticate the user for granting access to the resource. Any requests from the same user in the same session for other service providers in the same circle of trust do not require the user to repeat the authentication procedure again.

A federated identity use case is a user is required to have a federated identity and when the collaborating organizations agreed on how user will be referred [29]. This shows that the organizations have the same meaning of the identity of the user which is referred in the message exchange. When the identity is federated then the information about that specific user can be shared between different organizations.

There are many factors on which an Identity federation model depend e.g. some of the factors that influence identity federation are like whether the exchange of identity attributes about users should be allowed, whether the users have existing local identities, whether the temporal identifiers that are used for identity federation should be destroyed after session termination etc.

Some of the use cases defined by the SAML for identity federation [29] are as following:

- **Federation through Out-of-Band Account Linking:** In this case the identity federation is established without the use of SAML protocol e.g. it could be done through the database synchronization.
- **Federation through Persistent Pseudonym Identifiers:** In this case the permanent SAML pseudonym identifier is used for dynamically establishment of identity federation during the web SSO exchange
- **Federation through Transient Pseudonym Identifiers:** In this case a temporary identifier is used to temporary federate identity till the termination of user's web session. The advantage of using this approach is that an organization is not required to manage the local accounts of users from a collaborating organization.
- **Federation Termination:** the elimination of an existing federation.

The use of Identity federation enables the organizations to provide access to external users from the collaborating organizations without managing these accounts locally. Thus, simplify the identity management, reduce the administrative costs and enhance the security. In addition it also provides the advantage for external users in presenting a web single sign-on service. SAML is the standard solution which supports both the identity federation and single sign-on.

One of the important things for business environment is Identity and access management. In this mechanism the enterprise resources are protected from an unauthorized access by the inside and external users. Business objectives require the extreme usage of internet-based applications and a limited and secure access to the enterprise resources for external users in the collaborating organizations which can be partners, contractors or clients etc. The identity and access control management tasks are complicated by the variety and complexity of systems used by the business. The use of centralized storages for identity and account-related information and centralized authentication makes the things more manageable. But the possibility of always having a single directory is less because of the administrative, security and some other issues. For this purpose many identity and access management suites are developed to provide a centralized interface for management.

## 2.3 Smart card technology

Smart card technology is considered one of the important technologies for the modern information systems. There is huge deployment of smartcards because of the widespread use of smart cards in mobile phone networks, international payment systems e.g. MasterCard and Visa and transport and ticketing systems. Other than this the smart cards are also widely utilized in identification and access-control systems. The main reason for this success of smart cards is in the security services provided by it.

### Smart card definition and types

The definition of a smart card can be as a card that's size is about of a credit card and it contains an embedded integrated circuits. We are not considering the magnetic stripe cards which can be used only for information storing as they do not provide sufficient level of security and can be easily forged [31].

There are different types of smart cards with different functionality and purpose which are discussed as following

**Memory chip card:** The purpose of this type of cards is to store the information. They normally do not have on-board processing facilities. There are almost no security gains of Memory chip cards when compared to the magnetic stripe card [31, 32]. Although there is benefit of memory chip card of containing one-time-programmable memory which can be written once and cannot be rewritten later [33] as compared to the magnetic stripe card, but it is still easy to read the stored value and make the copy of the cards [31, 33].

There wired logic-integrated smart cards are more sophisticated memory cards that provides write/erase protection and a restriction on read access as well [34]. They have few predetermined extra functionalities and redesigning of the chip is the only way to alter the available small command set [32]. Even though the arithmetic logic unit is very limited but still it is able to perform simple cryptographic operations for the authentication and data encryption. E.g. the authentication of the reader based on the stored keys with encryption of all subsequent memory

operations is provided by MIFARE classic [35].

**Microprocessor chip cards:** It contains a microprocessor, an operating system, different types of memory and I/O circuits in this type of smart cards. For acceleration of execution of cryptographic operations, the smart cards may optionally contain a crypto coprocessor. The types of memory cards that Smart cards contain are Read Only Memory (ROM), Random Access Memory (RAM), and Electrically Erasable Programmable Read Only Memory (EEPROM). ROM contains the data which is stored during the manufacturing process and cannot be modified; the only operation allowed is read during the card operation. In ROM the operating system of a smart card is stored. RAM is a volatile type of memory as it is used as dynamic data storage and can lose its content on power shutdown. EEPROM is a non-volatile memory, which means that the data is still saved when power is off. EEPROM is used for the data and application program codes. The major problems of EEPROM memory is that it has limited number of write cycles, even though it can be read for unlimited amount of times. There is a slow gaining popularity of other memory types like flash memory with shorter write access time and longer lifetime as compared to EEPROM in smart cards [31].

There are two different chip card interfaces of the Smart cards which are contact and contactless. The Contact and contactless smart cards are standardized in ISO/IEC 7816 and ISO/IEC 14443 standards respectively. The major difference between the contact and contactless interfaces is that the contactless reader produces energizing radio frequency field for energy transfer to the contactless smart card through air and the modulation of this field enables transfer of data. Therefore, embedded antenna is available in contactless smart cards. There are some cards that have dual-interface which means that it have both the contact and contactless interfaces.

As defined in ISO/IEC 14443 the operational frequency for contactless operation is 13,56 MHz. The Contactless smart cards can be further divided into proximity and vicinity smart cards. There exists a limitation with Proximity smart cards that they must be in a close proximity (up to 10 cm) from a reader to function properly. The vicinity cards standard is described in ISO/IEC 15693. The vicinity cards operational range is up to approximately 1 m [33]. As according to the ISO/IEC 14443 the data rates supported for contactless smart cards are 106, 212, 424, and 848 Kbit/s [31]. The vicinity cards provide greater operational distance but the data transfer rate is lower than as for proximity card which is only 26.48 kbps [31].

The wireless interface introduces a honey spot for attackers to attack on smart cards which include but not limited to attacks like eavesdropping, denial of service and man in the middle. Though according to [34] contact and contactless cards provide the same level of security if the threats particular to contactless interface are taken into account in the security architecture of a smart card.

### **Security provided by smart cards**

One of the basic functions that a smart card should provide is secure storage for data [36]. Thus, a non-amendable memory is of great importance to a smart card security as it can be used for storage of system secret keys [33] (typically top keys of the key hierarchy). The other approach that can be used instead of storing the system secret key in ROM is to compute it on the basis of a unique chip serial number that is stored in the ROM [31, 33]. The secure microprocessor is the important factor of the smart card security system. The word “secure” is used where we mean that the microprocessor is protected against physical and side-channel attacks. One of the main

reasons of introducing the microprocessors in smart cards is of security reasons as the microprocessors made the cryptographically protected communications possible [34]. From the security perspective the main functions of the microprocessor is to generate a pseudo-random numbers for crypto protocols, generating the temporal keys, digital signature generation, encryption/decryption and performing the operating system security checks like checking whether the access to smart card resources should be granted. As a result along with providing the secure storage, the smart cards can also be used for secure execution of cryptographic algorithms [33].

The access to microprocessor smart card's resources is managed by an operating system which is run on a microprocessor. Therefore the logical level security is provided by the operating system access control system that is responsible for granting or denying the access to smart card resources. The modern smart card operating systems provide the following security mechanisms:

- **Access control:** In this mechanism smart card resource is accessed on the basis of specific access conditions and only allows authorizing the entities. The Access control can also be based on either a state-oriented or a command-oriented access model [33]
- **Authentication:** This mechanism is used for card reader authentication, user authentication (so before performing any operation the user should be authenticated) and authentication of communicating parties
- **Process isolation:** This mechanism is required so a process can't access resources owned by other processes
- **Atomic transactions:** This mechanism is required so either all of the operations that make up a transaction are performed or none of them
- **Secured communication:** It is required so that smart cards should use a secure protocol for communications as to protect the information during transfer.
- **Cryptographic protection:** The smart cards support the use of various cryptographic mechanisms e.g. encryption, hashing, digital signature, random number generation for security of data.
- **Key management:** In smart card the operating system is in charge for secure generation, distribution, usage and destruction of cryptographic keys
- **Security monitoring and audit:** In this process the system events can be monitored and analyzed for any potential security violations
- **Secure data deletion:** It is required so that the data after deletion should not be accessible or recoverable
- **Card locking:** This is required so to have an ability to either temporary or permanently disable a specific application or the whole card

On the other hand an attacker can directly attack the memory or buses during data transfer to get the data and avoiding the security checks of the operating system. Therefore it is required to handle this threat; therefore, smart cards also provide protection against physical attacks.

The Physical attacks on smart cards consist of two types which are invasive and non-invasive attacks. In the invasive physical attacks case it is required to remove the microprocessor from a card so that to get direct access to it. Sophisticated equipment is required in this kind of attacks (e.g. a microscope, laser, micromanipulators, focused ion beams) and in-depth technical knowledge [31, 33]. The analysis of the microprocessor structure is done first for starting the



Invasive attacks. That is why it is required that the first line of protection measures is deployed at this stage so that it can complicate the analysis for an attacker. The steps that can be followed in protection measures are to make a small size of IC components which will make it hard to extract information [33, 34]; make dummy structures on the chip which have no meaning to the functions of the chip but used for misguiding [33], the layout of the chip's blocks can be random [31], make the chip cover with a metal layer which will hide the layout of the chip [31]. Even though the hiding of the structure with cheap material will make the analysis difficult for an attacker, still it should not be considered as a suitable protection against the physical attacks. The mechanisms discussed below provide stronger protection specifically against intrusive attacks:

- **Buried and scrambled chip buses:** In this mechanism the buses are buried inside the chip to prevent direct contact. Other than this the buses are jumbled to complicate the understanding what these buses are responsible for [33, 34].
- **Current-carrying metal layer on top of the chip:** In this mechanism the whole chip is shielded with a metal layer on the top which can be current-carrying. In addition to the protection from internal structure analysis the metal layer also protects against the attacks that use electrical measurements from the chip's surface [33]. As a result, if the shield of metal is removed then the chip will not be operational.
- **Memory encryption:** In this case the volatile and non-volatile memory and some microprocessor registers are encrypted with some specific keys [33, 34]. But in this case if the data is read from memory still it will be required to be decrypted.
- **Buses encryption:** In this the data in buses during transfer is encrypted [34]
- **Anomaly sensors:** In this case the sensors are used to detect the abnormal environmental conditions e.g. Voltage monitoring, temperature monitoring and external clock monitoring are examples of available sensing techniques [31, 33, 34]. It is important to notice that the clocking is provided by an external source, and as a result the processing speed of a smart card is also controlled by this external clocking source. It will be quite useful for an attacker to lower the frequency significantly as it will make the measurements easier [33]. Therefore to defend against this attack it is required to implement the clocking sensor.

One of the examples of non-invasive attacks is side-channel attacks. For side-channel attack a monitoring of device during its normal operation is carried out in order to obtain some information. Some of the factors that can reveal information are timing information, power consumption levels, electromagnetic leaks that are correlated with execution of some commands/computations on secret data. In timing attack execution time of particular operations is measured to get some required information about the secret data. In power analysis the information about which operation and with what parameters is being executed is leaked by monitoring the power consumption levels. One of the powerful technique is Differential power analysis (DPA) as the statistical analysis is applied to the power measurements done while repeatedly processing the known data and then the unknown data to observe the difference in the power consumption for different input data [31, 22]. Through this way the unknown data can be exposed either partially or fully. The principle in electromagnetic analysis is the same as discussed but it deals with the electromagnetic radiations from a chip.

To protect against the timing attacks we need to use the constant execution time algorithms which consume the same execution time for different input values. But the microprocessors that have constant power consumption because of hardware tweaks are very expensive and are not

realistic [37]. Other than that we can use Random delays in an algorithm and other more sophisticated masking techniques to decrease/destroy correlation between measured parameters and secret data [31, 37]. Though, according to [37] most of these protection measures are fragile to differential fault analysis.

Injection of faults to disturb the operation of a microprocessor is the basis for the Differential fault analysis. Some of the factors are abnormal voltage, temperate, clocking and electromagnetic influence that can lead to skipped commands, misinterpreted commands and data read with errors [31]. The countermeasures [31] for such attack can be using checksums, having a variable redundancy when a variable has a copy that is modified along with the original and then they are compared, and execution of the same commands several times and comparison of results etc.

As Smart cards are complex devices so the security is achieved by the blend of logical and physical protection measures. A weakness in one of these two layers can partially or completely compromise the smart card. Thus it is required to careful design the software and for smart card security.

It is necessary to test the claimed security functionality of smart cards against some international standard against the claimed security by the company. It is necessary to have an international security evaluation standard that assures the security functionality of a product against some requirements. In that case the evaluation is done by independent certified laboratories. The depth and the scope of the evaluation concludes the evaluation assurance level (EAL) issued by the certifying organization. Thus EAL is the level of confidence that the security functionality of a product is the same as required [38]. The EAL is divided into seven assurance levels ranging from EAL1: “functionally tested” to EAL7: “formally verified design and tested”. Protection profiles explain a set of security features that should be provided for a specific product type.

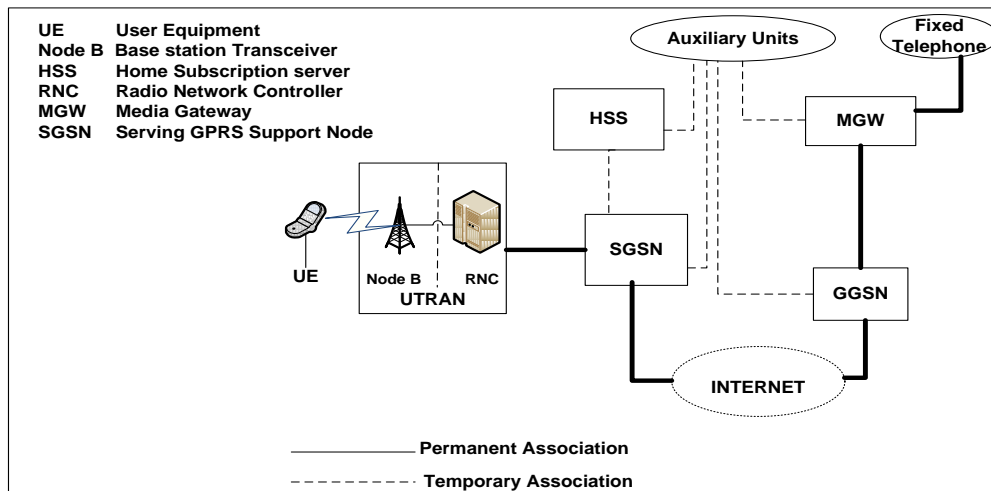
In a smart card there are three distinct layers/modules that are integrated circuit, operating system, applications. These components can be analyzed separately or as a whole system for security [34]. The modular approach (evaluating each module separately) is more suitable since the change in one module means only checking for that particular module and not for the whole platform [31, 34]. Moreover, after the modular security evaluation the composition evaluation (the whole system) can be done [31].

Security is the foundation of smart card development. The blend of logical and physical security mechanisms that form a unified system can be evaluated according to international standards to ensure a high level of security. Thus the ability to store information and execute cryptographic protocols in a secure manner provides a great success for smart cards to be used in sensitive security areas.

## **2.4 Identity management in UMTS system**

Universal Mobile Telecommunications System (UMTS) is one of the 3<sup>rd</sup> generation technologies, and now the 4<sup>th</sup> Generation, where “All IP” concept is introduced, is being developed.

UMTS phones are designed in such a way that they easily roam in UMTS networks. Besides, they have a capability to switch to GSM where UMTS coverage is not available, thus providing flexibility.



**Figure 4: ALL IP UMTS**

The General UMTS architecture consists of the following elements:

- User Equipment (UE) in the above diagram is the mobile terminal.
- Node B is a base station transceiver.
- Radio Network Controller's (RNC) are interconnected calls can be called transparently between them. RNC and node B together are called UMTS terrestrial radio access network (UTRAN).
- The Home subscription server (HSS) contains data including identities, IP/telephone numbers, service provisioning information and security support.
- The Auxiliary units support functionalities related to multimedia handling e.g. session control, multicasting, conference facilities.
- The location information is in Gateway GPRS support node (GGSN).
- Media Gateway (MGW) is a place where there is a format conversion between telephone network and an internet.

### Security of Identity in UMTS architecture

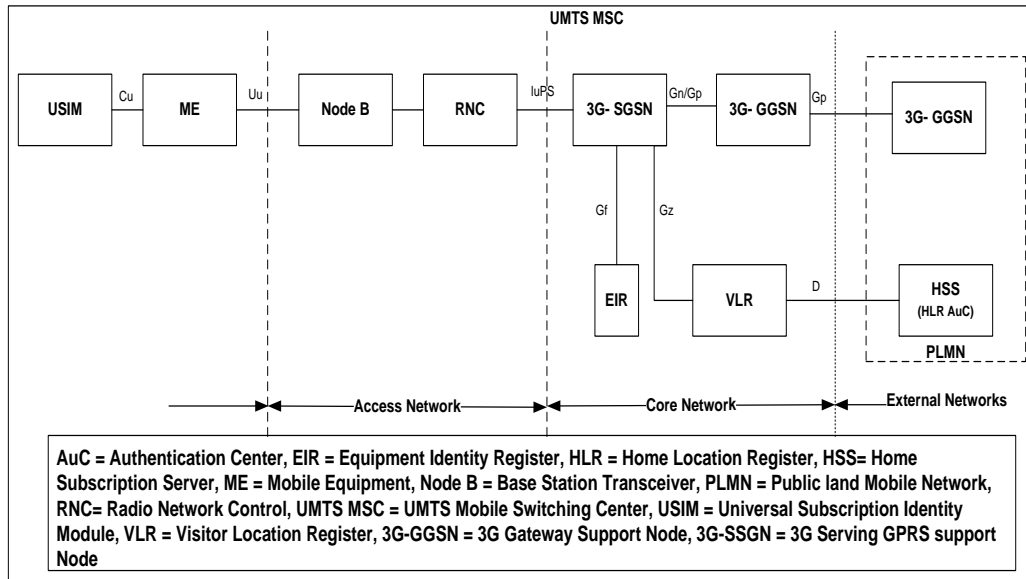
The architecture for the security of identity in UMTS is inherited from the previous architectures (GSM), added with some extra functionality [39]. It is one of the important part of current E-commerce and other applications to have secure and protected identity and anonymity of user. The security features of identity that are specified with the security of UMTS architecture discussed in [40] are as follows:

- **User Identity Confidentiality:** The property in which International mobile equipment identity is not be identified during communication.
- **User Location Confidentiality:** The property in which the location of the user cannot be found be found during the active link on radio interface.
- **User Untraceability:** To be unable to find out that the user is currently using which services.

### Identities in UMTS Architecture

In the UMTS architecture there are a lot of places where identity can be leaked as can be seen in

the diagram given below.



**Figure 5:** UMTS architecture [40]

To avoid the leakage of identity, different identities are used to protect user privacy. They are as following:

- Mobile Subscriber Integrated Services Digital Network (MSISDN): This identity represents the user phone or mobile number.
- International Mobile Equipment Identity (IMEI): This is the Mobile Equipment (ME) serial number. This can be helpful in fraud prevention.
- International Mobile Station Equipment Identity and Software Number (IMEISV): It is a like IMEI which represents both hardware and software.
- International Mobile Station Identity (IMSI): It is the permanent identity stored in the Universal Subscriber Identity Module (USIM) secure component i.e. smart card
- [Packet] TMSI is a temporary identity assigned to ME by the local network in which the user is registered.

The use of different identities is described in the table given below.

**Table 1:** Identities in UMTS components [40]

Parameter	Type	HLR	VLR	SGSN	GGSN
MSISDN	T	M	M	M	M
IMEI	T	-	-	C	-
IMSI	P	M	M	M	M
P-TMSI (Signature)	T	-	-	C	-
Legend: M = mandatory, C = conditional, T = temporary, P = permanent					

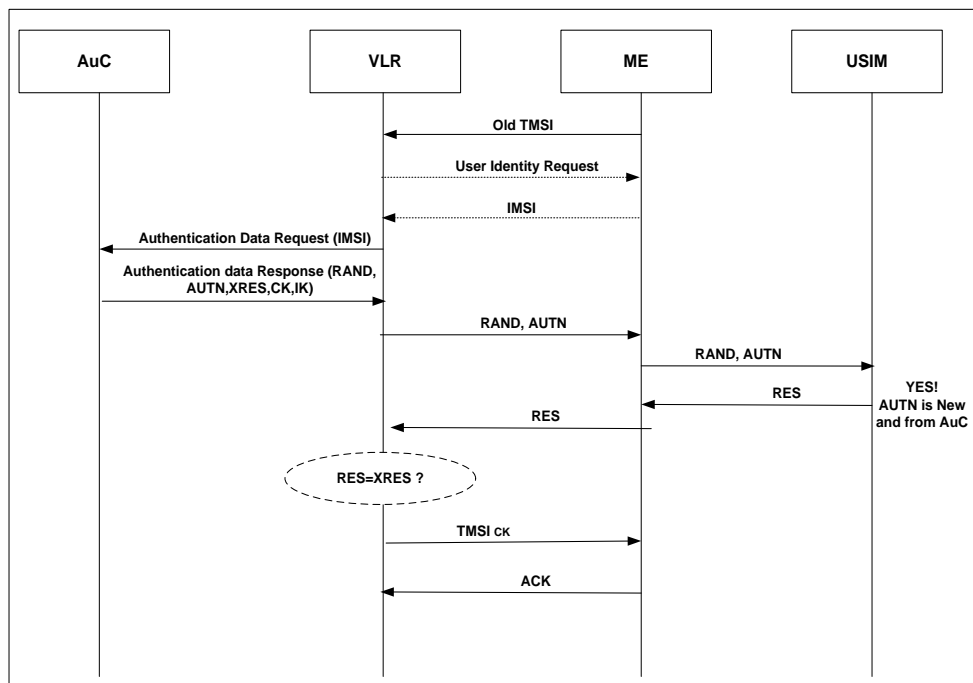
## Mutual Authentication in UMTS

The mutual authentication protocol in UMTS depends on the challenge-response pattern. The main aim of this is authentication is two ways:

- Authentication of Mobile Equipment (ME) at HLR or Base Station (BS) for billing and other purposes.
- Authentication of BS from ME to avoid from fraud BS.

The ME contains an USIM that is a tamper resistant smart card embedded with some cryptographic algorithm, a master key that is shared with the Authentication center (AuC) and a permanent identity IMSI. During the authentication procedure of ME, the AuC generates an integrity key (IK) for message authentication and a cipher key (CK) for message encryption.

The mutual authentication and key agreement are showed in the following diagram.



**Figure 6:** Mutual Authentication and Key Agreement [40]

The mutual authentication can be described by the following steps:

- When ME discover new VLR, it sends the old TMSI as acquired in the previous execution. If VLR does find the IMSI against the given TMSI then it asks for IMSI.
- When the VLR identifies the ME, it sends request for authentication data from Authentication center (AuC) of ME Home network.
- The AuC generates a challenge for ME from the master key with IMSI. The replay attacks are avoided by adding sequence no (SQN) and random no's (RAND). The AuC also computes an authentication token (AUTN), expected response (XRES), Integration key (IK) and Cipher key (CK) forming an authentication vector and send it to VLR

- When the VLR receives the authentication vector it challenges the ME with RAND and AUTN
- With master key embedded in the USIM and RAND, the USIM can authenticate the challenge through AUTN.
- Then the USIM computes the response (RES) to challenge the cipher key (CK), Integrity key (IK) and authentication key (AK). The RES is returned to VLR
- The VLR compare it with the expected response (XRES). If they match the keys CK and IK can be transferred to RNC, which can establish a secure communication channel with ME.

At times we send the TMSI clear over the connection which is against the security features for UMTS architecture (User location confidentiality and user Untraceability), as an eavesdropper is able to link the different connections established under the given TMSI.

There are some solutions proposed in [40], it discussed the problem and its solutions in detail. In [41] other attacks on UMTS are discussed.

## 2.5 Authentication and authorization in Linux

There are many ways of having secure authentication and authorization in Linux, and with time the mechanisms are improving.

### Passwd File

Normally for all Linux distributions user information is saved in the *passwd* file [44] which can be found at */etc/passwd*. This text file contains the following information

- User login
- Encrypted password
- Unique Numerical User ID (UID)
- Numerical group ID (GID)
- Optional Comment field (Such as name, phone etc)
- Home Directory
- Preferred shell.

The entry in the *passwd* file can be seen as

**<User login: Encrypted password: UID: GID: optional comments: Home Directory: Preferred shell>**

For example, we can see the entry as [44]:

*Pete:K3xcO1Qnx8LFN:1000:1000:Peter Hernberg,,1-800-FOOBAR:/home/pete:/bin/bash*

### Shadow Passwords

As *passwd* file is readable for all users and it contains all information including the encrypted passwords, so making it easy for accessing the password cracking. To get rid of this problem shadow password were developed. In this method the password field in *passwd* file has been replaced by “x” and the encrypted password is stored in the shadow file which is stored at */etc/shadow*. This file is only accessible by root user so making it a bit safer than *passwd* file.

The shadow file contains the following information:

- User Login
- Encrypted password
- Number of fields relating to password expiration.

The entry in passwd can be seen now after shadow password as [44]

```
Pete:x:1000:1000:Peter Hernberg,,1-800-FOOBAR:/home/pete:/bin/bash
```

And the entry in shadow file will be [44]

```
pete:/3GJllg1o4152:11009:0:99999:7:::
```

## Group Information

The group information in Linux is stored in a group file located at */etc/group*. The data in the group file are as follows

- Group name
- Password
- Numerical ID of group (GID)
- Comma separated list of group members

The entry in the group file can be seen as

```
pasta:x:103:spagetti,fettucini,linguine,vermicelli
```

“X” in the above example is a group password and it can also be shadowed. The shadowed group password is stored in */etc/gshadow*.

Some of the built in groups in Linux are showed in the table 2 given below.

**Table 2:** Linux built-in groups

<b>Group</b>	<b>Rights</b>
Root	This is a group which has full all the rights and is administrator. This group has ID 0 [45]
Wheel	The wheel group is a group which limits the number of people who are able to “su” to root, meaning that other user can have root privileges.[46]
Lpr	lpr is for the printer. Adding a user to this group allow that user to use the printer.
Shadow	for programs needing access to shadowed-passwords
Users	By default every new user is assigned this group.

## Encrypted Passwords

UNIX used crypt () function for encrypting the passwords. But with time the encrypted passwords were cracked, so new mechanism were introduced to make the cracking of password difficult. MD5 is one of the hashing algorithms that are added for encryption by most Linux distributions, but still this does not make it the full safe from cracking.

## Pluggable Authentication Model (PAM)

Nowadays Pluggable Authentication module (PAM) comes with many Linux distributions and is one of good mechanism for authentication of users. Before this scheme as discussed above, we do not have any method for authentication for more than one program. If a program needs user information for validation then there was no solution if we have different authentication schemes for different programs. PAM is the solution of it by enabling programs to transparently authenticate users.

The purpose of PAM project was to have separate development for privilege granting software's and secure authentication schemes. It doesn't deal where the password is saved like in passwd file or on a server in different place. When a program needs to authenticate a user it provides library which contains the proper functions for authentication.

PAM strength is its flexibility. We can configure PAM in many ways for a program that can be as follows:

- we can allow certain programs to authenticate users
- we can only allow certain users
- to warn when some program attempt to authenticate
- we cannot allow any user to have login privileges

In short PAM enables strong per-service authentication features, shadow passwords, strong hashing functions, and change mechanisms with changing system requirements. The flexibility is available at the small cost of increased complexity [47]. Thus, it provides a complete control on the authentication.

## Support of PAM by Linux distributions

The PAM is supported by nearly all distributions. Some of the names of the distribution are as following [44]

- Redhat since version 5.0
- Mandrake since 5.2
- Debian since version 2.1 (partial support in 2.1 -- complete support in 2.2)
- Caldera since version 1.3
- Turbolinux since version 3.6
- SuSE since version 6.2
- Ubuntu

## PAM Configuration Files

PAM configuration files are stored in the */etc/pam.d* directory or all the relevant configuration information can be stored in the */etc/pam.conf* file. If *pam.d* directory and *pam.conf* file are present then *pam.d* is given preference and *pam.conf* file is ignored.

In */etc/pam.d* directory each service has its own configuration file, which is named as the program or service name. E.g. the login application (*/bin/login*) is configured in */etc/pam.d/login* file. When the application programmer defines their applications, they should install its configuration file for installation of that service. Usually the operating system installers do this.



## Format of Configuration file

The command of a PAM configuration file consists of four possible arguments which are as following

- Module interface
- Control flag
- Module path
- Module arguments.

The command in configuration file can be seen as:

```
interface control_flag module_path [module_arguments]
```

In the above command structure, the module arguments field is optional. The following is the example line that has the first three fields:

```
auth required pam_unix.so
```

**Module interface:** It shows the type of authorization of a module. A PAM module may consider one or all four possible interfaces. The administrator can specify for each interface, in the configuration file for a service and for the module. The four possible interfaces discussed are as following [47]:

- **Account:** This interface checks if an account is authorized to use the system, e.g. to check if it exists, its expiration time, either it is allowed to access at a particular time or a particular service.
- **Auth:** This interface authenticates a user either by than checking a password or can be by another mechanism. **auth** modules are also allowed to set credentials such as group memberships or Kerberos tickets.
- **Password:** This interface is for checking and setting password authentication.
- **Session:** This interface configures and manages a user's session.

**Control flags:** In configuration file a control flag is specified for each interface that determines that what PAM will do next, depending upon the result of the check performed. There are four control flag types which are discussed as following [44, 47]:

- **Optional:** In case of Failure to authenticate using this module results in direct denial of authentication.
- **Required:** In this case Failure will result in denial of authentication, although PAM still will call the other modules programmed for this service before completely denying authentication.
- **Requisite:** If this module authenticates successfully, then PAM will authenticate it, even if the previous required module results were failed.
- **Sufficient:** The importance of this module success or fail is only considerable if it is the only module for this type of service.

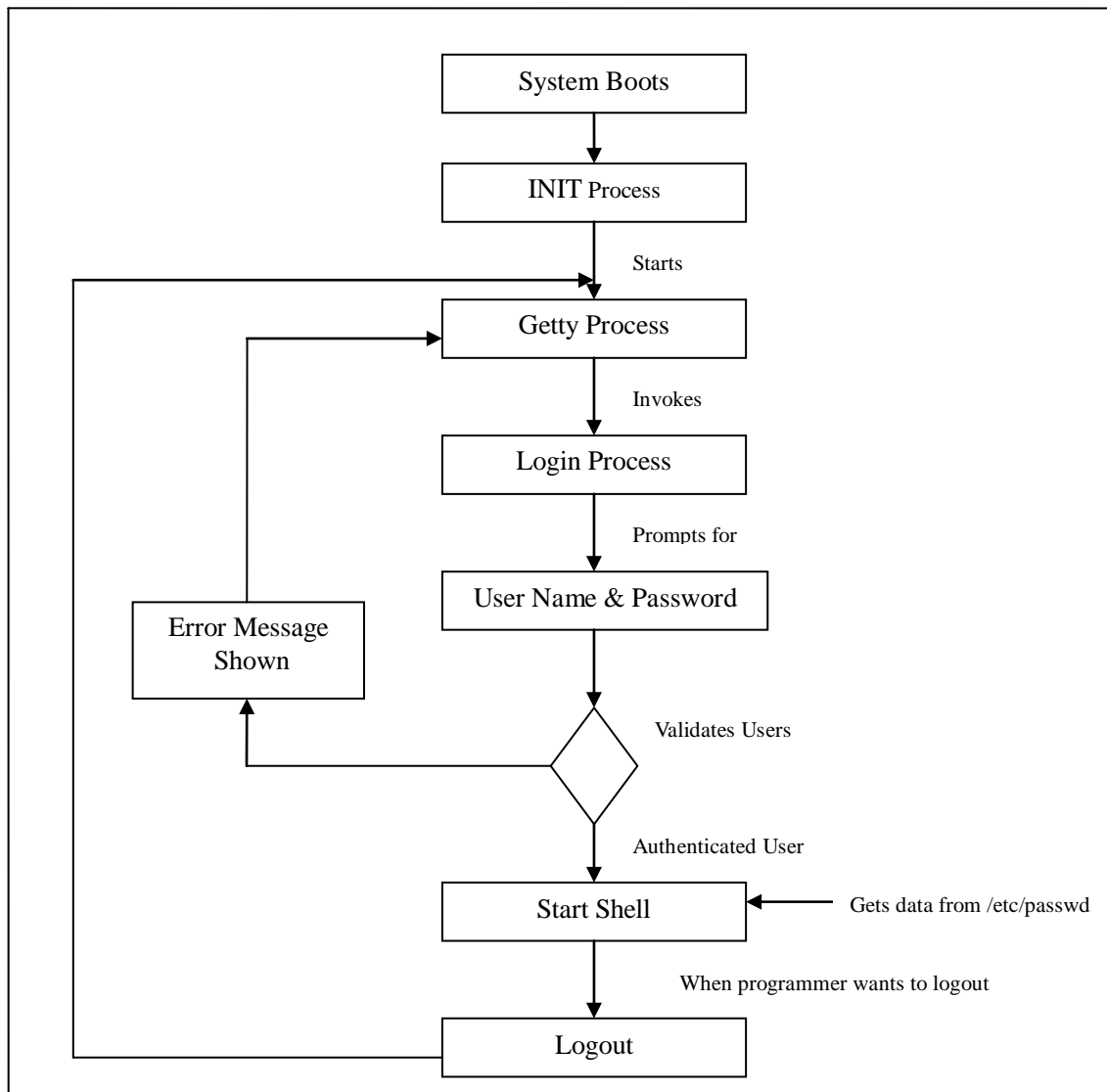
**Module path:** It tells PAM the location of the module, which is normally a full pathname that includes the module name and extension, e.g. */lib/security/pam\_unix.so*. In case where no path is specified, then the PAM by default searches the module in */lib/security*. The module path may also include the variable *\$ISA* as shipped from the vendor.

## Login and Authentication Process in Linux

When the system boots, there will be login prompt on screen. This login prompt is generated by a program, generally **getty** or **mingetty**, and it is regenerated by the **init** process whenever a user ends a session on the console. The process of login can be defined as following [48]:

- When the system boots, the **init** process starts the **getty** process.
- The **getty** process invokes the login process and the system asks for user name.
- The user enters the user name to login
- After this, the login process asks the user for a password, authenticates it.
- If the authentication is successful, the user's shell is started. The login program will get the data required from **/etc/passwd** file to decide which shell program to run. While on failure the program displays a message of error and then **init** will again start the **getty**.
- When the user logout, the shell program ends and we are again in the process.

The process is depicted in the chart below as well.



**Figure 7:** Login Process in Linux

## Access Control Lists (ACL) in Linux

Access Control Lists are an element of the Linux kernel and they are supported by Ext2, Ext3, ReiserFS, JFS and XFS. With the help of ACLs, complex problems can be solved without implementing complex permission models on the application level.

The use of ACLs can be in such situations where the traditional file permission concept is not sufficient. It allows assigning permissions to individual users or groups even if they are not from the owner or the owning group.

As discussed in [49] and [50], the traditional POSIX (Portable operating system interface) Access control lists permission concept which is similar to traditional file system, uses 3 classes for assigning permissions in the file system, the *owner*, the *owning group*, and *other* users. It uses 3 permission bits for setting each user class, giving permission which can be read (r), write (w), and execute (x).

In this method, the owner class permissions describe the access rights of the file owner; the group class permissions describe the access rights of the owning group, and the other class permissions describe the access rights of all users that are not in the two previous classes.

An ACL has a number of entries which describes the rights of each file system object with an ACL representation. e.g., “`rwX r-- ---`” means that for a regular file it has read, write and execute access for the owner class, read access for the group class, and no access for others. The entry types are as in the given table below.

**Table 3:** Types of ACL entries [50]

Entry Type	Text Form
Owner	<code>user::rwx</code>
Named user	<code>user:name:rwx</code>
Owning group	<code>group::rwx</code>
Named group	<code>group:name:rwx</code>
Mask	<code>mask::rwx</code>
Others	<code>other::rwx</code>

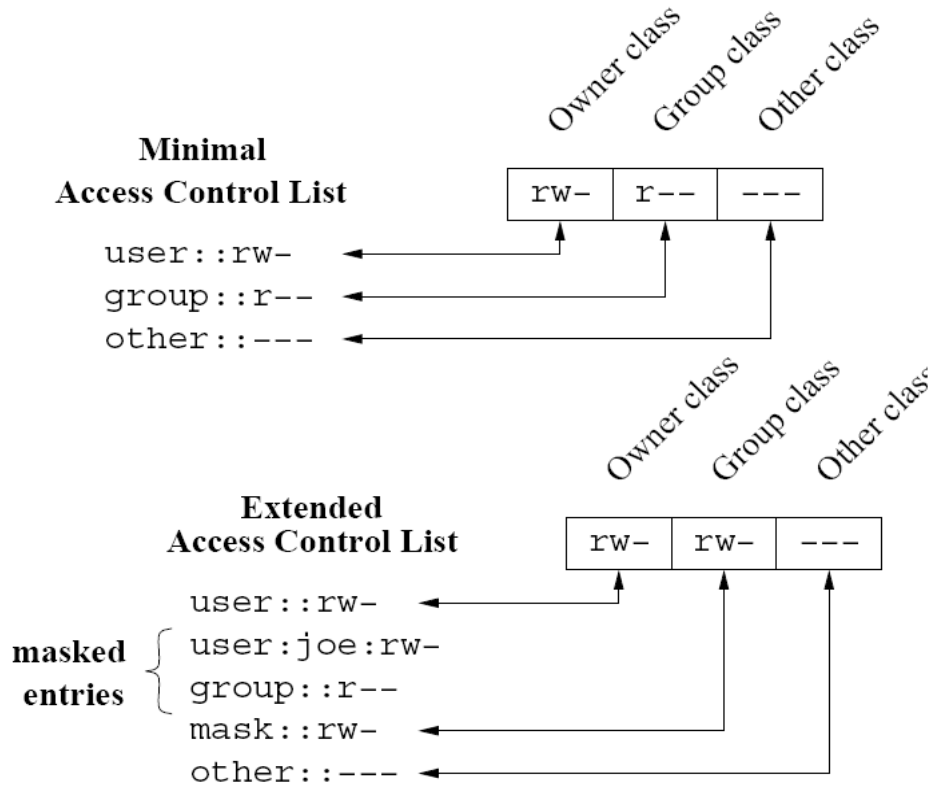
Each of these entries has the following syntax: *Type: qualifier: set of permission*

Qualifier defines to which user or group the entry applies. The qualifier is not defined for entries that require no qualification.

When ACLs are same as the file mode permission bits are then it is called **minimal ACLs** and they have three ACL entries. The ACLs having more than three entries are called **extended ACLs** and these ACLs also contain a mask entry and they may contain any number of named user and named group entries.

The named group and named user entries are allocated to group class, which already have the owning group entry. So now the group class may contain ACL entries with different permissions. The group class permissions are no longer only enough to represent all the detail of permissions of all ACL entries. Now the group class permissions represent an upper bound of the permissions.

In minimal ACLs, the group class permissions are exactly the same as the owning group permissions. But in extended ACLs case, the group class permissions are mapped to the mask entry permissions using mask. The mapping of the group class permissions can be seen in the figure given below.



**Figure 8:** Mapping between ACL entries & File mode Permission Bits [50]

As in the group class permissions denote the upper bound of the permissions granted by any entry. So With extended ACLs, this is implemented by masking permissions which can be seen in the table given below.

**Table 4:** Masking of permissions [50]

Entry Type	Text form	Permissions
Named User	user : joe : r-x	r-x
Mask	mask : : rw-	rw-
Effective permissions		r--

**Access ACL:** It determines the user and group access permissions for all kinds of file system objects.

**Default ACL:** These ACLs can only be assigned to directories. They conclude the permissions a file system object inherits from its parent directory when it is created.

## Support of Smart cards in Linux

The Linux-PAM login module also supports X.509 certificate based user login. For every user the unique digital certificates for that user can be stored on smart card. To access the certificate and its dedicated private key in Linux we can use an appropriate *PKCS #11* module.

For the approval of the ownership of a certificate (smart card), to login as a specific user, *pam-pkcs11* uses several modules called *mappers* that perform certificate-to-login mapping.

We can configure PAM PKCS#11 module to authenticate the user with smart card, normal username/password method or both at the same time.

## Writing PAM Module

Some time we need to write our own PAM module for carrying out some task that we want to be additionally added with the existing functionality. PAM modules are mostly written in C language. For development of PAM modules we need to install *libpam0g-dev* package and the following header files should be added in the file while writing a PAM module.

```
#include <security/pam_appl.h>
#include <security/pam_misc.h>
```

A sample code has been given in Appendix A with compilation steps. The steps involved in the writing a PAM module can be seen in the figure given below as discussed in [52]:

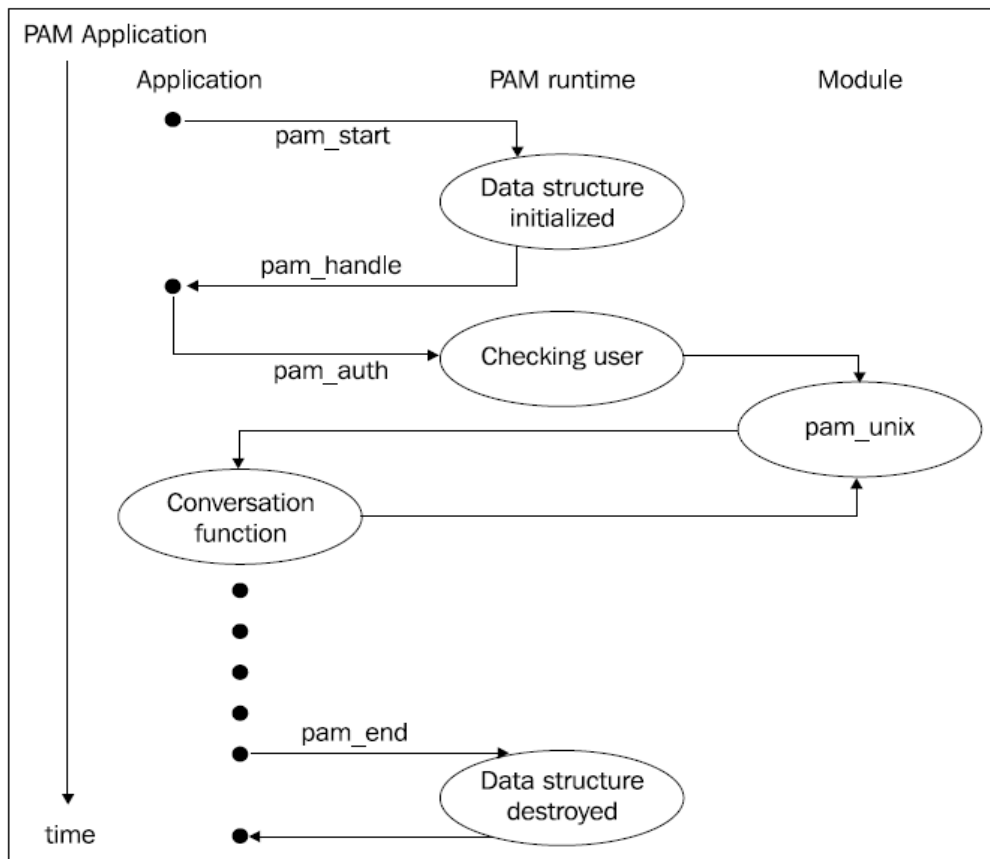


Figure 9: PAM Application Structure

- The first step in writing a PAM module is to initialize a PAM session. With initialization of PAM session we initialize all the required data structures and we can access the data with a PAM handler which is returned after initialization of PAM session. The following function starts PAM session and initialize the data structure

```
retval = pam_start("servicename", user, &conv, &pamh);
```

In above function the first parameter is the service name e.g. Login. The second parameter is the user name. Third parameter is the pointer to the conversation function, and the last parameter is PAM handler with which we can access the data structure later. The type of return values are discussed later

- We can authenticate the user by using the following statement

```
retval = pam_authenticate(pam, 0)
```

The first parameter is the PAM handler which is created in the first step. The return value types are discussed later.

- We can manipulate with the data structure now at this step. The data of the user can be accessed with the *passwd* structure in the *pwd.h* header file. The *passwd* header file contains the following data.

```
struct passwd {
    char *pw_name; /* user name */
    char *pw_passwd; /* user password */
    uid_t pw_uid; /* user id */
    gid_t pw_gid; /* group id */
    char *pw_gecos; /* real name */
    char *pw_dir; /* home directory */
    char *pw_shell; /* shell program */
};
```

- At the end it is required to end the PAM session. The session is ended with the following statement.

```
pam_end(pamh, retval);
```

**Conversation Function:** conversation functions are required to handle the call backs. The conversation function is used by modules to interact for the authentication like login name or password during authentication. The Linux has its own default conversation function. We can write our own conversation if required.

**Return Codes:** There are specific return values that are returned by the function as can be seen above. The *retval* shows the return value in the above statements.

**Table 5: Return Codes**

Return code	Management group	Meaning
PAM_SUCCESS	All	Everything went well
PAM_USER_UNKNOWN	Auth, Account, Password	The authentication token (user name) is not known
PAM_SESSION_ERR	Session	Any error related to opening or closing sessions
PAM_AUTH_ERR	Auth, Account	Authentication failed
PAM_ACCT_EXPIRED	Account	Account has expired

## 2.6 Authentication and authorization in Windows

### Security principals and access to objects

The Security in Windows system depends on the subject-object-actions relationship. In case when the subject requests for some action to perform on an object, it is checked by the operating system whether the access should be granted on the basis of security permissions associated with that object. As defined in [53] a subject can be a user, a computer or a service. And according to [54] an object can be a file, a job, a device, a process, a thread, shared memory sections or volumes etc.

The identity of the principal must be known by the operating system for making decision whether to grant some access to an object. So, to access the resources a user must be authenticated to the system during Windows logon process. Windows operating system uses unique security identifiers (SIDs) for identification of security principals. The SID's are assigned during the logon process.

The structure of the SID is as following:

$$\underbrace{S - 1 - 5 - 21 - 2443930396 - 124871960}_{\text{Domain identifier}} - \underbrace{1960245352 - 1000}_{\text{RID}}$$

S – SID designator. It consists of a character "S";

1 – It is the revision number of the SID specification;

5 – It is authority identifier. The value 5 is for SECURITY\_NT\_AUTHORITY. Other values are: SECURITY\_WORLD\_SID\_AUTHORITY, SECURITY\_LOCAL\_SID\_AUTHORITY, etc;

Domain identifier – it is used to identify a domain or computer that issued SID.

RID – It is Relative identifier. They are used to guarantee uniqueness of SIDs. The RIDs for the non-default users and groups starts from 1000 and increase by 1 with every new principal.

The user and group accounts are divided in to two types in the Windows systems which are user-defined and automatically created by the system. The Automatically created groups are further classified in built-in and system groups. System groups have automatic and dynamic membership which depends on the principal's activity type. The Built-in groups are not much different from the user-defined groups and the reason for using is to support the default Windows security model [55].

Some of the well-known built-in accounts in Windows system are as following.

**Table 6:** Built-in accounts in Windows

SID	user/group	description
S-1-5-domain-500	Administrator	Built-in system administrator account
S-1-5-domain-501	Guest	Built-in account that can be used to give limited access to those who do not have personal account
S-1-5-domain-502	KRBTGT	Service account used by Kerberos KDC
S-1-5-domain-512	Domain Admins	Members of this group are authorized to administer a domain.
S-1-5-domain-513	Domain Users	Includes all user accounts in a domain
S-1-5-domain-515	Domain Computers	This group includes client stations and servers in a domain
S-1-5-32-544	Administrators	Initially contains Administrator account as the only member. After the system joins a domain, Domain Admins group is added to this group.

**Table 7:** Well-known system groups in Windows

S-1-1-0	Everyone	This group includes completely all users
S-1-2-0	Local	This group includes users who log on to computer locally via connected terminal
S-1-5-2	Network	Contains users who log on via a network
S-1-5-4	Interactive	Includes users that log on interactively. Terminal server users are in this group, but they are not included in Local group [52].
S-1-5-5-X-Y	Logon Session	This SID is used to identify a logon session, not a principal. Each logon session of a user is assigned a unique ID (X and Y values are changed)
S-1-5-11	Authenticated users	Contains authenticated users

An access token is created by the Windows after a user is successfully authenticated. The Access token consists of the SID of the user, SIDs of all groups in which the user is a member that includes built-in and system groups, the SID which identify the logon session, the assigned privileges of the user and user's groups etc. If there is some change in assigned privileges of the user or change in one of the groups in which the user is a member of, or the user membership in some group changes, after that the user must have to relogin for the changes to take affect [55]. A new access token which contains updated information is created after the user relogins. Every process or thread executed by the user contains a copy of the user's access token. In case if a process or a thread requires access to some secured object then they present an access token to the system for getting access.

There is a security descriptor associated with every secured object in the system. The security descriptor defines who can access an object and what kind of access is allowed. The security descriptor consists of the SID of the owner of the object, SID of the primary group of the owner (as discussed in [53] the primary groups in Windows access control model are only used for POSIX compliance), a system access control list (SACL) and a discretionary access control list (DACL).

The DACL defines the users and groups and if they are allowed to access an object. In case when an object does not have associated DACL then it can be accessed by everyone [56]. The DACL contains a number of access control entries (ACEs). Each ACE have the SID of the subject, the flag that shows if access is either allowed or denied, and an access mask which describes the access rights for subject. In case if the DACL of object contains no ACEs, then no one can access the object [57]. The following are the access rights defined in Windows [58]:

- **Generic access rights:** It includes the basic rights like READ, WRITE, EXECUTE and GENERIC\_ALL that is combination of all three
- **Standard access rights:** it includes the rights like DELETE: the right to delete the object, READ\_CONTROL: right to read object's security descriptor, WRITE\_DAC: right to modify DACL, WRITE\_OWNER: to change the owner of the object and SYNCHRONIZE right.
- **SACL access right:** It is for getting or setting SACL in the object's security descriptor
- **Object-specific access rights:** One of the example can be the right to create files in a directory is FILE\_ADD\_FILE for this kind of rights.

The SACL controls in case of attempts to access an object are logged. The ACEs in the SACL indicate which access attempt types and subjects should be logged. Failures and successful



attempts both are logged. The majority of objects in the Windows system do not have SACL [53].

In the case of discretionary access control model the owner of an object declares the access permissions for other users and groups. The owner of an object and the system administrator has the full control over the object in Windows. If the object has empty DACL no one can access the object including the owner, but still the owner can change access permissions [53]. Though, if the DACL of the object has ACE with the SID S-1-3-4 “OWNER\_RIGHTS”, in that case the implicit READ\_CONTROL and WRITE\_DAC rights of the owner are ignored. This technique was introduced to prevent the users from modifying permissions for their own files [53].

When a secured object requested to be accessed by any process or a thread, the system checks the provided access token against the DACL of the object taking into account the requested access type. ACEs in the DACL are analyzed sequentially for SIDs that match those in access token until access is granted or denied or reach the DACL end. Thus, access is granted when one or more access permitted ACEs allow all requested access rights for any subset of SIDs in the security token of the principal [59]. The Access is denied in that case when the deny ACE, for one of the SIDs in the access token which denies any of the requested access rights is encountered in the DACL [59]. The access is denied in case if the end of the DACL was reached and there is some requested access right that was not allowed by ACEs.

For every user in Windows system there are rights and privileges. As defined in [60] a privilege is a right to perform system-related management operations which can be changing time, rebooting the system, loading drivers etc. Hence the privileges differ from the access rights that the subjects have for the securable objects. The Privileges reside in the assigned access token while access rights are described in the security descriptor of the securable objects. It is required to differentiate between user rights and access rights for the securable objects. The Logon rights are same as privileges except that they are used to allow a user for log on to the system and privileges describes what a user can do after logon [53, 55]. Some of the rights for logon are discussed in the following table.

**Table 8:** Logon rights in Windows

<b>Logon right</b>	<b>Description</b>
Access this computer from a network	Determines which users/groups can logon via a network. By default this right is granted to everyone.
Allow log on locally	defines who can interactively logon to the system via connected terminal
Allow log on through Terminal Services	defines who can logon to a remote computer via Remote Desktop Protocol [54]
Deny access to this computer from network	defines for whom network logon is denied
Deny log on locally	defines for whom local logon is denied

### **Role-based access control**

The alternative to discretionary access control system is Role-based access control (RBAC) which is used in Windows. The resources are accessed on the bases of role. Thus, privileges are assigned to roles. The role defines which operations and job functions a user can perform. When a user has the assigned role he obtains the privileges of that role.

The support for RBAC in Windows is provided by Windows Authorization Manager (AzMan). The Administrative tools allow defining the role-based authorization policies against which the access control decision will be made.

## **Authentication and logon process**

The two types of authentication/logon models in Windows are interactive and non-interactive. In case of interactive authentication the user is prompted for credentials. On the other hand non-interactive authentication uses the credentials that are previously entered by a user during the interactive authentication. Thus it can be concluded that the interactive authentication must always precede the non-interactive authentication. Non-interactive authentication is done when the user requests connection to other stations/servers in the domain.

The interactive logon can be either local or a domain logon. With local interactive logon the user can only access the local system resources while in case of the interactive domain logon a user can access the resources of the whole domain.

### **Logon process in Windows XP**

The components include in the Windows XP interactive logon architecture are Winlogon process, Local Security Authority (LSA), Graphical Identification and Authentication (GINA) dll and authentication packages (NTLM and Kerberos). The process that is responsible for managing logon procedure is Winlogon process. It makes sure that no other illegitimate processes can interrupt the logon information provided by the user [54]. Winlogon depends on the GINA for obtaining the user logon information. After obtaining the credentials the GINA calls LSA to authenticate the user by using one of the authentication packages. The result of authentication is forwarded to GINA which returns it to the Winlogon process. If the authentication is successful then Winlogon process starts the user's shell.

The logon process can be started by a user with pressing CTRL+ALT+DEL which is the secure attention sequence (SAS). During the boot process Winlogon registers this sequence and other processes cannot interrupt this sequence other than Winlogon [54]. After detecting the SAS Winlogon calls GINA for obtaining the user's credentials. An interface is provided by GINA to get the credential from the user. To modify the standard interactive logon procedure, the default GINA dll (MSGina.dll) can be replaced by a custom GINA. E.g. a custom GINA can communicate with an external device for getting the user's credentials. If the default authentication packages are unable to analyze the credential information obtained through the custom GINA, then a custom authentication package should be used. LSA supports custom authentication packages. The ability to use a custom GINA and authentication packages gives us the advantage of implementing any authentication scheme virtually.

After the user's credentials are obtained the GINA calls the LsaLogonUser function to authenticate a user by using one of the authentication packages. LSA uses particular authentication package to authenticate the user. In case of local logon the local LSA authenticates the user, but in case of the domain logon, the LSA on the domain controller authenticates the users [61]. There are two packages provided by Microsoft authentication. One is MSV1\_0 authentication package which is used for the local logon and the second is the Kerberos authentication package for the domain logon. The MSV1\_0 compares the user name and hashed password with the stored user data in the Security Account Manager (SAM) database [54]. In case of the cached domain logon, MSV1\_0 can be used and in that case the cached credentials are stored in LSA database in the encrypted form. The result of authentication is returned by MSV1\_0 to LSA which is then forwarded to GINA, and if the authentication succeeds then a logon session is created. The Kerberos authentication package operates principally in the same manner but the only difference is that authentication exchange is done through the network and the authentication decision is made on the domain controller.

If the authentication succeeds then the LSA checks the local policy database for the logon rights of the user. The logon session is terminated and the failure notification is sent to Winlogon in case the user does not have appropriate logon rights [54]. LSA creates access token with appropriate account SID, group SIDs, session SID, and a set of privileges retrieved from the LSA policy database in

case when the user has appropriate logon rights. LSA passes the authentication result to GINA and then forwarded to Winlogon by GINA. Winlogon additionally receives the user's access token if the authentication succeeds and the Winlogon launches user's shell (default is Explorer.exe) and provides it with user's access token [55] so that the shell can perform operations on behalf of the user. Other processes are created from the shell and each process inherits the user's access token.

### **Logon process in Windows Server 2008, Vista, and Windows 7**

Credential Provider model is used for the logon architecture in Windows Server 2008, Vista and Windows 7 and redesigning of the authentication model is done so that GINA is not used [62]. Furthermore, in previous Windows versions a console session along being interactive logon session also hosted system processes and services [63]. In the new architecture session 0 (Console session) became the non-interactive authentication process, so the users log on to separate sessions starting from session 1. This makes the services in session 0 isolated from the user applications and services that run with higher privileges are protected from attacks and malicious application code [64].

The new interactive logon architecture consists of the components that are Winlogon process, the logon user interface (LogonUI) process, credential providers, LSA and authentication packages. The default credential providers support password and smart card authentication [62]. Thus, makes possible to install multiple custom credential providers. To support different identification mechanisms, custom credential providers can be developed. It can be decided by the user to use which credential provider or the selection of the credential provider can be event-driven.

When a user enters the SAS the logon process begins. After beginning of logon process the Winlogon starts the LogonUI for providing the user interface for logon. LogonUI requests the credential providers for obtaining the user's credentials. LSA is called to authenticate the user through one of the authentication packages after obtaining credentials. If authentication succeeds, LSA checks the local policy database for the logon rights of the user. If the rights are sufficient then it creates the access token and forwards the authentication result and the access token to Winlogon. If authentication is successful then the Winlogon launches the user's shell.

As discussed in [65], the logon customization is much easier and more secure in the new architecture with credential providers as compared the old model where the custom GINA dll had to be developed and GINA was responsible for the graphical logon interface. In the new architecture the graphical logon interface responsibility is taken by LogonUI. In new technology the credential provider informs the LogonUI where graphical control elements like checkboxes or edit boxes are required to obtain user's credentials. If LogonUI doesn't work for some reason, then Winlogon will simply restart it [62].

### **Standard smart card logon**

The advantage of smart cards is that it provides a two-factor authentication that is based on the possession of a smart card and the knowledge of a PIN. The smart card stores a private key and a corresponding X.509 certificate with a public key securely and the private key never leaves the card. Moreover, it is stored only on the smart card [66]. For authentication the smart card performs the cryptographic operations using the private key for proving to the authentication server that the principal's smart card contains the key. Before performing the cryptographic operations, a user must first authenticate to the card by presenting the PIN. The user is only prompted for the PIN. The workstation communicates with the smart cards via smart card reader.

In Windows systems smart cards can be used to log on with domain accounts and not with the local accounts. That's why the standalone smart card logon is not natively supported in Windows systems still there are some commercial products that offers this functionality. The domain smart card logon supports an offline logon capability which means that even in case of a network service disruption or a failure it is still possible to logon to a workstation that belongs to that domain.

The process of Smart card domain logon session for Windows XP [66] is done as following:

- First the Smart card is inserted into the card reader. This will automatically start the logon process.
- GINA is called by Winlogon for obtaining user's credentials. A logon screen is presented by GINA to the user and the user is prompted only for a PIN.
- GINA forwards the received PIN to the LSA;
- The PIN is used by the LSA for accessing the smart card.
- Kerberos Authentication Package (Kerberos SSP) is called by LSA. Kerberos SSP creates a Kerberos Authentication Service Request and sends it to the KDC. The request contains principal's certificate and a cryptographic signature generated with the corresponding private key for the Kerberos pre-authentication [67].
- The KDC validates the certificate which includes verification of the certification path, checks revocation status, etc. and checks the digital signature. After these checks the KDC retrieves the user account information from Active Directory. This information is required to construct a TGT which is created. The TGT Authorization data fields contains the principal's SID, the SIDs for domain groups to which the user belongs and (in a multi-domain environment) the SIDs for any universal groups in which the user is a member. For encryption of symmetric encryption session key, the public key from the certificate is used. The KDC digitally signed the response with other things that are TGT, the KDC certificate, and the encrypted session key. The client will be able to decrypt the session key and use it for subsequent interactions with KDC in case if the client possesses the private key that corresponds to the public key in the certificate.
- When the client receives the response, it validates the KDC certificate and checks the digital signature. With using the private key, the client can decrypt the session key for communication with KDC. For log on to the computer it is required to obtain the Ticket Granting Service (TGS) to the local computer from KDC. The remaining part of the authentication procedure is identical to the standard logon session.

Windows caches credentials after successfully authentication which gives the capability of performing authentication local or offline log on to the computer with the domain account even if the domain controller or the network connection is failed. Though, during the local smart card logon with cached credentials a Certificate Revocation List (CRL) check is not performed [68].

### **Windows Vista Smart Card Infrastructure and logon procedure**

Windows Vista supports the following not like the previous version of Windows [69]:

- Smart cards that contain several certificates only for logon purpose other than the certificates for other purposes. The smart card memory space tells that what number of the certificates can be stored.
- Another option is to change the PIN and unblock a smart card without requiring logging on first with a standard user name and password.

The Windows Vista supports a password credential provider and a smart card credential provider by default. For enabling the custom authentication mechanisms, a custom credential provider should be used.

The Logon steps in Windows Vista are discussed as following [69]:

- The logon does not start with insertion of card only; it starts after the SAS is pressed. In that case the WinLogon requests the logonUI for obtaining the credential information.
- A list of smart card readers and a list of inserted smart cards exist in the smart card credential provider. It checks for every card whether the logon certificate exists on the

card. The Found logon certificates are retrieved from the smart card and copied into a temporary secure cache. After this the smart card credential provider provides the logon certificates to the LogonUI.

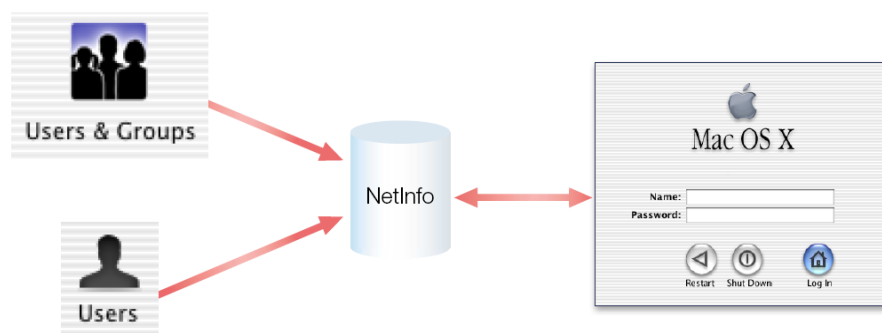
- A logon user interface is displayed by the LogonUI with found certificate logon tiles to a user. The user has the option to select one of the tiles and a PIN input box is displayed for getting the PIN from the user. The PIN entered is encrypted by the smart card credential provider.
- The smart card credential provider forwards the encrypted PIN, user name etc. to the LogonUI which calls the LsaLogonUser function and provides the received data to the LSA.
- Furthermore, the LSA calls Kerberos Authentication Package (Kerberos SSP) to create a Kerberos Authentication Service Request. The remaining part of the Kerberos authentication procedure is identical as in Windows XP.
- In case of successful authentication the certificates are read from the card (including the root certificates) and stored in the user's certificate store (MYSTORE).
- When the card is removed, the certificates are also removed from the temporary secure cache. Still the certificates will be present in the user's certificate store (MYSTORE).

As discussed in [70] Windows 7 and Windows Server 2008 R2 have some negligible enhancements to the smart card platform as compared to the Windows Vista that are mainly related to the Plug and Play service and smart card drivers.

## 2.7 Authentication and authorization in Mac OS X

One of the most obvious differences between UNIX implementations and Mac operating system is that the user information is stored in database called NetInfo, not in flat files like `/etc/password` as in UNIX systems.

As discussed in [71], in old versions of Mac the hashed password is used to be stored in the database and was accessible by every user, but now they use to store it in a shadow file to make it secure and only accessible by root user. The shadow files can be found at `/var/db/shadow/hash/` directory. The password is stored in each user name file which is not the same like the user name but generateduid NetInfo field.

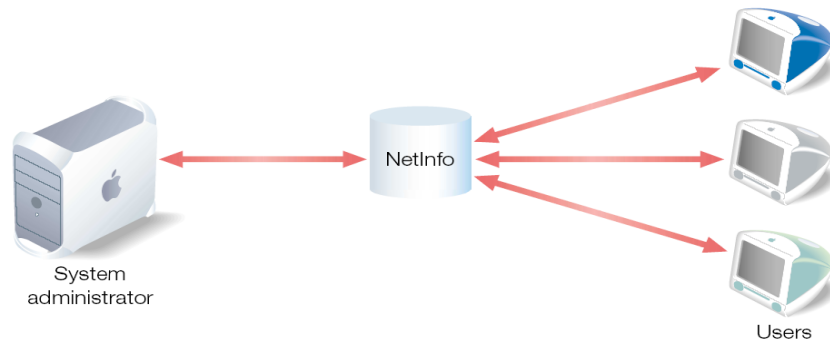


**Figure 10:** Interaction of NetInfo and Users [72]

In the figure above all the information of users and groups are stored in the NetInfo so when user logs in the information is authenticated with the information in the database.

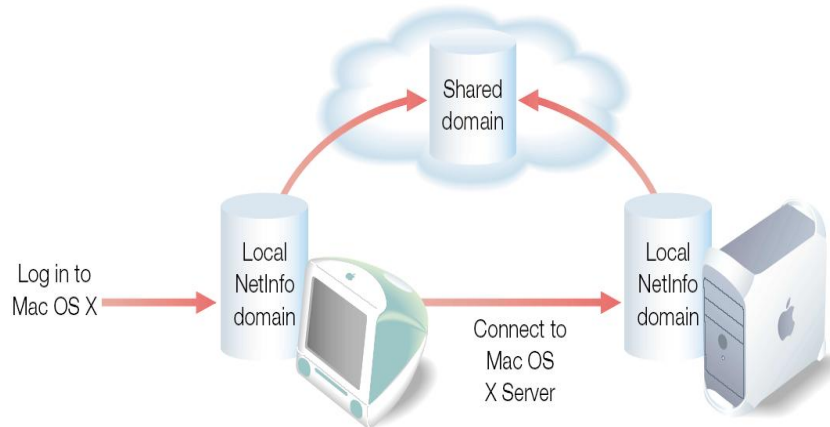
With NetInfo it is not required to know where the information is stored. When any process requires some information, it request for the required information which is provided by NetInfo. Second benefit of NetInfo is for administrators who are managing more than one computer. They can easily

manage the computers as can be seen in the figure given below.



**Figure 11:** Administrator Management with NetInfo [72]

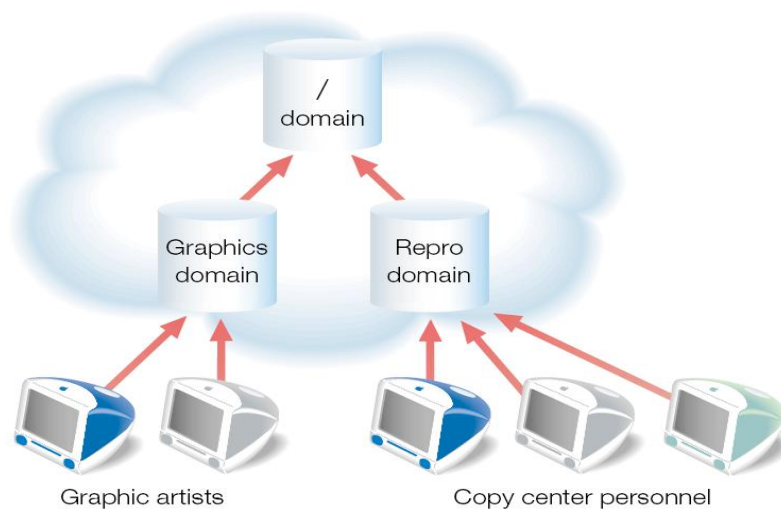
One of the real powers of NetInfo is storing data not only in local but can save it in shared domain which makes the work of administrators much easier.



**Figure 12:** Shared Domain in NetInfo [72]

In shared domain as shown in diagram shown above the user data is first search in Local NetInfo domain, if not found then it goes to the shared domain for getting the required data.

We can share the data in hierarchal form with NetInfo as well. The root node is denoted as “/”. The following diagram can explain scenario.



**Figure 13:** NetInfo Hierarchies [72]

In the above diagram there are two domains one is Graphic artists and the second is Copy center personnel. The “/” is the shared between both the domains. The child and parent are attached with the method called Binding. There are three kinds of binding discussed in [72] are as following:

- Broadcast Binding
- Static Binding
- DHCP Binding

Each record of user in the NetInfo consists of the following items:

- User ID (UID)
- Short Name
- Real Name (Users full name)
- User’s password encrypted using one-way encryption algorithm.
- Home Directory style. (It is used by Server admin to distinguish among home directory styles like none, local, custom)
- Absolute path of user’s home directory
- Home location is present if home directory is on an Apple file server.
- Group ID (GID)

The administrator account is called root. Normally it is given UID 0 and can make the changes in the database. UID’s below 100 are reserved for system use. When we add the user with add user pane the UID’s starts from 500 and are assigned automatically. The GID’s are assigned automatically when we add a group from menu and starts from 100.

When a Mac OS X computer boots and domain binding occurs, then the following steps occur:

- NetInfo daemon called nibindd is started.
- The nibindd daemon starts another daemon called netinfod for each domain on the computer. The netinfod process is sometimes referred to as a NetInfo server
- The nibindd listens for requests from netinfod processes asking for parents, checking for the appropriate netinfod process and initiating binding. The nibindd and netinfod run in the background.
- Another process that is relevant to NetInfo is called lookupd. It is used to interact with NetInfo when legacy UNIX software requests administrative information which is now stored in NetInfo. The lookupd process makes it likely for software that uses Posix or BSD calls to retrieve administrative information from NetInfo.
- Then the user data is used for authentication. If validates than allowed otherwise rejected
- The UID in the record describes the operations that user can access.
- The GID also has effect on the access of a user privileges as group has assigned rights. It check at local system if not found that proceed towards the root domain.

## **Authorization**

In Mac OS X we have a security server i.e. a core services daemon which deals with authorization and authentication. It determines whether everyone is allowed or some users have the access.

The Security process is same like an official doing visa processing which can be seen in the table 9.

**Table 9: Authorization in MAC OS**

<b>Immigration</b>	<b>Authorization</b>
The immigrant provides a passport and visa to the immigration official.	The application provides the authorization reference, authorization rights set, and authorization options to the Security Server.
The immigration official uses the visa number to access information about the immigrant.	The Security Server uses the authorization reference to access credentials.
The immigration official uses the picture in the passport to validate the identity of the immigrant.	The Security Server asks the user to provide a user name and password for authentication.
The immigration official uses the privileges requested in the visa to look up the laws in the policy book.	The Security Server uses the rights in the authorization rights set to look up the rules in the policy database.
The immigration official uses the credentials to determine if the immigrant complies with the laws and should be granted the privileges requested in the visa.	The Security Server uses the credentials and authorization options to determine if the user complies with the rules and should be granted the rights requested in the authorization rights set.
The immigration official informs the immigrant whether or not he grants the privileges requested in the visa.	The Security Server returns a result granting or denying the authorization rights.

There is a policy database which contains a set of rules that the security server uses to authorize the rights of the user. Each rule has a set of attributes which is discussed in the given table.

**Table 10: Rule Attributes and Description [73]**

<b>Rule attribute</b>	<b>Generic rule Value</b>	<b>Description</b>
key		The key is the name of a rule. A key uses the same naming conventions as a right. The Security Server uses a rule's key to match the rule with a right. Wildcard keys end with a '.'. The generic rule has an empty key value. Any rights that do not match a specific rule use the generic rule.
group	Admin	The user must authenticate as a member of this group. This attribute can be set to any one group.
shared	True	If this is set to true, then the Security Server marks the credentials used to gain this right as shared. The Security Server may use any shared credentials to authorize this right. For maximum security, set sharing to false so credentials stored by the Security Server for one application may not be used by another application.
timeout	300	The credential used by this rule expires in the specified number of seconds. For maximum security where the user must authenticate every time, set the timeout to 0. For minimum security, remove the timeout attribute so the user authenticates only once per session.



## Support of Smart cards in MAC OS X

We can use smart cards in MAC OS X for authentication. Smart cards can exchange information with a personal computer through a smart card reader. MAC OS X provides a Smart Card Services software development kit (SDK), which has code that can be used to implement a PC/SC-Supported application. (The PC/SC is a work group that defines standards for accessing data from smart card reader).

The files can be found in Mac OS X at `/System/Library/Frameworks/PCSC`. The support of smart card in MAC OS X is based on the Movement for the Use of Smart Cards in a Linux Environment (MUSCLE) Open Source implementation of the PC/SC standard.

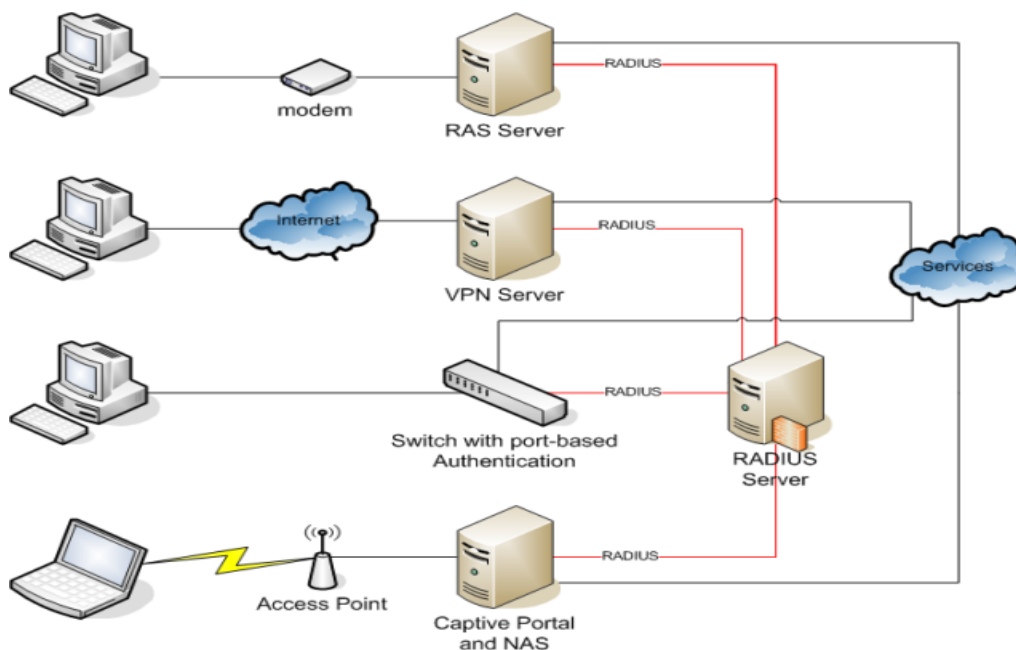
In MAC OS X v10.4 support login with smart card, the card must support signing with a public key. MAC OS X v10.4 and later version uses Keychain Services for supporting smart cards. It uses a plug-in called Tokend which serves as an interface that allows smart card developers to make their cards appear to be keychains. The three tokens installed with base MAC OS X are [74].

**Table 11:** Tokens installed with MAC OS

Token	Smart card specification
"CAC"	US Federal Government: CAC: Common Access Card GSC-IS: Government Smart Card Interoperability Specification v2.1
"BEPIC"	Belgian Personal Identity Card
"JPKI"	Japanese Public Key Infrastructure card

## 2.8 Remote Authentication Dial in User Service (RADIUS)

In a network we need to have a centralized Authentication and Authorization system to make the work of administrator much easy and the resources more accountable. For this purpose we can use Remote Authentication Dial In User Service (RADIUS) which is a networking protocol that can provide a centralized Authentication, Authorization and Accounting management system for computers to connect and use a network service [75, 76]. RADIUS is mostly used by ISPs and enterprises to supervise access to the Internet or any kind of internal networks which can be wired or wireless networks or integrated e-mail services. These networks may integrate modems, access points, DSL, access points, network ports, web servers or VPN's [75] which can be seen in figure 14 given below.



**Figure 14:** RADIUS Architecture [75]

RADIUS is a client server protocol which operates in the application layer, using UDP protocol for transport. The main functionality of the RADIUS server is as following

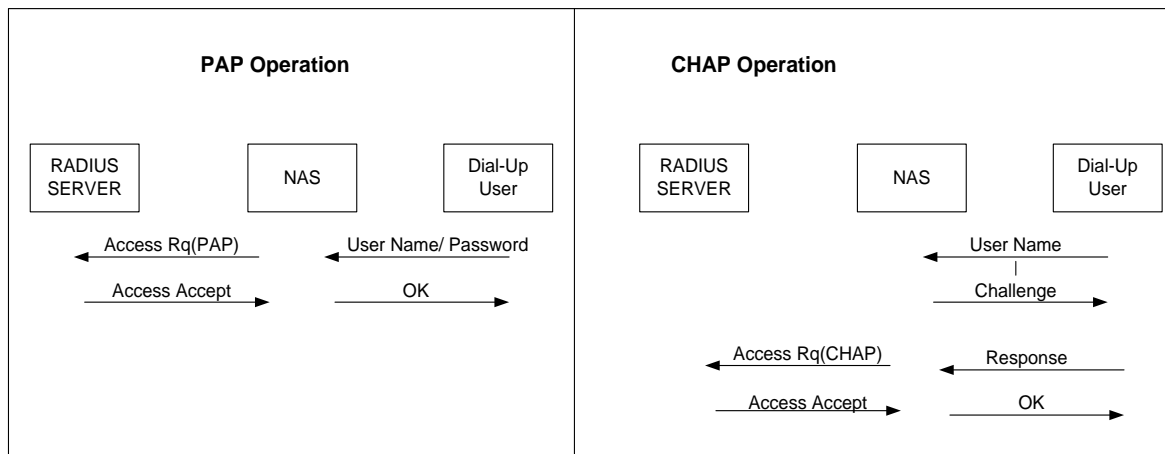
- The first task is to authenticate users or devices before giving them access to a network.
- The second task is to authorize users or devices for specific network services.
- The third task is to have an accounting for usage of those services.

### Core Messages

The core protocol of RADIUS is very simple and consists of basic four messages which are as following:

- Access-Request (From Network Access Server (NAS) to Authentication Server (AS))
- Access-Challenge (From AS to NAS)
- Access-Accept (From AS to NAS)
- Access-Reject (From AS to NAS)

From the above messages we can conclude that the point to point (PPP) dial-in modem protocol, have two options for authentication which are PAP and CHAP [76]. PAP is a simple method of authentication and requires the user name and password for authentication. While in CHAP a server is required to send a random number called challenge and the dial in system must encrypt and send it back for security checking. In CHAP password is not sent as clear but encrypted making it more secure than PAP. It is still vulnerable to dictionary attack because both the unencrypted and encrypted versions of the challenge are accessible to an attacker. The process of PAP and CHAP can be seen in the figure 15 given below



**Figure 15: RADIUS Different Authentication Operations**

### Core Message Format and Attributes

As there are four basic message but we can use attributes to change the other messages. The main structure of the RADIUS message consists of a series of attributes; each is a self-contained package of information that has meaningful message to both communicating parties. The structure of the message format can be seen in figure 16 as given below:

Code	Identifier	Length	Authenticator	Attributes.....
------	------------	--------	---------------	-----------------

**Figure 16: RADIUS Message Format**

**Code:** The code byte represents the type of the message e.g. 1 for Access-Request, 2 for Access-Accept, 3 for Access-Reject and 11 for Access-challenge etc.

**Identifier:** It is an arbitrary number used to match the request and response.

**Length:** The length field shows the number of bytes in the message.

**Authenticator:** The Authenticator field is 16 bytes (128 bits) long and depends on the type of the message. In Access-Request message this field contains the unique number called nonce. It is used for two purposes. One if password is being sent then it is used in combination with secret key for encryption. And the second purpose is it is used in reply message for integrity check. So we can see that in reply messages like Access-Accept, Access-Reject, and Access-Challenge, the nonce value is used to check whether the response is from a valid RADIUS Server or not.

**Attributes:** It is one of the fields that give RADIUS the power of extending itself to other technologies like EAP. Each message can carry one or more attributes and each is a self-contained package of information. Each attribute has the same format [76]:

- A Type field of one byte to identify the attribute
- A Length field of one byte that defines the number of bytes in the whole attribute
- Attribute specific data

## 2.9 Bluetooth security

Bluetooth is an open wireless standard which can be used for short range frequency communication. Bluetooth is integrated in many devices like PDAs, mobiles, laptops, printers etc, so we can transfer data without using cables.

Like normal wireless networks Bluetooth is vulnerable to service attacks, eavesdropping, man in the middle attack, and other attacks. Some of the attacks are: Bluesnarfing, Bluejacking, Bluebugging, Car whisperer, Denial of Service (DoS), Fuzzing attack. So, proper steps should be taken to make the communication secure.

Some of the features of Bluetooth that help in protecting from eavesdropping and malicious access to some extent are the frequency hopping and radio link power. But we need to take care of it so that our communication is safe.

### Bluetooth security modes

There are four security modes defined in various Bluetooth specifications which are as follows:

- **Security Mode 1:** In this mode no security is offered so attackers can attack easily.
- **Security Mode 2:** In this mode service level imposed security can be applied. First the Link Management Protocol (LMP link) is established and then the security procedures are initiated before Logical Link control and adaption protocol (L2CAP). L2CAP is in the link layer and provides connection oriented and connection less communication to the upper layers. In this mode of security the administrator can allow certain services to be accessed and can remove the access rights as well for some services. The authentication and encryption techniques are applied at LMP layer in this mode of security.
- **Security Mode 3:** In this mode the link level imposed security can be applied. And the security procedures are implemented before the physical link is established in this mode of security. In this mode all connections from and to the device are authenticated and encryption mechanism is applied to it making it secure. The authentication and encryption mechanism in this mode depend on the secret link key shared between the paired devices before the communication. In this mode like mode 2 the authentication and encryption is applied at LMP layer. This mode supports unidirectional and mutual authentication as well.

- **Security Mode 4:** This mode is like mode 2 where service level imposed security can be applied and security procedures are applied after the link connection establishment. In this method Secure Simple Pairing (SSP) uses Elliptic Curve Diffie Hellman (ECDH) for key exchange and generation. There are four classes defined in this mode:
  - Authenticated link key required
  - Unauthenticated link key required
  - No security required

Secure Simple Pairing (SSP) was introduced in Bluetooth v2.1 + EDR specifications. SSP surpasses the maximum security level provided by the use of a 16 character alphanumeric PIN in Bluetooth v2.0 + EDR and earlier versions [77]. Bluetooth v2.0 + EDR devices, operating in security modes 2 and 3, derive the link key from the shared secret PIN [78]. SSP does not use permanent shared secret PIN to derive the link key, ECDH exchange is used instead.

Secure Simple Pairing uses ECDH thus providing protection against passive eavesdropping [77]. It also provides protection against man-in-the-middle attacks [77, 78], but not in all association models.

Figure 17 shows the SSP link key establishment. The link key established here is used in next steps for authentication and encryption.

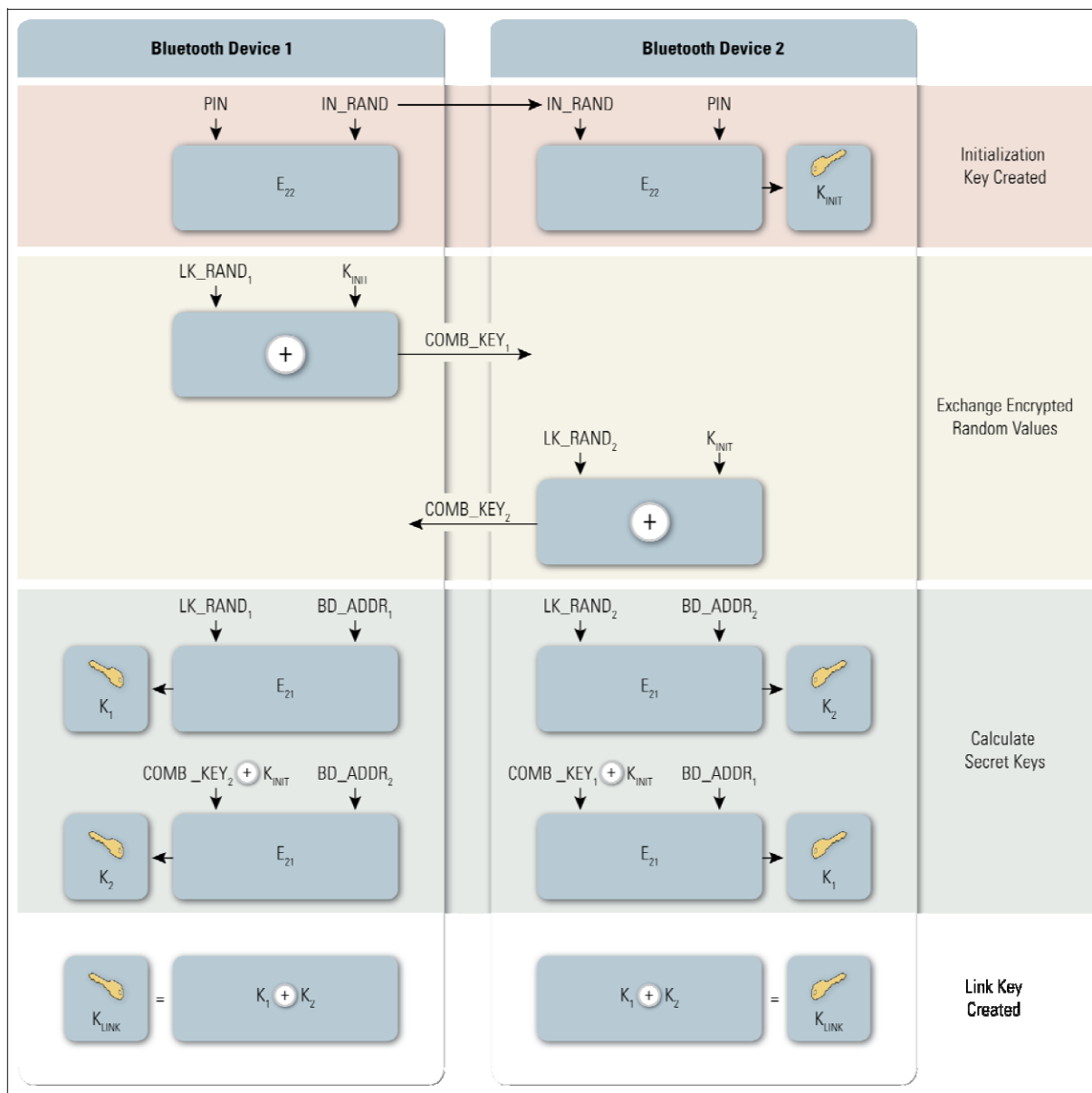


Figure 17: SSP Link Key Establishment for pairing [78]

SSP association models provide authentication service, and whether the link key is authenticated depends on the used model. The authentication procedure/pairing confirmation by the user according to these models take place in the SSP Authentication Stage 1. There are four models offered by the SSP:

- I. Numeric Comparison:** In this case it is considered that both the Bluetooth enabled devices have screens to show data and have “YES” or “NO” option to decide. A six digit is displayed on both screens while pairing and if match succeeds the user is given the option of “YES” on both devices. If the user selects yes the connection will be established otherwise the connection will not be established. The main benefit of it over using PIN is that the six digits are not used in link key, so if the attackers see it, will not benefit him. Numeric comparison model provides protection against man-in-the-middle [77].
- II. Passkey Entry:** This scenario is designed for those devices in which one have input and display capability but the other device only have display capability. So the device that has display capability shows six digit number, which is entered by the user in the other device with, input capability. In this case like numeric comparison the six digits are not used to generate link key. This model also provides protection against man-in-the-middle attacks.
- III. Just Works:** This situation is designed for the devices which don't have display and input capability. In this case the authentication is done as in the numeric comparison case but the only difference is that the digits are not shown on the screen. Just Works model does not provide authenticated link key [78] and does not provide protection against man-in-the-middle [77, 78].
- IV. Out of Bound (OOB):** This designed for such cases where other means are used for other technology (e.g. Near Field Communication). In this case the user accepts the pairing via single push button. In this case the other technology used should be made secure to keep the system secure from the attackers.

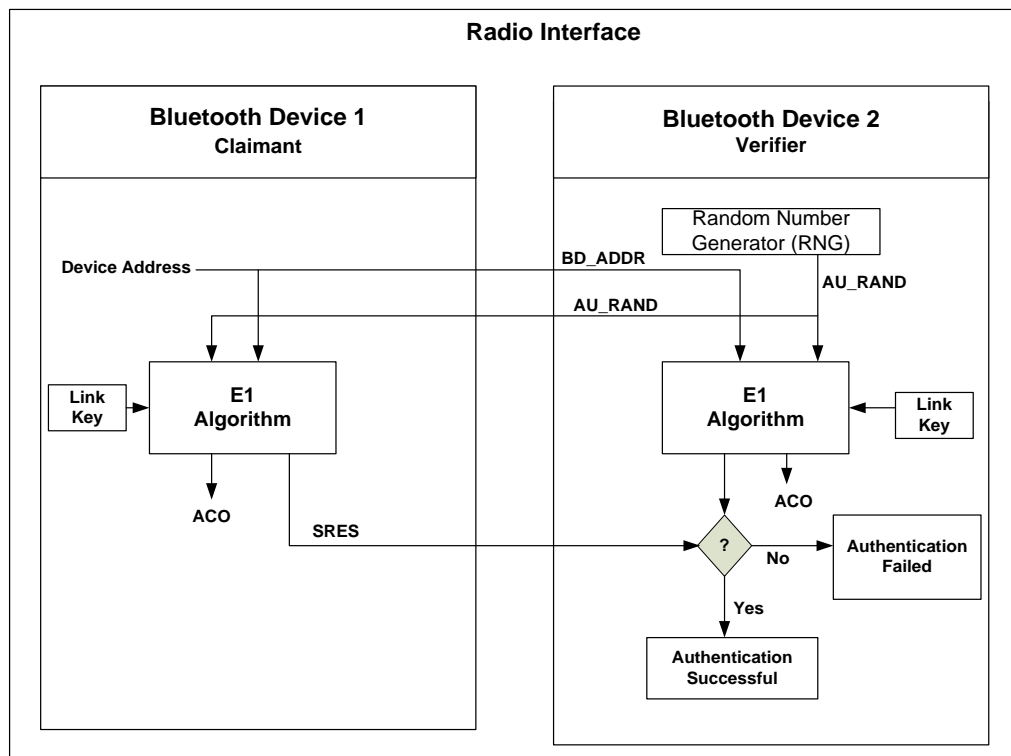
Authentication stage 2 is used to ensure that peers successfully completed exchange in Authentication stage 1 and to confirm the peering between devices. Unlike in Authentication stage 1, there is no user intervention in stage 2. After the peering has been confirmed the link key is generated from the derived shared key and some publicly exchanged values.

### **Bluetooth Authentication**

This phase starts after the link key was derived. In authentication process it uses the link key produced in the key generation process which can be seen in the figure 17. Bluetooth authentication uses a challenge-response protocol to authenticate the peer based on the knowledge of the link key. One device takes the role of claimant who has some identity and other is verifier who proves the identity of claimant. The authentication consists of the following steps:

1. At the verifier side it generates a random challenge of 128 bits (AU\_RAND), saves a copy of it and sends it to claimant side as well.
2. The claimant has now the unique 48 bit **Bluetooth Address** (BD\_ADDR), the **link key** generated before and the **Random number** received.
3. The claimant side uses **secure and fast encryption routine** (SAFER). The input to the algorithm is BD\_ADDR, AU\_RAND and link key, which produces a 128 bit output.
4. The most significant 32 bits of the output (SRES) are used for authentication while the remaining 96 bits which is called Authenticated Ciphering offset (ACO), is used in encryption in later stages.

5. The SRES (32 bits) is send to the verifier side.
6. Step 2 to 4 is also performed at the verifier side.
7. It compares both the SRES. If matches than authentication is successful, otherwise authentication fails. The procedure is also shown in the figure below:



**Figure 18:** Bluetooth Authentication [42]

For the mutual authentication the claimant and the verifier must switch roles.

### Bluetooth Encryption

There are three modes of encryption used which are as following:

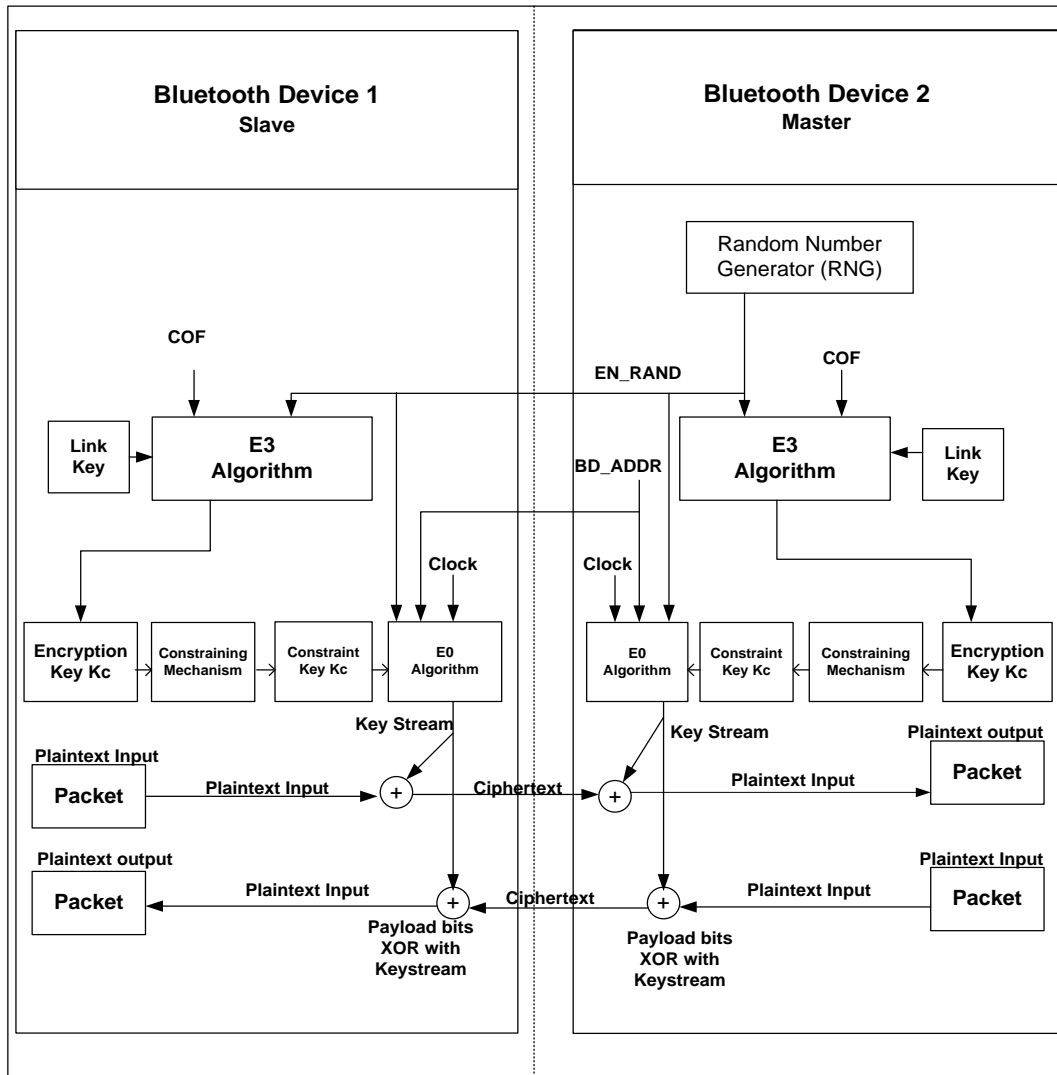
- **Encryption Mode 1:** No encryption is used in this mode.
- **Encryption Mode 2:** In this mode individual traffic is encrypted using keys generated from the link key produced at first stage, but the broadcasted data is not encrypted.
- **Encryption Mode 3:** In this mode all data is encrypted using keys generated from the link key.

The steps involved in the encryption are as following:

1. The keys for the encryption are produced using internal key generator (KG).
2. The internal key generator produce steam cipher keys (Kc) established from 128 bit link key, 128 bit random number (EN\_RAND), 96bit cipher offset (COF) which is the Authentication off set (ACO) produced in the authentication phase.
3. The stream cipher key (Kc) varies from 128 bits to 8 bits. There is key negotiation between the master and the slave Bluetooth devices. By default the key size is kept 128 bit to provide maximum security but can vary. The least security is 8 bits keys, which will make easy for attacker to attack.

4. The encryption Algorithm E0 as shown in the diagram below, used is based on the linear feedback shift register (LFSR).
5. The input to the algorithm is master identity (BD\_ADDR), 128 bit random number (EN\_RANDOM), a slot no and encryption key and the output is XORed with the plain text. (The slot no changes with each packet making some change with each packet).
6. The linear feedback shift register (LFSR) is initialized before sending a packet. The LFSR is reinitialized with each packet with having other static values and the in this case the slot no helps in bringing change with each packet.

The encryption process is shown in the following diagram below:



**Figure 19: Bluetooth Encryption [78]**

Bluetooth Security mode 4 with SSP is the most secure solution to be used for device peering. Besides, this mode, unlike legacy modes, does not require the knowledge of the shared secret PIN from both sides. The “Just works” association model being susceptible to the man-in-the-middle attack cannot be used in environments where high security is required. Thus we consider the SSP “Numeric comparison” and “Passkey entry” association models to be suitable solution for the secure dynamic establishment of the Bluetooth channel.

### 3. Mobile Phone Based Authentication Systems

The process of Authentication is verifying the identity of a user, user device, or some other entity as defined in [79]. For authentication it required to present a proof of identity to the authenticating party. All authentication schemes are based on the combination of the facts that something you know, something you have and something that you are. One of the most common authentication schemes nowadays used is static password authentication. But it has many weaknesses which are:

- It is hard to remember the strong randomly generated passwords that consist of the combination of letters, numbers, and special characters with adequate length.
- The human chosen passwords are often compromised with a simple dictionary attack;
- Several user-name/password systems make it hard to remember all data which consequently costs some users either with reuse of passwords or write them down.
- A Too strict password policy likes hard to remember passwords or frequent change can instead of strengthening the overall security of a system can actually weaken it, as users can end up writing down passwords.

The use of default passwords and careless users who reveal their passwords either accidentally or because of social engineering attacks can further reduce the security level of the system.

The username/password authentication may be sufficient for some systems but systems that have sensitive data require stronger authentication schemes e.g. biometric, smart card based, or one-time password based authentication schemes are considered to be much stronger than the ordinary user-name/password authentication, still the cost of the deployment and maintenance makes these systems less common.

An authentication solution that uses a mobile phone with a UICC card as a security token provides much stronger level of security as compared to the user-name password authentication with reduced operational costs [3].

#### 3.1 General mobile phone authentication schemes

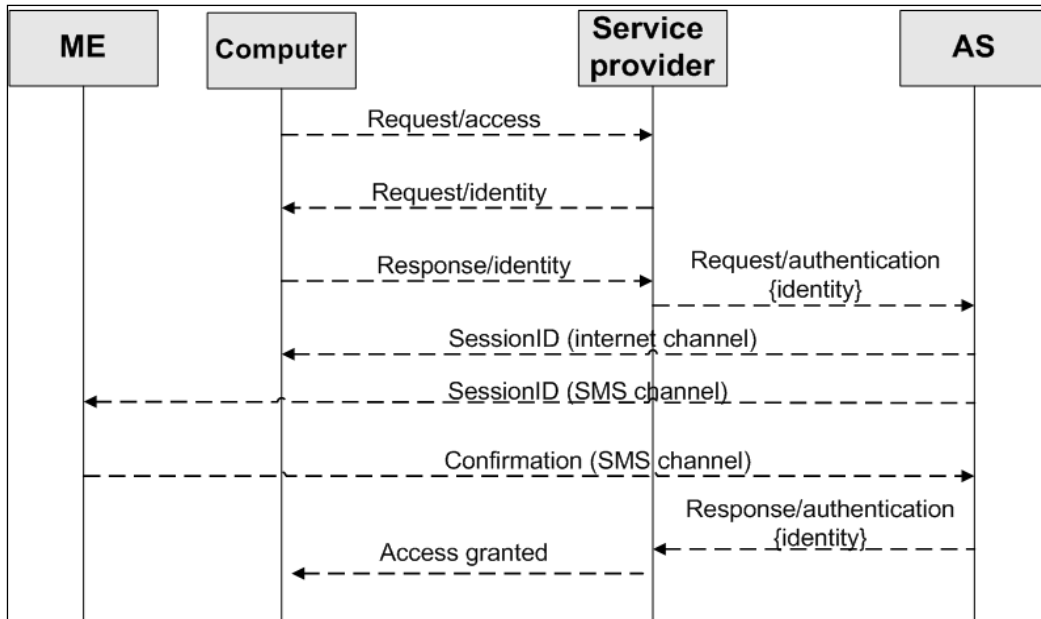
It is important to make sure that the same user controls both the devices in authentication scheme that use two devices (a computer and a mobile phone) and multiple channels for authentication protocol exchange (internet connection from the computer to the authentication server and GSM connection from the mobile equipment (ME) to the authentication server) [3].

##### 3.1.1 SMS authentication with session-ID check

In this mechanism the mobile network operator (MNO) has a role of an IDM provider. The basis of this authentication scheme is on the fact that a user is already authenticated in the GSM/UMTS network. Therefore the authentication process consists of the steps that make certain that the owner of the ME and the computer are the same. For verification a sessionID is send to both devices, to computer over the Internet and to the ME over GSM/UMTS network. The ME owner or the ME itself (automatically) matches the received sessionIDs with computers and sends a confirmation message via SMS to the authentication server. The Authentication procedure completes when the AS receives the confirmation SMS. Figure 20 shows the procedure which can be seen as following.

A service provider is client of the Authentication Server (AS), it outsources authentication to the AS.





**Figure 20:** SMS authentication with sessionID check

For automatic verification of sessionIDs by the ME a Bluetooth connection is required between the computer and the ME.

A mutual authentication is not provided by the described authentication scheme as it only authenticates the user to the service provider and the security of this scheme relies on the security of the underlying GSM/UMTS network which is much stronger in case of the UMTS network. This scheme does not have any explicit integrity protection as well. SMS forgery threat can be avoided by the GSM/UMTS security mechanisms. Still it is possible to spoof SMS sender address [3].

In case if the attacker controls one of the intermediate nodes between the user's computer and the service provider the following attack can be easily executed by an attacker. When the user starts the authentication process to the server provider, at the same time attacker also starts the authentication (supplying the identity of the victim) to the same or the other service provider that uses services of the same AS. The attacker blocks the original request of user so that it does not reach the intended service provider. It responds with a forged Request/identity to the user. When the attacker receives the sessionID through the Internet channel from the AS, he sends it to the victim. Now the user will receive two similar sessionIDs as one send by the attacker through Internet channel and the second one sent by the AS via SMS. The user (the ME in case of automatic verification) does not have any method to check that the received sessionIDs were actually issued by the AS to provide authentication service for the computer with different internet address and possibly for the different service provider. So the user will issue a confirmation SMS and the AS will reply to the service provider that the authentication was successful.

In case of the automatic sessionID verification by the Mobile Equipment (ME) using the Bluetooth connection for communication with the computer, it is possible for the attacker to authenticate to any service provider registered at the AS even without the user noticing it (unless the PIN is required to issue confirmation message), in case if the Bluetooth security is compromised. Still, the attacker would need to be in the close proximity to launch this attack.

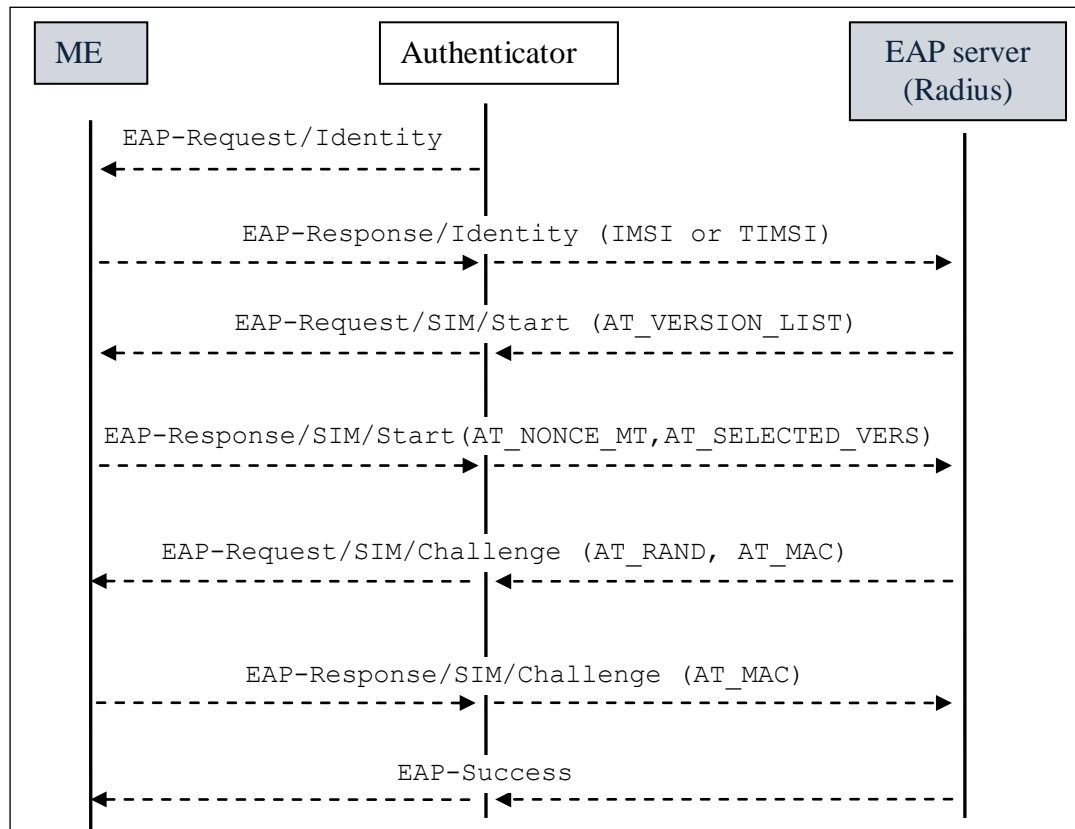
### 3.1.2 SIM Strong authentication

The SIM Strong makes use of the EAP-SIM protocol to provide mutual authentication [4]. In this method the MNO has a role of the IDM provider. Therefore the service provider depends on the MNO to perform authentication.

The EAP-SIM authentication scheme provides higher level of security as compared to the GSM authentication scheme. The EAP-SIM provides enhancements to the GSM Authentication and Key agreement (AKA) procedure which are discussed as following [80]:

- For deriving the Master key a 64-bit long GSM encryption key Kc is used which is not used directly.
- To create authentication responses and session keys of greater strength than the individual GSM triplets, the multiple authentication triplets combined for this purpose.
- The Master key is used to derive the Transient EAP Keys for protecting EAP-SIM packets, a Master Session Key for link layer security and Extended Master Session Key.

The EAP-SIM provides mutual authentication, integrity, confidentiality and replay protection.



**Figure 21:** EAP-SIM authentication

SIM Strong authentication can be run either through Internet and Bluetooth channel or via SMS channel by using the sessionIDs [4]. In case of Internet and Bluetooth SIM Strong authentication, the ME has Bluetooth connection with the computer and the EAP-SIM authentication communication is performed through the Bluetooth channel and Internet channel, not utilizing the GSM radio channel. The complexity and the number of the messages in the EAP-SIM exchange makes it impractical for the user to manually perform this exchange without using Bluetooth. Bluetooth security should be considered while using this scheme. By compromising the Bluetooth security the attacker has a chance to trick the ME to communicate with attacker's computer and perform the authentication for the attacker.

SIM Strong authentication that uses the SMS channel must also have the mechanism to make sure that the same user controls both the computer and the ME. This mechanism is done with sessionIDs. A sessionID is generated by the devoted applet that resides on the SIM card [3]. The generated sessionID can be entered by the user in the computer and transferred through Internet channel to the AS. After that the applet performs the mutual EAP-SIM authentication with the AS over the SMS

channel. The advantage of this scheme is that it does not require Bluetooth connection [4].

Both types of SIM Strong authentication need specialized applet on the SIM card to perform EAP-SIM [4].

### 3.1.3 EAP-AKA

As defined in RFC 5448, EAP-AKA is a mechanism for authentication and session key distribution that is used in UMTS Authentication and Key Agreement (AKA) mechanism. Unlike EAP-SIM which is based on the GSM AKA, this authentication protocol is designed to work with third generation networks. EAP-SIM and EAP-AKA have many common ideas as they were developed in parallel [97].

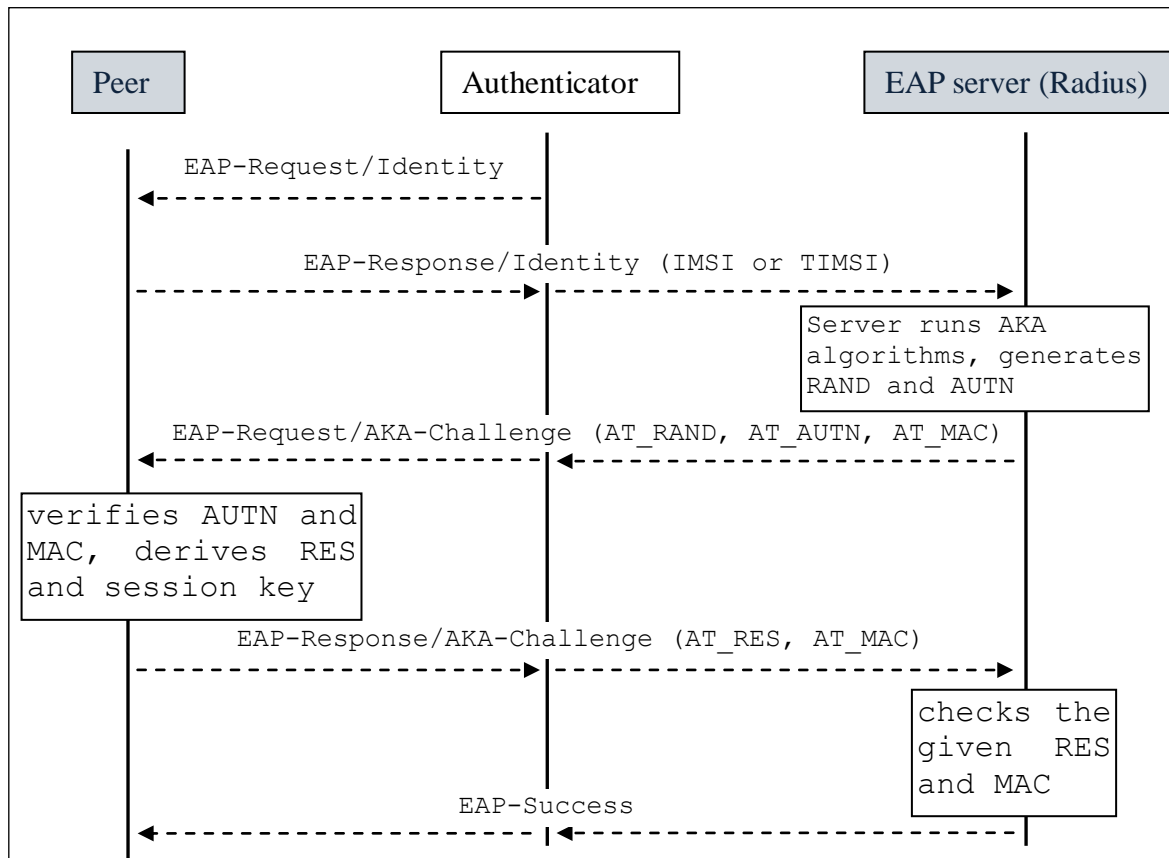


Figure 22: EAP-AKA authentication

As EAP-SIM, the EAP-AKA also provides mutual authentication, key derivation, confidentiality protection, integrity protection and optional identity privacy protection. EAP-AKA can also be used for SIM Strong authentication in the same manner as EAP-SIM.

## 3.2 One Time Password schemes

One of the secure methods of authentication is one time passwords (OTP). As the name suggests, OTP is used for authentication only once and after expiration of a session is useless. Therefore OTP is not susceptible to replay attacks. We can see the use of one time password (OTP) device in banks. Banks issue OTP tokens to their customers for online transaction authentication.

OTP use the concept of randomness. Randomness is an important factor because if passwords are not random then the attacker can guess the new passwords from previous observations. There are various methods used for OTP generation:

**Mathematical Algorithms:** We can use mathematical algorithms which can generate passwords

from the previous data. One method proposed by Leslie Lamport [81] uses one-way hash functions. It works by starting with generating an initial seed value. Then a set of one-time passwords is generated by applying a hash function multiple times to the initial seed. Every next password is generated by taking the previous passwords as seed and hashing it. The security of this scheme is based on the non-reversibility of the hash function. In this technique we have to change the seed when the set of password generated from one seed exhausts.

**Counter-based:** the scheme is based on the ever increasing counter and a secret shared by the token and the AS. RFC 4226 describes a counter-based scheme that uses HMAC-SHA-1 function to generate OTP value.  $HOTP(K, C) = \text{Truncate}(HMAC\text{-SHA-1}(K, C))$ , where K is a shared secret key and C is a counter value [82]. Truncation is used to enable the user to easily enter the resulting value in computer. It is important for the counter to be synchronized between a token used by the user and the authentication server.

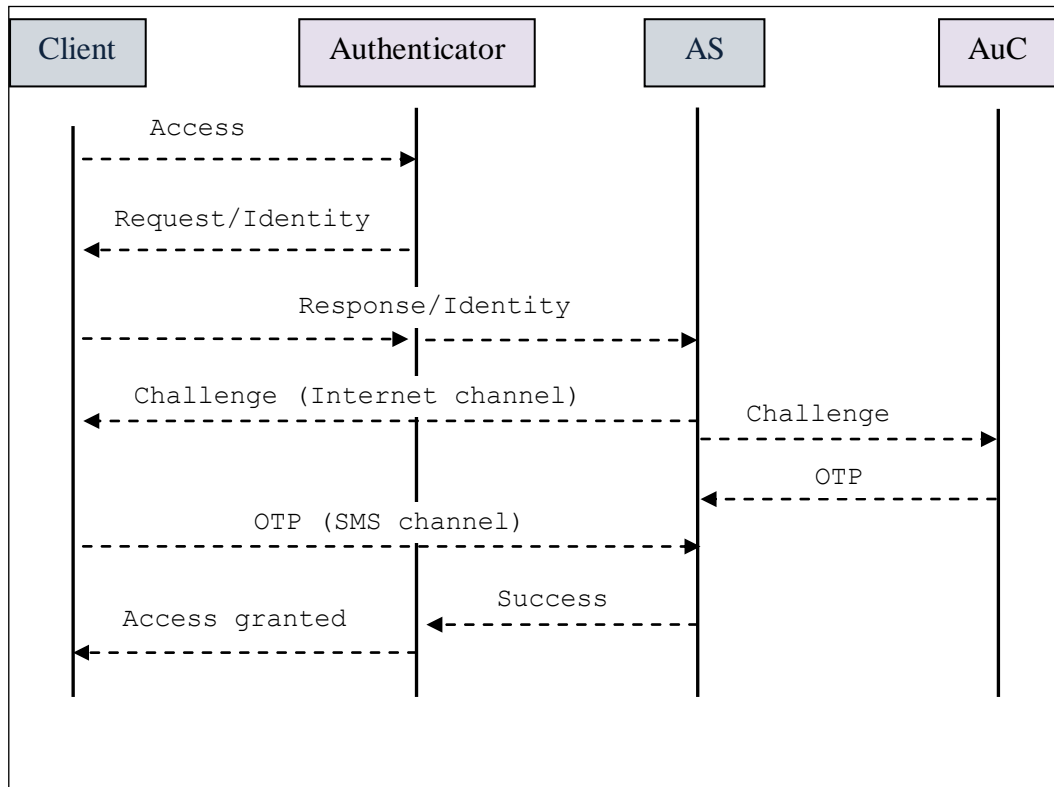
**Time-based:** This approach is based on the time synchronization between Authentication Server and the user who provides password for his authentication. The user is given some device which needs to be synchronized with the server to generate time-based OTP (TOTP). Usage of time as an input parameter ensures that OTP values do not repeat. As described in [83], the TOTP value can be generated by hashing the shared secret and a time parameter. Time synchronization is crucial for the TOTP scheme.

**Challenge Approach:** this scheme is based on randomly challenge values and shared secrets. In this method a verifier gives a randomly generated challenge to a party that wants to be authenticated. Based on the shared secret and the received challenge value the OTP password is generated. OATH Challenge-Response Algorithms, described in [84], provides one-way or mutual authentication based on the challenge-response concept. The advantage of this scheme is that parties do not need to maintain synchronization of any values.

### 3.2.1 One Time Password from PC to SMS

This scheme is a multi-channel challenge-based OTP authentication system. It is a one way authentication scheme. Only the user is authenticated to the service provider. The following exchange take place between the client and the service provider as can be seen in the Figure 23.

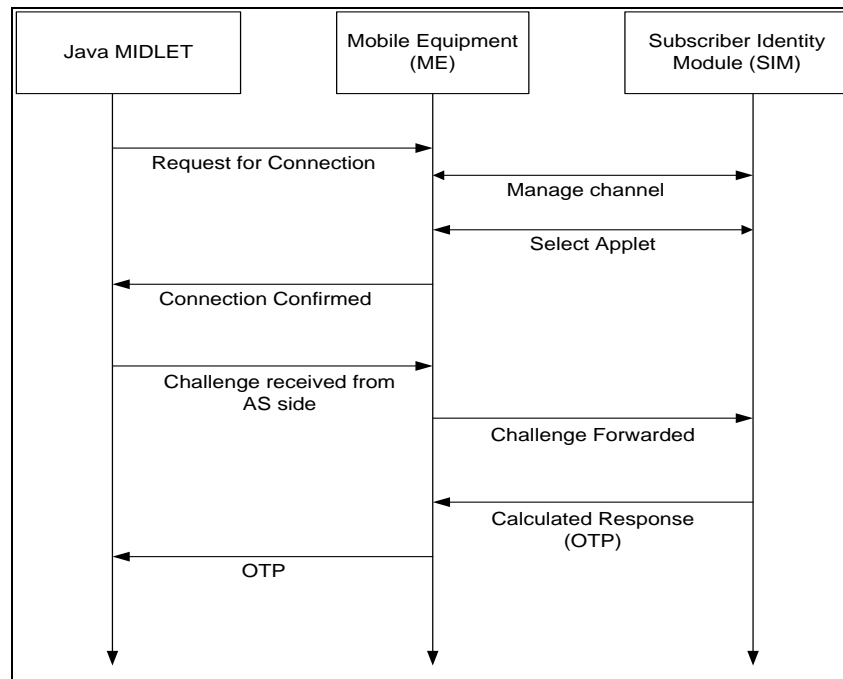
- Client requests for access to the service provider (SP).
- The service provider asks to prove his identity.
- The client writes his username in the browser and sends it to the service provider which forwards to authentication server (AS) for authentication process.
- When the Authentication server receives the credentials of the client it generates a challenge which can be a random number taken from the client data stored on the server side.
- It send the challenge to the client through Internet channel
- When the client receives the challenge he enters the challenge on the mobile phone.
- There is an OTP applet in the SIM. When the challenge is entered the OTP applet calculates the OTP.
- The OTP is send to the Authentication server through SMS.
- When authenticator receives the OTP it compares it with the one it generated itself.
- If both are same the client is authenticated and now the service provider can provide the required services.



**Figure 23:** OTP from PC to phone authentication [3]

When the user receives the challenge from the AS via Internet channel he can either enter this value to the ME manually, or a Bluetooth connection between the computer and the ME can be used to transfer challenge. The process of generating OTP on mobile consists of the following steps which are also summarized in the Figure 24.

- There is Java MIDlet installed on the ME.
- The user can start the MIDlet manually or, if we want the MIDlet to interact automatically then we can use Wireless Manager API for this purpose. The MIDlet communicates with the SIM using SATSA-APDU as can be seen in the figure given below.
- The challenge that is entered is communicated to the OTP Applet which generates OTP from the challenge.
- The generated OTP value is sent through SMS to the AS either manually or automatically using Wireless Manager API.



**Figure 24: OTP Applet [3]**

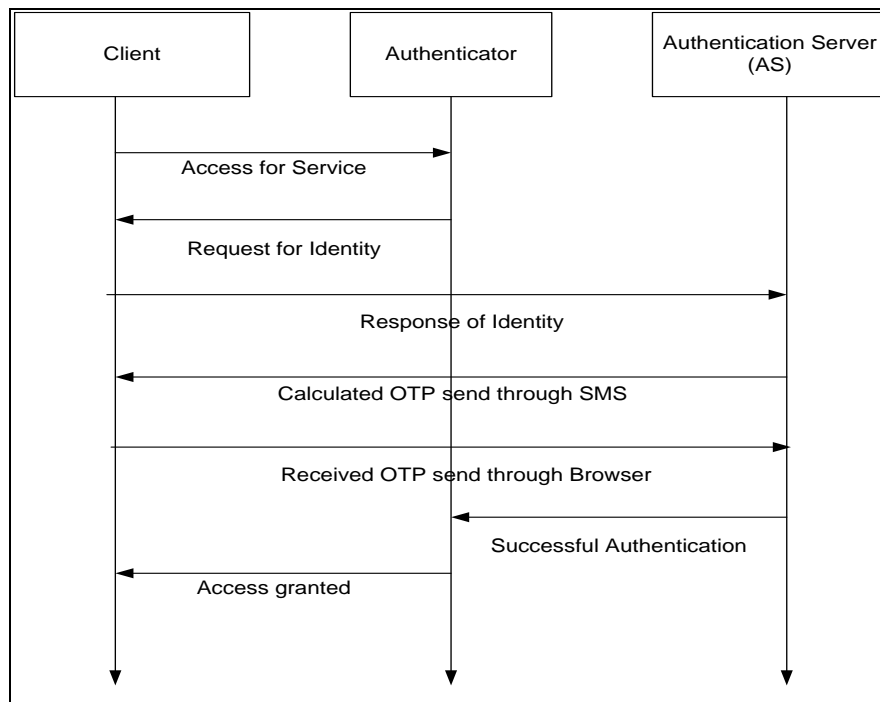
The OTP PC to SMS scheme requires a dedicated OTP generation application on the SIM card. By sending the OTP value via GSM/UMTS SMS channel the user assures AS that he controls both the computer and the ME. The security is based on the assumption that only the legitimate subscriber's SIM can generate OTP and on the fact that the radio channel is encrypted, making it almost impossible to intercept SMS and extract OTP value.

This scheme neither provides confidentiality protection nor temporal key derivation. Besides, since this scheme does not provide mutual authentication and integrity protection it is susceptible to many attacks. As in SMS authentication with sessionID check scheme, it is also possible to launch session hijacking attack. This attack can be easily executed if the attacker controls one of the intermediate nodes between the user's computer and the service provider. When the user starts authentication to the server provider the attacker also starts authentication (supplying the identity of the victim) to the same or the other service provider that uses services of the same AS. The attacker blocks the original request from the user so that it does not reach intended service provider, and responds with a forged Request/identity to the user. When the attacker receives a challenge via Internet channel from the AS, he "forwards" it to the victim. The user will receive the challenge from the attacker. The computer/user/ME has no way to check that the received challenge was actually issued by the AS to provide authentication for the user's session. Believing that this challenge was intended for him the user will unblock OTP generation function by supplying the PIN to the ME, the OTP value will be generated and sent to AS. AS will check whether the received and self-generated OTP values match and will authenticate attacker.

If Bluetooth connection is used between the ME and the computer, it should be well protected. The attacker can authenticate to any service provider registered at the AS even without the user noticing it (unless the PIN is required to issue confirmation message), if the Bluetooth security is compromised.

### 3.2.2 One Time Password from SMS to PC

In this architecture we consider that the user with mobile phone is authenticated by GSM/UMTS network. The main difference between this architecture and the previous architecture is the processing done at the server side, making user free from processing tasks. The steps involved in this architecture can be seen in the figure below



**Figure 25: OTP SMS to PC authentication [3]**

The authentication exchange consists of the following steps:

- User requests Access to some service provided by the service provider (SP)
- The Service provider asks for user's credentials for authentication
- The user provides the username which is forwarded to the AS
- The AS generates the OTP value
- The generated OTP is forwarded to the client mobile through the SMS channel
- The Client inserts the OTP received through SMS either manually or automatically using Bluetooth technology
- If the OTP generated at the server side and received from the user are same than the user is authenticated.

In the automatic version of this architecture we use java applet to communicate with SIM using SAP. When the mobile will receive the SMS it can notify the Java applet running on client's computer and the SMS can be retrieved from the mobile.

The OTP SMS to PC scheme utilizes the fact that the user is already authenticated in the GSM/UMTS network. Thus the AS needs to confirm that the owner of the ME actually controls the computer. This is done with OTP exchange. Although the OTP value is sent to the user's ME via SMS channel, it is not used to provide mutual authentication. The only difference between the randomly generated sessionID that could be used and the OTP value is that the latest is actually associated with the HTTP session created by the user [3]. To do session hijacking the attacker needs to send OTP value from his computer. However, the attacker cannot intercept this value on the radio channel since this link is protected by GSM/UMTS security mechanisms. Only if Bluetooth is used between the ME and the computer can the attacker obtain OTP value by compromising the Bluetooth security. Though there is no need for the attacker to do this. By intercepting the OTP value, sent back to the AS via Internet channel, and sending it from his computer attacker can hijack the session. Since the OTP value is bound to the HTTP session, the attacker cannot authenticate to arbitrary service provider if the user's computer performs OTP check.

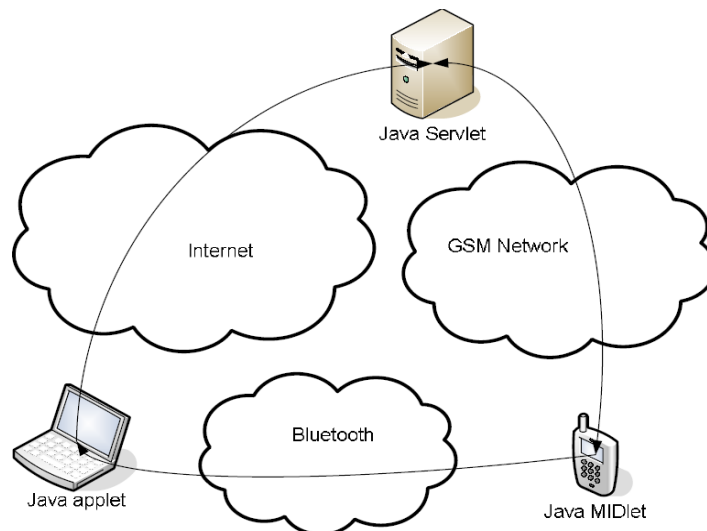
If Bluetooth connection is used between the ME and the computer, it should be well protected. The

attacker can authenticate to any service provider registered at the AS even without the user noticing it (unless the PIN is not asked to authorize the transfer of OTP value to the computer), if the Bluetooth security is compromised.

Session hijacking is possible since this scheme does not provide integrity protection. It also does not provide confidentiality protection and temporal key derivation mechanism.

### 3.2.3 Enhanced OTP from PC to SMS authentication

This architecture, described in [5], is an enhanced version of the multi-channel challenge-based OTP from PC to SMS solution, which provides integrity protection. The components architecture is shown in the diagram given below:



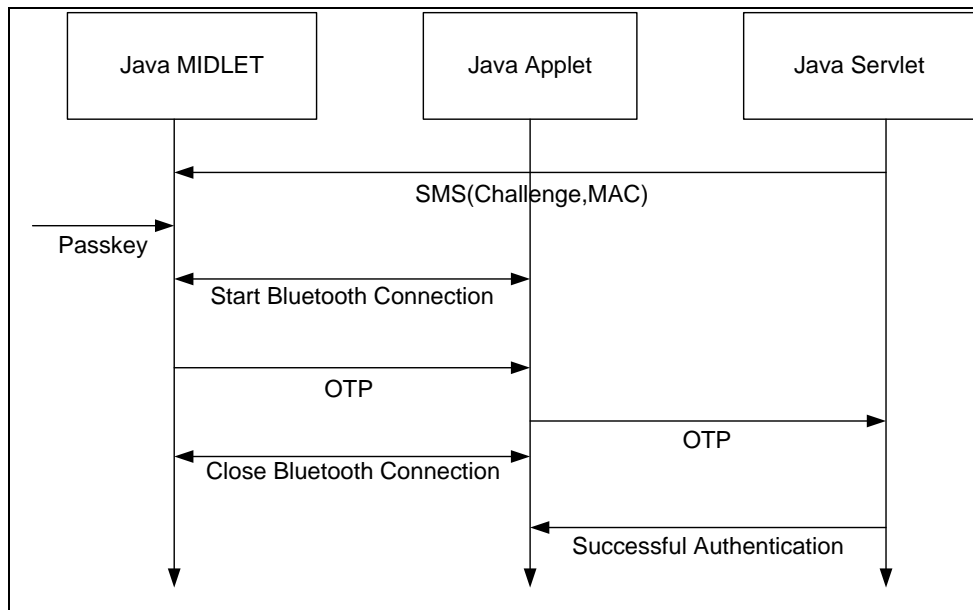
**Figure 26:** Component Architecture [5]

The steps involved in the authentication process, shown in Figure 27, are as follows:

- When the user wants to get some service he accesses the server, presents his user name to the Java Servlet present at server side.
- The server generates a challenge and it computes the OTP value in the following way:  $OTP = \text{hash}(\text{challenge} \parallel \text{secret key})$
- The message authentication code (MAC) is generated on OTP.
- The challenge and the MAC are sent to the MIDlet through the SMS channel.
- Upon receipt of the SMS, the MIDlet in the ME is activated and asks for user password. When the user enters the password it is hashed and then compared with the stored value.
- After checking the password, the password is used to decrypt the secret key.
- The challenge received is concatenated with the secret key and the result is hashed to generate OTP at the MIDlet side.
- Message Authentication code (MAC) is generated on OTP and matched with the MAC received through SMS message. If it is different the procedure is aborted.
- On successful match the MIDlet start communicating with the Java applet running on the computer through Bluetooth. The OTP value can be entered in the computer manually without the usage of Bluetooth, though it should be truncated for usability.
- The Java applet sends the OTP value to the Java servlet via Internet channel, and the connection between Java MIDlet and Java Applet is aborted.



- The server compares the self computed OTP with received OTP. If they match the user is authenticated.



**Figure 27:** Authentication process

**Java MIDlet:** It is one of the important components of the ME as it gives the capability of OTP generation and communication with the Java Applet and Authentication server. Java MIDlet can be downloaded by registering it on the website. After registration a push message is sent to the user which enables him to download and install it. When the MIDlet is installed, keys are exchanged between authentication server and the MIDlet at the ME through SMS. After the MIDlet completes key exchange, it starts working.

There are 2 passwords used for authentication. A PIN code selected by the user (used to provide two factor authentications) and the other one is the OTP, which is generated by the Server and shown on the web page when the Java MIDlet is downloaded. This initial OTP is used to authenticate the key exchange.

The wireless Messaging API (WMA) enable the MIDlet in the ME to send and receive SMS messages. The SMS communication can be explained in the following steps:

- The MIDlet is registered with a port and protocol in such manner that if the SMS message is received at the port, the Application Management Software (AMS) forwards it to the MIDlet. The registration with the port can be done statically by using the Java ME application descriptor (JAD) file.
- At the server side the SMS message is sent at the specified port of the mobile using the desired protocol.
- When the AMS receives the SMS message at specified port it forwards it to the MIDlet as MIDlet is listening at the specified port.

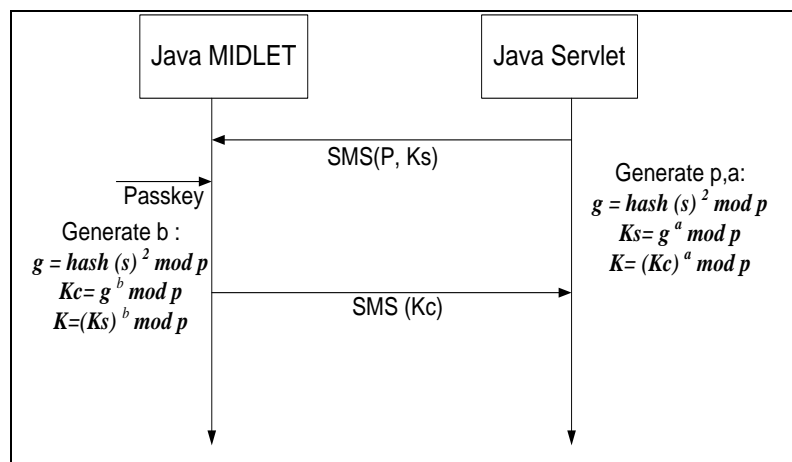
**Java Applet:** It serves as server for MIDlet and helps MIDlet to connect with the computer through Bluetooth. For communication between Java MIDlet and servlet, established Bluetooth connection should be present. A 16 byte pass phrase is used while connecting through Bluetooth to keep the Bluetooth connection protected. When the applet receives the OTP it sends it to the servlet at the server where it is matched with the computed OTP.

**Java Servlet:** It acts as an authentication server to clients. When the user downloads the MIDlet it

is registered with the Servlet. At the registration time the key exchange takes place to compute shared secret key  $K$  at the MIDlet and the servlet side. The key is saved with the user profile in the database. The servlet can be configured to send SMS at a specific port so that MIDlet can receive it on that port. The servlet asks for the user name when the user is authenticated for service. It creates a challenge and sends it to the MIDlet through SMS. The MIDlet computes the OTP and sends it to the servlet with the help of java applet that is connected with the MIDlet using Bluetooth.

At the beginning (after the MIDlet was downloaded) the Java MIDlet on the ME and the AS do not have shared secret to generate and check OTP values. The shared secret is derived via key exchange procedure. However, before the shared key is derived, the authentication procedure should be performed. When downloading a MIDlet the user is shown the OTP value on the server's web page. The user enters this OTP value to the MIDlet only once. This OTP value is used to authenticate the key exchange [5]. For the key exchange the Simple Password Exponential Key Exchange (SPEKE) protocol, which is an improved version of Diffie-Hellman, is used [5].

The key exchange procedure can be seen in the figure below



**Figure 28:** Key Exchange procedure [5]

The key exchange steps are the following [5]:

- At the server side a large random prime  $p$  is generated. Then value  $g$  is calculated as  $g = \text{hash}(s)^2 \text{ mod } p$ , where  $s$  is short OTP which is displayed in the browser after registration. The server also computes  $Ks$  using the following equation:
- $Ks = g^a \text{ mod } p$ , where  $a$  is the secret random number.  $Ks$  and  $p$  are sent to the MIDlet via SMS channel.
- At the MIDlet side the following values are computed:  $g = \text{hash}(s)^2 \text{ mod } p$ ,  $Kc = g^b \text{ mod } p$ , where  $s$  is the short OTP as in first step and entered by user;  $b$  is the secret random number. The computed  $Kc$  value is send to the server through the SMS message. Then the MIDlet computes the shared secret key  $K$  as:  $K = (Ks)^b \text{ mod } p$
- After receiving  $Kc$  the server computes the shared key  $K$  as:  $K = (Kc)^a \text{ mod } p$ .

Now both the servlet and the MIDlet share the same secret key  $K$ , which will be used to generate OTP. The key  $K$  is further encrypted using user selected password, so that if mobile is lost no one can take benefit of it unless the password is known. The key is then stored in the encrypted form in the Record store in J2ME. In the MIDlet a hash of user password is stored so that it can verify the correctness of the password entered by the user at the start of the procedure.

It is important to note that the user does not need to prove his identity, though he needs to prove that he controls both the computer and the ME. The security relies on the fact that the ME with

subscriber's UICC card is already authenticated by the mobile network operator. The OTP value that is used as the input to the hash function ensures that the user, with whom the key exchange via SMS is performed, is the same user that requested the service from the server (it means that the user controls both the computer and the ME). Only the legitimate subscriber could receive SMS messages, decrypt them, and compute shared secret K based on the initial OTP value and the values sent in SMS. The GSM/UMTS network provides encryption of the radio channel and ensures that SMS is forwarded to the destined receiver.

In the subsequent exchanges the user proves that he controls the computer by sending the OTP value, generated by the ME based on the shared secret and the challenge sent via SMS, via the Internet channel to the AS. Along with the challenge the MAC value (computed on OTP) is sent to the user via the SMS channel. After computing the OTP value, the user can check whether the challenge was sent by the party that knows the shared secret key derived during the key exchange phase. Only the party that knows shared secret could compute OTP value and calculate the MAC value of the OTP. Thus the user is ensured that the challenge is sent by the same server that displayed the initial OTP value on the web page. However, it is still not the mutual authentication, since the user does not check the identity of the server when making initial HTTP request to the server, thus masquerade attack is possible. The attacker can intercept the initial HTTP request, respond with the initial OTP value in the HTTP response to the user, go through the key exchange phase and derive the shared secret for OTP generation. This scheme provides only integrity protection. The initial OTP value cannot be used to authenticate the server (even if it were based on the secret shared by the server and the user) since it is transmitted openly to the user, and the attacker can intercept it.

Since the shared secret for the OTP generation is dynamically derived by the Java MIDlet, it is not mandatory for the MNO to act as IDM provider. The server can execute SMS exchange with the user directly. The advantage of this scheme is that the dedicated applet on the UICC card is not required, however the shared secret key used for OTP generation has to be stored in the encrypted form in the Java Record Store.

### **3.3 Proposed Solution for Mobile Based Linux Workstation Logon Service**

To design a solution for secure mobile based authentication for Linux workstation logon, it is required to make the solution more secure, user friendly and supported by the existing technologies. At the client (user) side a Pluggable Authentication module (PAM) is used to make the solution technology specific free. By using the PAM API and the functions we can access the user information and authenticate user at the server without taking care of the technology at the server side like LDAP, Kerberos etc. The PAM API converts the required messages to the technology.

The user information is saved at the authentication server. We require the user name and the mobile number so that we can send the One Time Password (OTP) to the user as SMS. The user information is accessed using the *passwd* structure. The *passwd* structure has the following information accessed from the server.

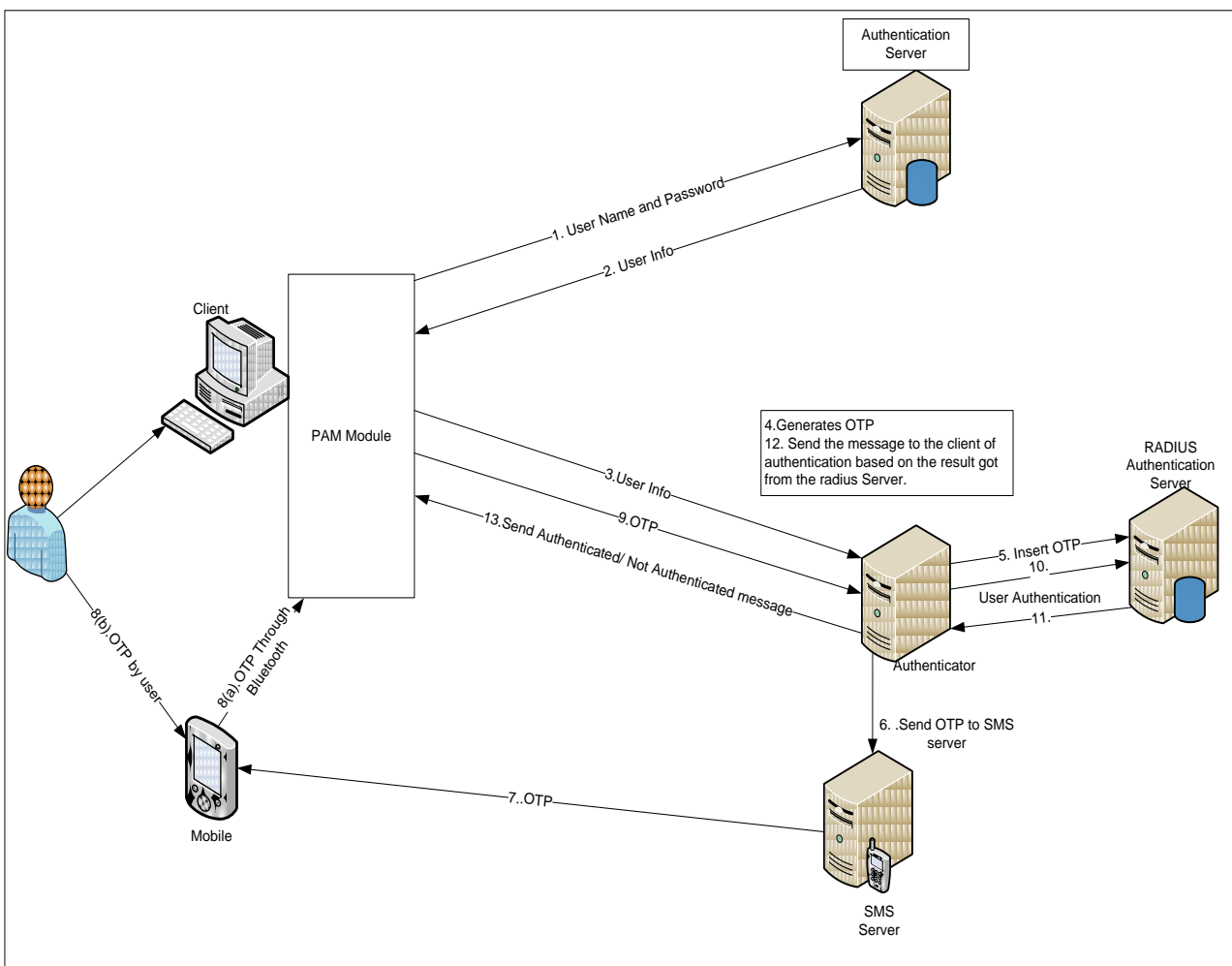
- User name
- User password
- User id
- Group id
- Real name
- Home directory
- Shell program

For getting the mobile number of the user we have suggested the use Real Name field to be used as Real Name, mobile number e.g. George, 004712345678. We will access the Real name field and get the mobile number from it as well.

After having the user authenticated from the existing technology and having the user information, we can pass it to the Mobile Based Authenticator that generates the OTP, send it to the user as SMS and insert the OTP in the RADIUS server against the user information. The communication between the client computer and Authenticator will be done by using socket communication. The Authenticator serves as the connection point for the client and the RADIUS Authentication server.

When the user will receive the OTP as SMS, they can insert it in the client computer manually or by using Bluetooth technology. The password will be passed to the Authenticator which will forward it to the RADIUS Authentication Server for authentication process. The Authenticator will send the message authenticated if the RADIUS server passwords matches with the OTP entered by user. Otherwise user will not be authenticated.

The process of the solution can be seen in the figure given below:



**Figure 29:** Mobile Based Authentication System for Linux Workstation Logon

The steps involved in the process are as following

1. The user enters name and static password which will be forwarded to the Authentication Server like LDAP, Kerberos Server etc.

2. The user will be authenticated by the static password which the user will have. After authentication the user information will be accessed and returned to the client.
3. At the client side the PAM module will check the user information. If it finds the mobile number in the Real Name field of *passwd* structure then it will continue with the second authentication process based on mobile. Other wise it will authenticate user based on the first authentication process. This option will give administrator the choice of allowing some users to use only the existing authentication technology and others using mobile based authentication technology.
4. The Authenticator serves as the connection point for the PAM module and the RADIUS Authentication server. It will generate OTP for the user.
5. The OTP will be inserted in the database against the user information in the RADIUS Authentication Server.
6. The OTP will be sent to the SMS server with the mobile number, so that it can be send to the user.
7. The SMS server will send the OTP to mobile phone.
8. Upon receiving the OTP the user will have the option of transferring the OTP from mobile phone to the client computer in following ways:
  - a. Using Bluetooth.
  - b. By user reading from mobile and entering on the Mobile OTP page on clientThe manual option will help for those user who do not have the Bluetooth capability in their mobiles.
9. The client will send the OTP to Authenticator for Authentication.
10. The Authenticator will forward the OTP and user name to the RADIUS Authentication server for Authentication process.
11. The RADIUS Authentication Server will send the result of Authentication process by comparing the OTP and the password of the user in the data base.
12. The Authenticator will forward the Authentication process result to the PAM module.
13. The PAM module will authenticate the user if user is authenticated by the RADIUS Authentication Server, otherwise the user will not be authenticated.

The solution which is proposed for the mobile based authentication for Linux workstation logon process will make the logon process more secure giving the administrator the option to select the users with their authentication method. The fact that the solution can be used with the existing technology and the infrastructure is already available, makes the solution less costly. Thus this makes the solution more suitable.

## 4. Analysis

This chapter consists of system requirements, Use cases, interaction diagrams and a proposed solution for secure mobile authentication for Linux log on, to specify the required functionality, components and the interfaces of the system.

### 4.1 Requirements

The requirements define the characteristics or features of the desired system. Those features which the system must perform are called *Functional requirements* and those features which are relevant to the system performance are called *Non functional requirements*.

#### 4.1.1 Functional Requirements:

The functional requirements of the system are as following

- Secure authentication of user for Linux workstation Log on in an enterprise, using mobile phone.
- A support for mutual authentication should be added.
- The system should support two factor authentications.
- A support for existing authentication schemes should be added.
- The password used during mobile phone based authentication should be strong enough to protect against dictionary attack.
- Sensitive data should be protected.
- The passwords used with mobile phone authentication should be used for once. (One Time Passwords)

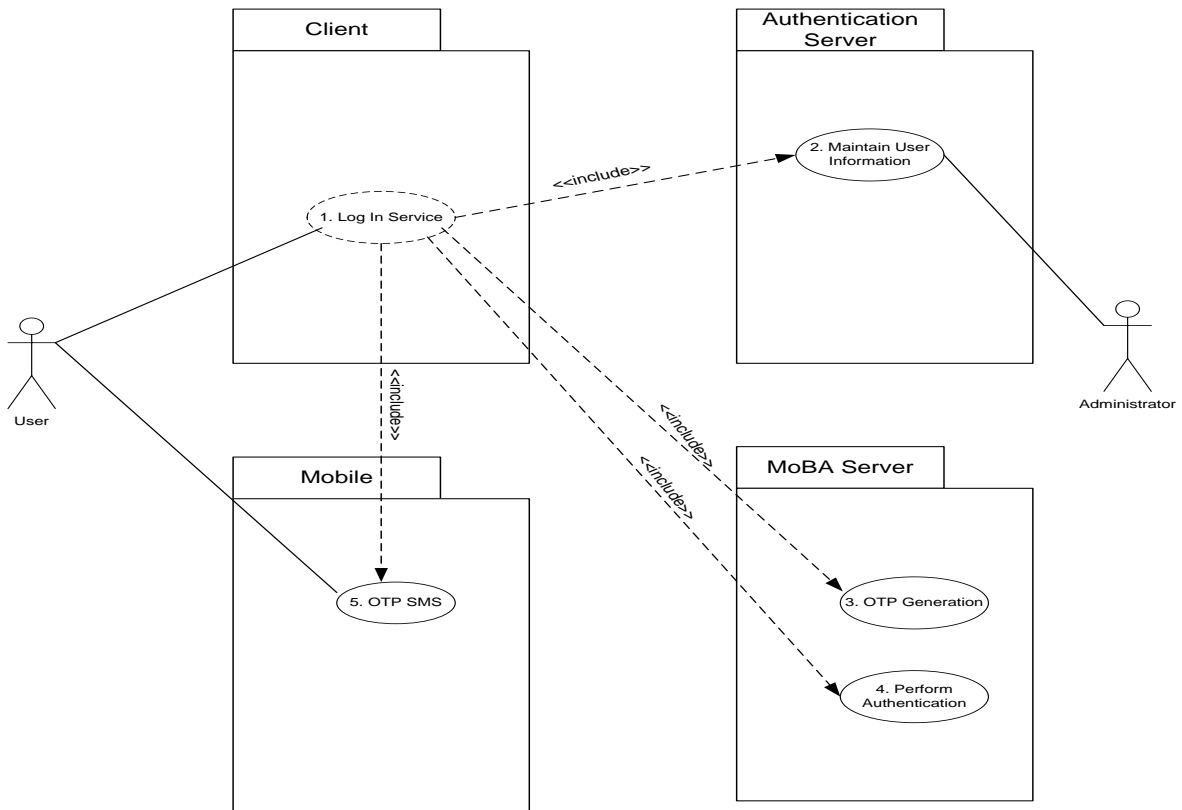
#### 4.1.2 Non Functional Requirements:

The non functional requirements of the system are as following:

- **User Friendly:** The proposed solution should be user friendly.
- **Cost:** The cost of deployment should not be more than the existing solutions. Cost of the solution should be less.
- **Scalability:** With increasing number of users the system should be easily scalable.
- **Performance:** The performance of the proposed solution should be good. The delay in the authentication should not be more than the existing systems.
- **Availability:** The system should be available all the time when the user requires it.
- **Reliability:** The system should be reliable. The percentage of error occurring should not be more than the expected value or existing systems.

### 4.2 Use Case Diagrams:

The use case diagrams are used for capturing the basic functional requirements of the system. Following are the general use case diagram in which functional requirements are defined



**Figure 30: General Use Case Diagram**

Figure 30 shows the overall overview of the system. The tables given below explain the main scenarios

**Table 12: Use Case-Login Service**

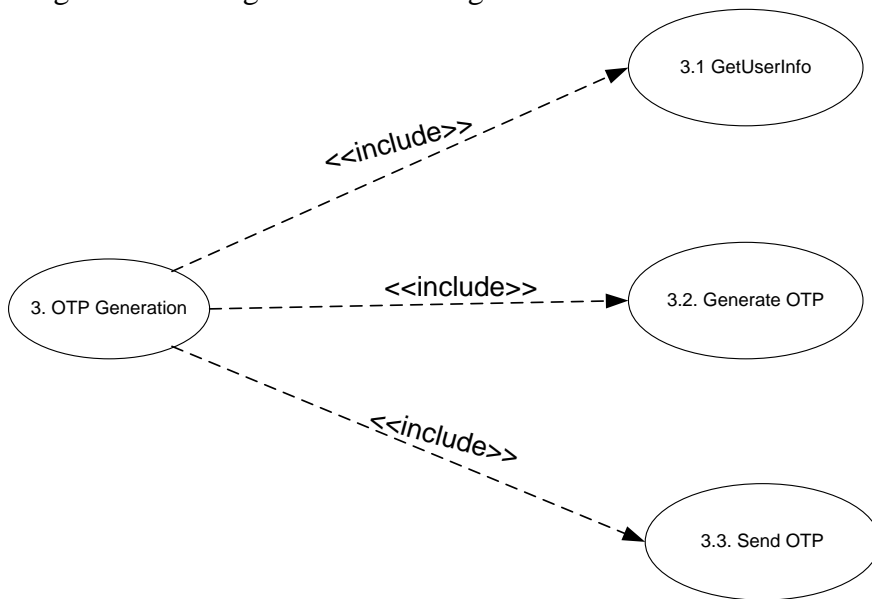
Use Case	Login service
<b>Description</b>	The user wants to login securely using mobile based Authentication (MoBA).
<b>Actors</b>	User Client Authentication Server MoBA Server Mobile
<b>Assumptions</b>	1. The client is connected with Authentication Server and MoBA Server. 2. The MoBA Server is connected to mobile phone using some wireless technology like GSM or UMTS. 3. The user provides the correct credentials.
<b>Steps</b>	1. The user enters the username. 2. The user gets the user information from Authentication server (Starts UC2). 3. The user information is then forwarded to MoBA Server (Starts UC3) 4. The OTP is generated and sent to user mobile phone. (Starts UC5). 5. The OTP is sent to MoBA Server by user or through Bluetooth from mobile to client and then to MoBA Server (Starts UC4). 6. And after that on the bases of OTP the MoBA server decides for authentication of user.
<b>Variations</b>	1. The user enters incorrect username. 2. The user enters the incorrect password.
<b>Issues</b>	

**Table 13: Use Case-Maintain User Information**

<b>Use Case</b>	<b>2. Maintain User Information</b>
<b>Description</b>	The user information is saved at the Authentication Server and the administrator has the right to assign rights to other users and can manipulate the users information
<b>Actors</b>	Authentication Server Administrator
<b>Assumptions</b>	1. The user can access his information. 2. The administrator can add new users and assign rights to users.
<b>Steps</b>	1. The Authentication Server receives the user name from client. 2. The Authentication Server fetches the data of the user in the database. 3. The data of the user is send back to the client
<b>Variations</b>	The information of user not found in the database.
<b>Issues</b>	

**Use Case 3: OTP Generation:**

The details of OTP generation are given below in Figure 31:



**Figure 31: OTP Generation Use Case**

The details of the above use case are explained in the table 14, 15, 16, 17, 18 and 19.



**Table 14:** Use Case-OTP Generation

<b>Use Case</b>	<b>3. OTP Generation</b>
<b>Description</b>	The client requests for password so that user can be authenticated.
<b>Actors</b>	Client MoBA Server
<b>Assumptions</b>	1. The user has entered valid username. 2. Data for user is found in the database. 3. The data is forwarded to the MoBA Server.
<b>Steps</b>	1. The client enters the username. 2. The data from database is retrieved for the username and sent to the MoBA Server. 3. The MoBA Server generates the OTP. 4. The OTP is forwarded to the user mobile phone using SMS service.
<b>Variations</b>	User enters invalid username.
<b>Issues</b>	

**Table 15:** Use Case-Get User Information

<b>Use Case</b>	<b>3.1. Get User Information</b>
<b>Description</b>	The MoBA receives the user information from the database in the Authentication server through client.
<b>Actors</b>	Authentication Server Client MoBA Server.
<b>Assumptions</b>	1. The Authentication server is connected to the client 2. The Client is connected to the MoBA Server.
<b>Steps</b>	1. The client receives the user information from the database in the Authentication server. 2. The client forwards it to the MoBA Server 3. The MoBA retrieves mobile number and other information from the data.
<b>Variations</b>	Invalid username entered and data cannot be found in the database.
<b>Issues</b>	

**Table 16:** Use Case-Generate OTP

<b>Use Case</b>	<b>3.2. Generate OTP</b>
<b>Description</b>	A One Time Password (OTP) is generated by MoBA Server.
<b>Actors</b>	MoBA Server
<b>Assumptions</b>	The valid username is provided to the MoBA Server
<b>Steps</b>	The MoBA Server generates the OTP.
<b>Variations</b>	
<b>Issues</b>	

**Table 17:** Use Case-Send OTP

<b>Use Case</b>	<b>3.3. Send OTP</b>
<b>Description</b>	The MoBA Server sends the One Time Password (OTP) to the user mobile
<b>Actors</b>	MoBA Server Mobile
<b>Assumptions</b>	1. The MoBA Server is connected to the mobile through some wireless technology like GSM or UMTS. 2. The user information is valid.
<b>Steps</b>	The MoBA sends the OTP to the user mobile.
<b>Variations</b>	If the user information is invalid
<b>Issues</b>	

**Table 18:** Use Case-Perform Authentication

<b>Use Case</b>	<b>4. Perform Authentication</b>
<b>Description</b>	The user can be authenticated by the valid OTP received through mobile and sending it to the MoBA Server.
<b>Actors</b>	MoBA Server
<b>Assumptions</b>	The user provides the valid password.
<b>Steps</b>	1. The user inputs the password received on mobile phone. 2. The MoBA server compares the OTP's and makes the decision.
<b>Variations</b>	The user enters invalid password.
<b>Issues</b>	

**Table 19:** Use Case-OTP SMS

Use Case	5. OTP SMS
<b>Description</b>	The mobile receives the OTP by SMS and it is forwarded to the client by user or through Bluetooth.
<b>Actors</b>	Mobile. Client. User.
<b>Assumptions</b>	In case of Bluetooth the connection should be established between the client and the mobile phone.
<b>Steps</b>	1. The mobile phone receives the OTP by SMS. 2. The OTP is forwarded to the client through Bluetooth or by user.
<b>Variations</b>	
<b>Issues</b>	

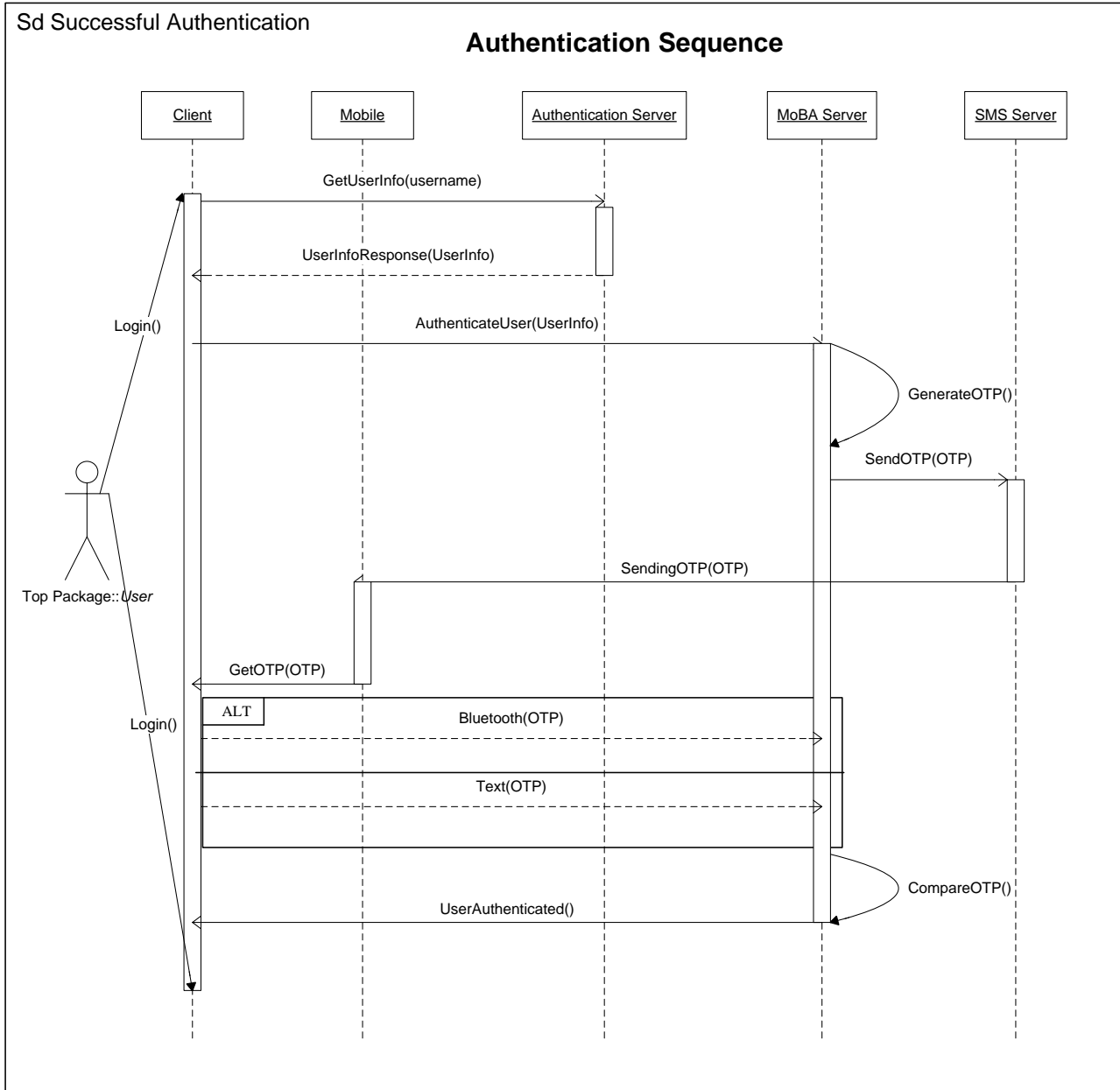
### 4.3 Interaction Diagrams:

The interaction diagram show how the different objects will interact with each other. The messages it will pass to each other and the information. One of the commonly used interaction diagram is sequence diagram which is very useful in showing the passing of messages. The sequence diagram of the successful authentication is shown in the below given diagram.

#### 4.3.1 Successful Authentication

The steps involved in the successful authentication are discussed as following:

- i. The User enters the username for login in the computer system (Client) in an enterprise.
- ii. The user information is forwarded to an authentication server to fetch the user information from the database in the Authentication Server
- iii. The user information is returned to the client, which is forwarded to the Mobile based Authentication server (MoBA). The MoBA server serves as Authenticator as well as the Authentication server.
- iv. The MoBA Server generates an OTP and forwards it to the SMS Server.
- v. The SMS server sends the OTP to the mobile through SMS message.
- vi. Upon receiving the OTP message at the mobile phone either the user can type the OTP or it can be forwarded to the client using Bluetooth technology.
- vii. The client forwards the OTP to the MoBA Server.
- viii. The MoBA server compares the OTP received from the client with the generated one.
- ix. If the OTP's match the user will be authenticated and will be given right to login in the system.



**Figure 32: Sequence Diagram- Authentication**

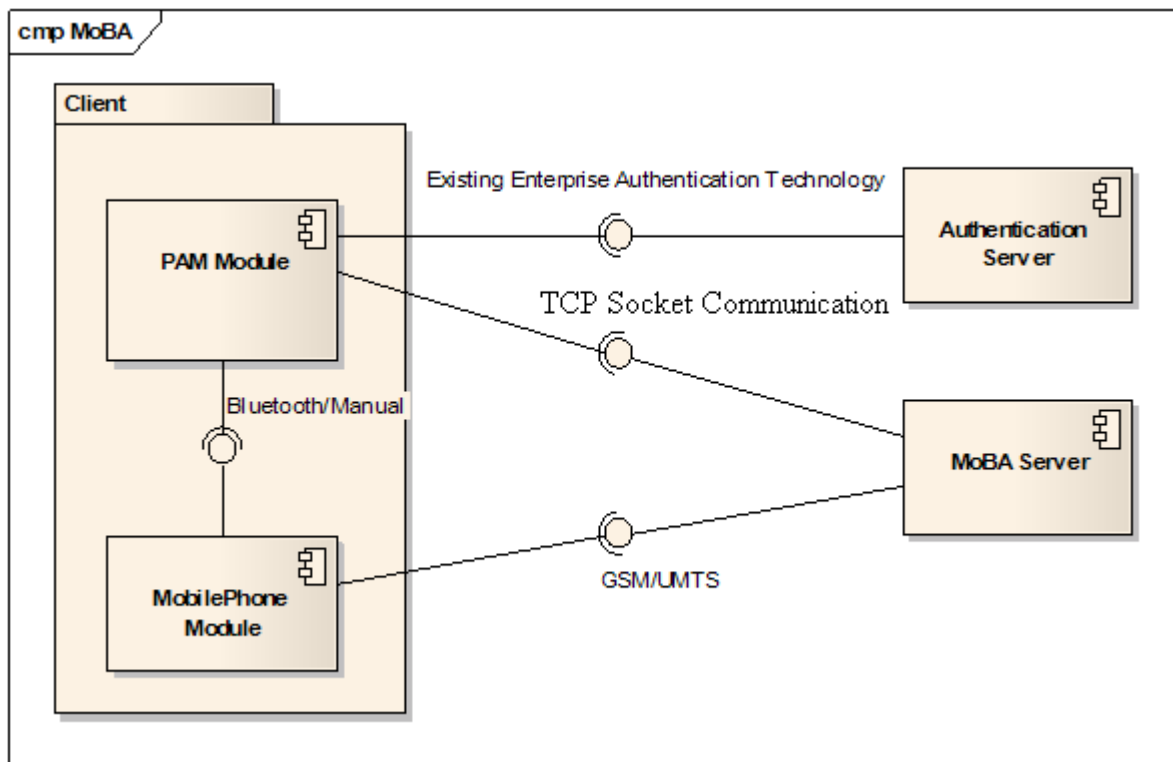
## 5. Design

The design chapter will identify the components, packages and classes of the system that the system will be composed of.

### 5.1 Components:

Components are modular parts of the system [85]. They are the individual pieces which are independent from each other. The behaviour of the components is defined by the provided and required interfaces.

The Figure 33 shows the main components of the system. The client package is added for showing the clarity in the Figure 33. There are four main components as can be seen in the given figure. The client side consists of the PAM module and the Mobile phone component, while the server side has been divided in to two portions. The Authentication Server has the user database and is controlling the existing technologies for authentication like LDAP, Kerberos etc. The other part Mobile Based Authentication (MoBA) Server handles the One Time password (OTP) and doing authentication on the basis of the OTP. The detail of each component is explained in this chapter.



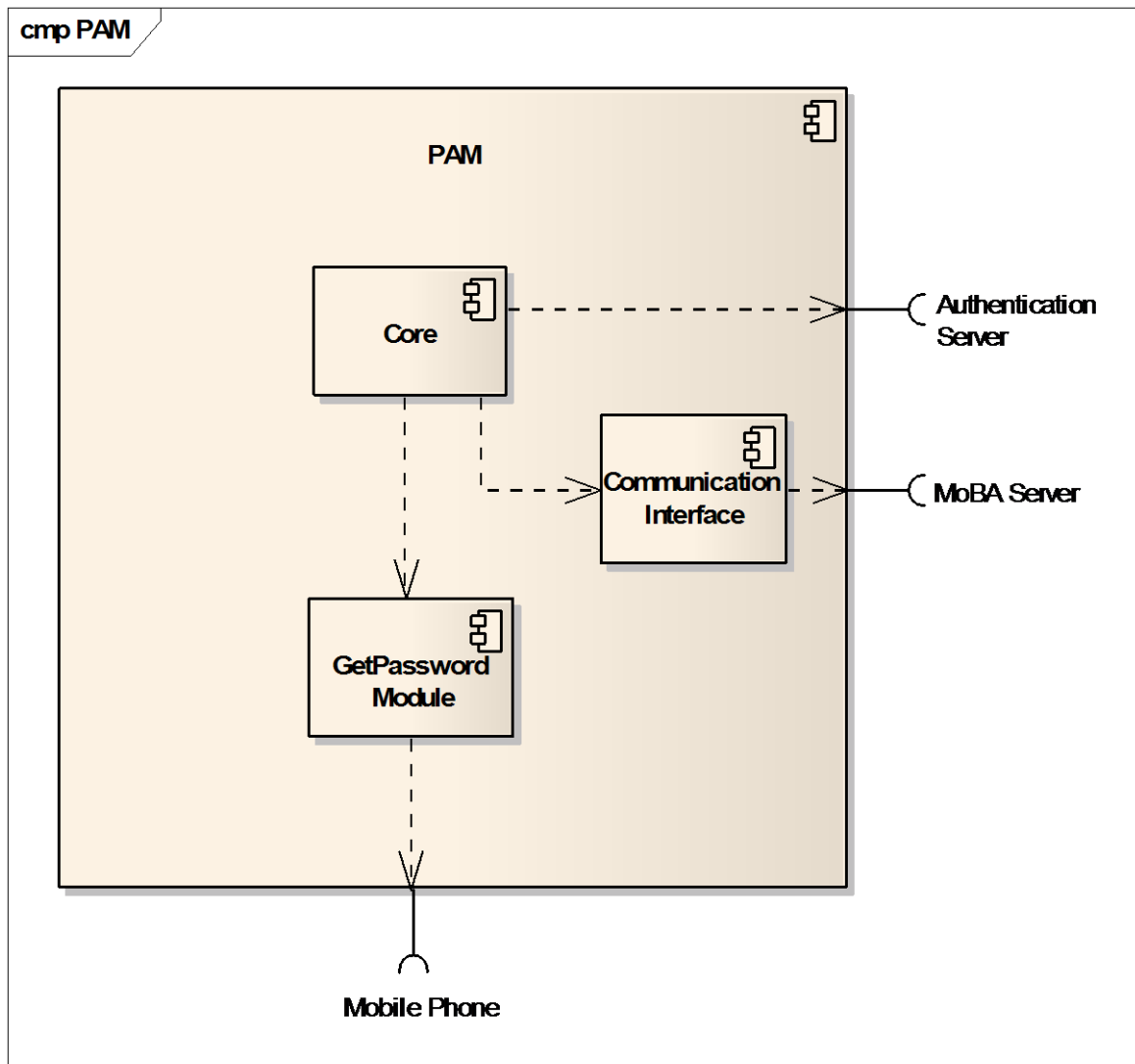
**Figure 33:** Main Component Diagram.

The PAM module at the client side communicates with the Authentication Server with the existing technology messages. It communicates with the Mobile Phone component with Bluetooth or manually as chosen by the user. The Mobile phone component has interface with MoBA Server for communication of OTP through SMS. The PAM module has also communication interface with MoBA Server for communication.

#### 5.1.1 PAM Module:

The PAM module handles the login service at the client side. It communicates with the existing

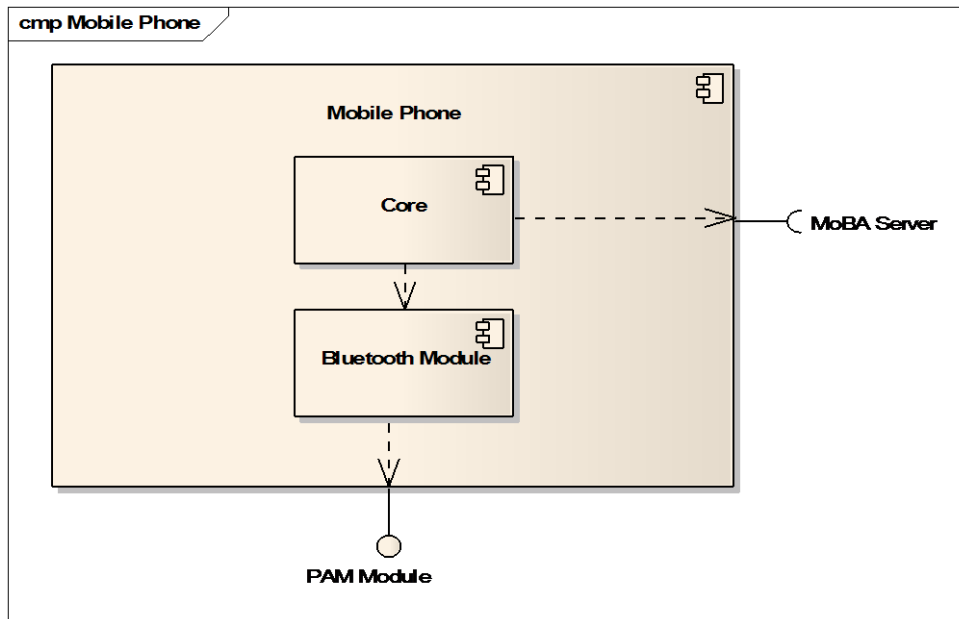
Authentication server to get the user information. For Authentication of user and for OTP it communicates with MoBA server. The PAM Module is divided in to 2 components; the GetPassword module accepts the password from user either through Bluetooth or manually depends on user choice. The communication interface controls the communication with the MoBA server for OTP and authentication.



**Figure 34: PAM Module Component Diagram.**

### 5.1.2 Mobile Phone Module:

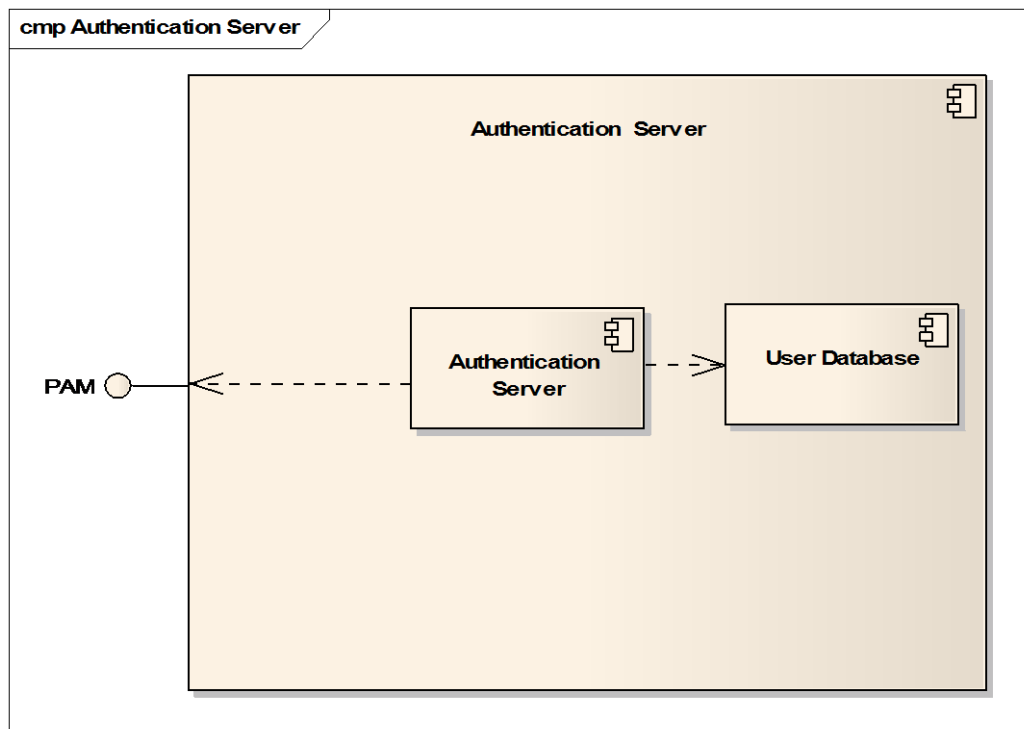
The main task of this component is to receive the OTP through SMS. The detail of the components can be seen in the Figure 35. It communicates with the MoBA server for OTP. It has Bluetooth Module which task is to forward the OTP through the Bluetooth to the PAM Module as it has the interface available for this task. After receiving the OTP it will be the user choice to forward the OTP either manually or through the Bluetooth.



**Figure 35:** Mobile Phone Component Diagram.

### 5.1.3 Authentication Server (AS):

The Authentication Server (AS) performs the authentication of users with the existing Authentication technology like LDAP, Kerberos etc. The detail of the Authentication server can be seen in Figure 36. It has user database which has all the information required for the authentication. It has an interface with the PAM module for communication.

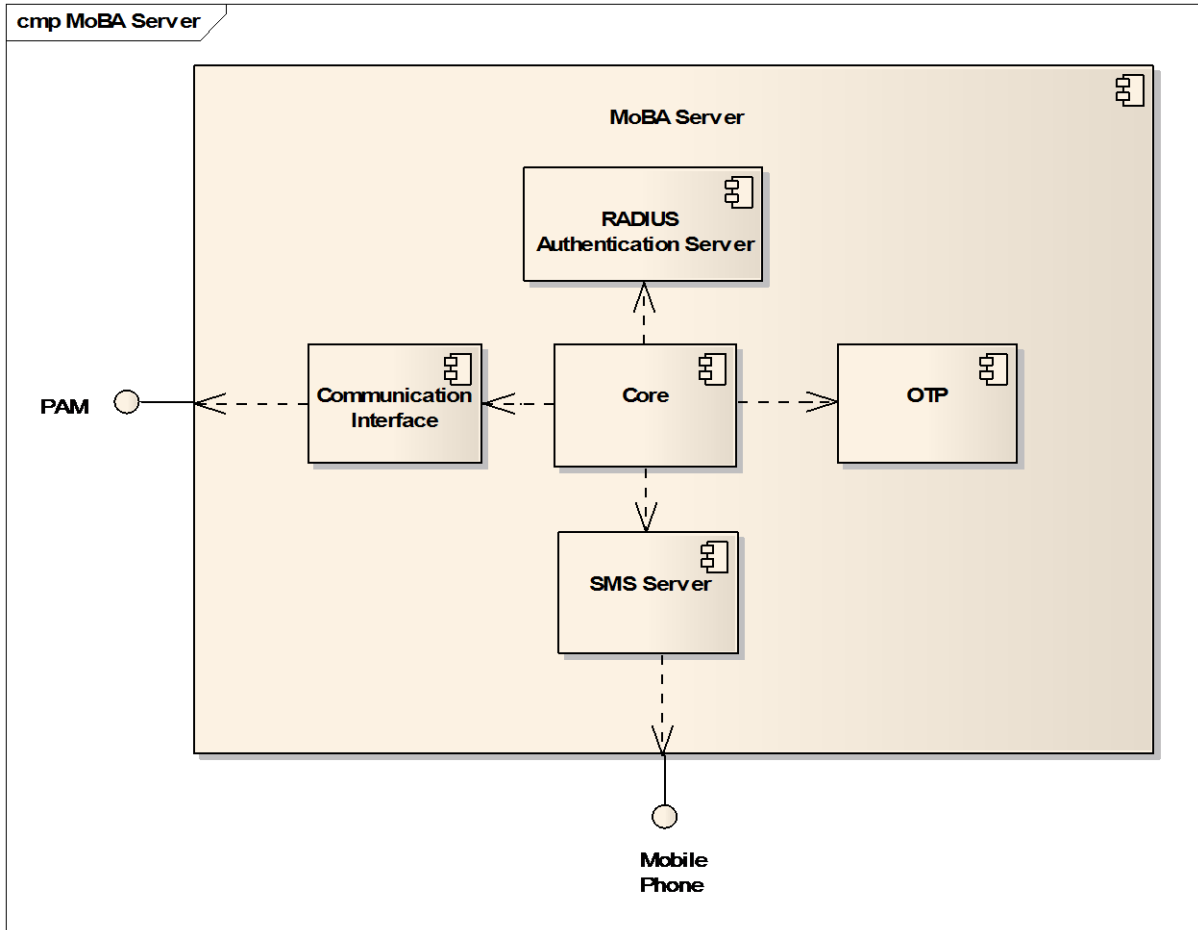


**Figure 36:** Authentication Server Component Diagram.

### 5.1.4 MoBA Server:

This is the main component of the system. It handles the generation of OTP and authentication of

user based on that OTP. There are four main components which can be seen in the figure given below. The communication interface handles the communication with the PAM module. The OTP handles the generation of the OTP. The RADIUS Authentication Server handles the Authentication of user based on the OTP. The SMS server handles the communication with the SMS server and SMS communication with the Mobile Phone module as it has an interface attached with it.



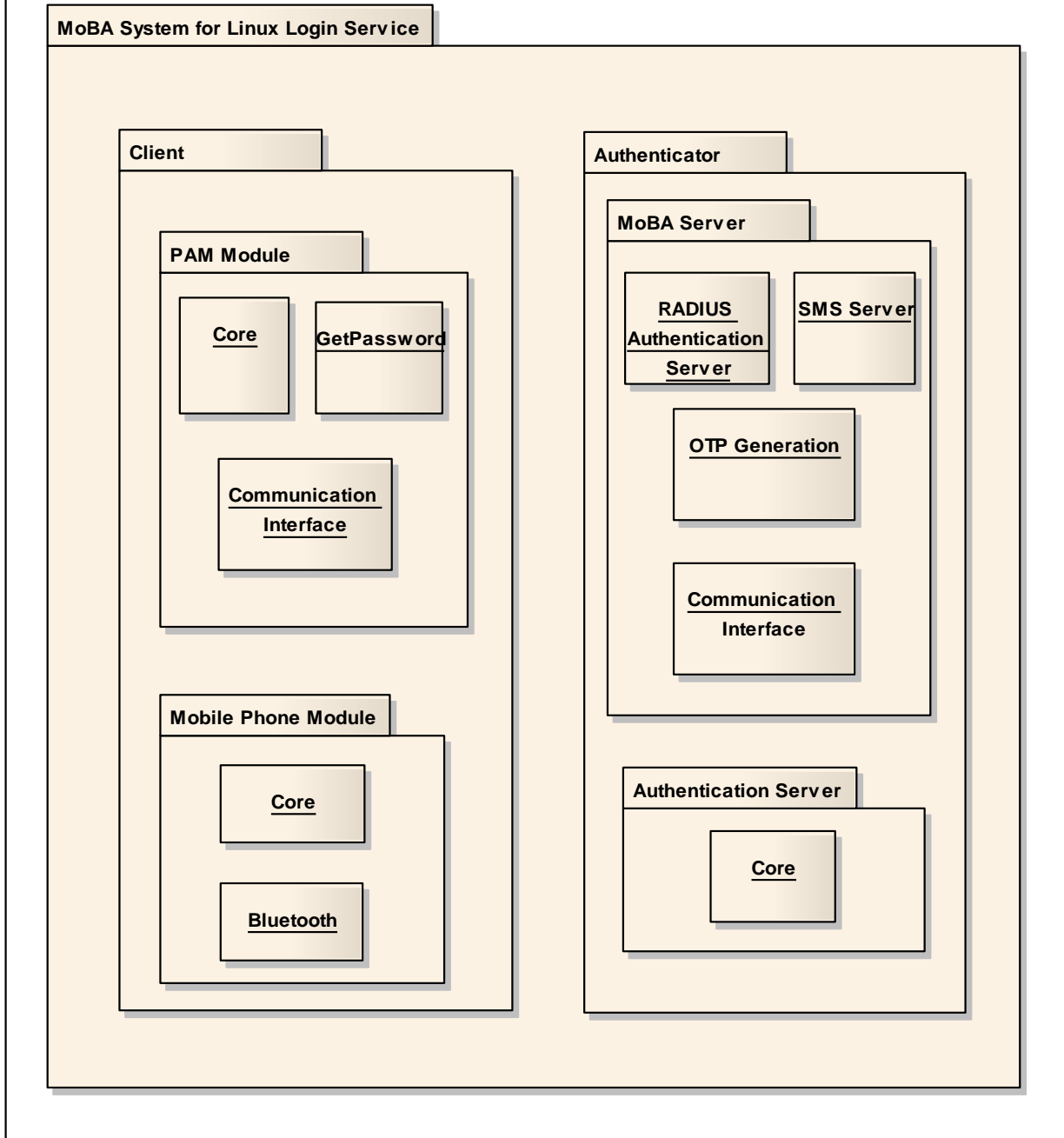
**Figure 37: MoBA Server Component Diagram.**

## 5.2 Package and Class Diagrams

The class diagrams show the different objects in the system and their relationship. It also shows the attributes, operations and the constraints with the objects connections [86].

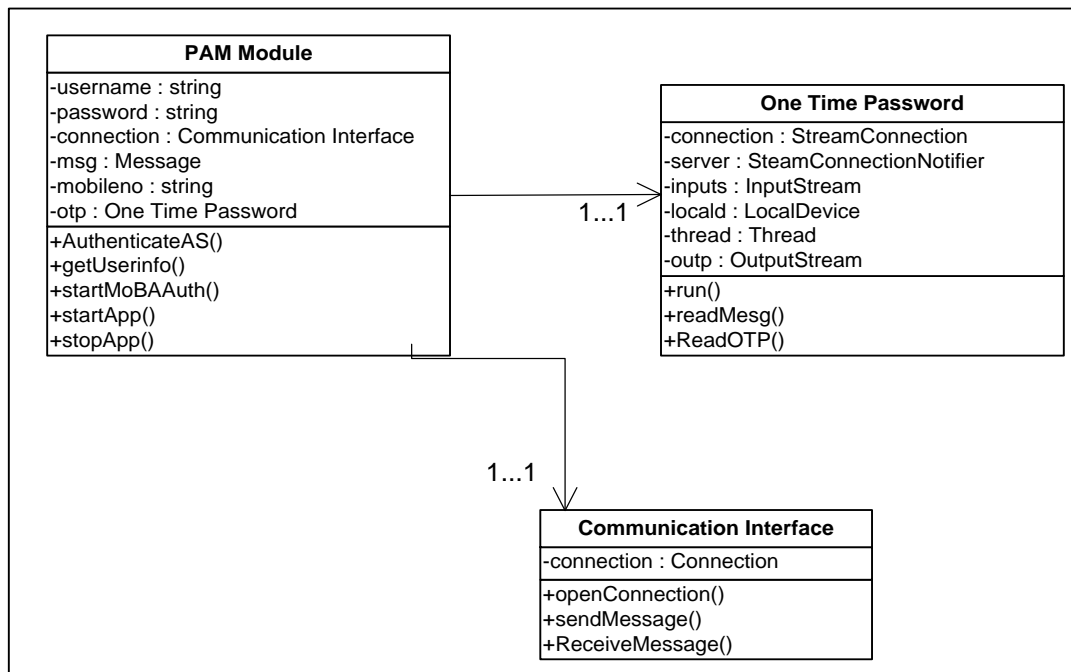
Package diagrams are used to see the overview of the system as the classes from the same part are grouped together to form a simple view of the system. The figure given below shows the package diagram of the system.





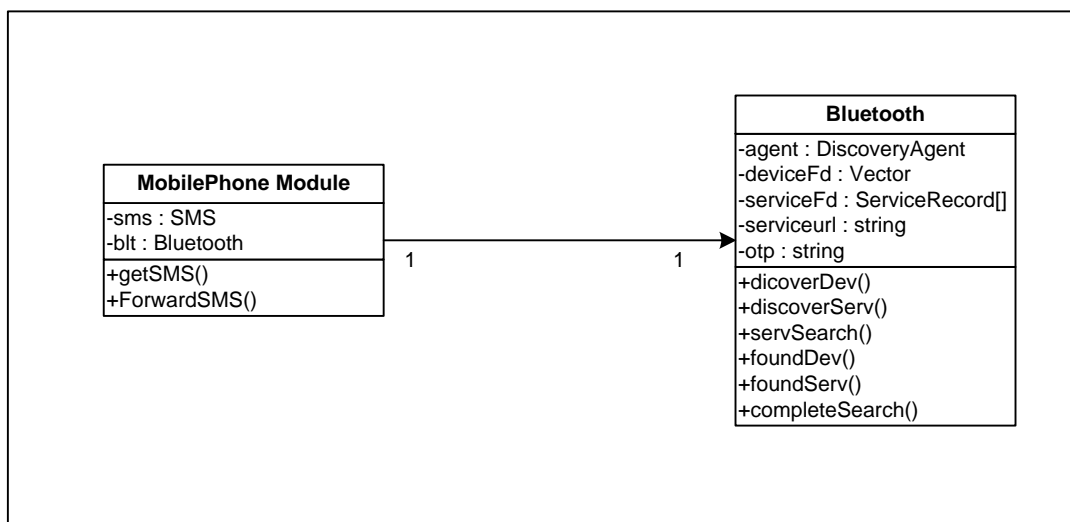
**Figure 38:** MoBA System for Linux Login Service Package Diagram.

The diagram given below show the PAM Module class diagram. The PAM module handles the user login and the functionality is implemented by PAM Module class. The One Time Password class implements the functionality that it will extract the OTP received by the user through SMS to forward it to PAM Module either manually or by bluetooth. The communication interface class implements the socket communication with the MoBA server for authentication. It controls all the communication with the MoBA Server.



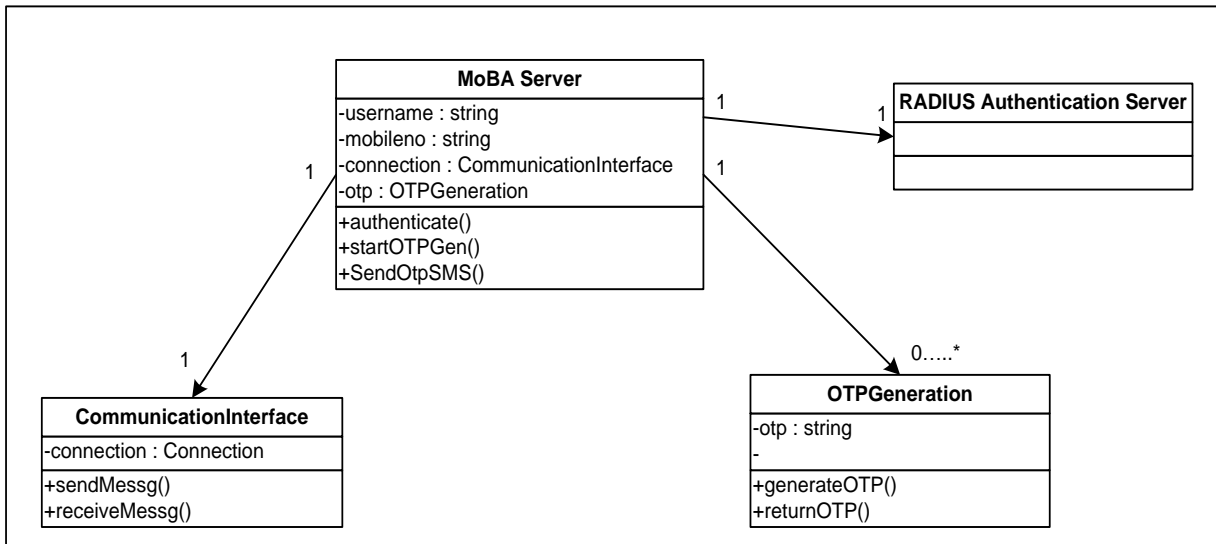
**Figure 39:** PAM Module Class Diagram.

The following diagram shows the class diagram of Mobile Phone Module. The Mobile phone module implements the functionality of receiving the OTP as SMS and then depends on the user choice to forward the OTP through Bluetooth or manually. The Bluetooth class implements the functionality of forwarding the OTP as SMS to the PAM module.



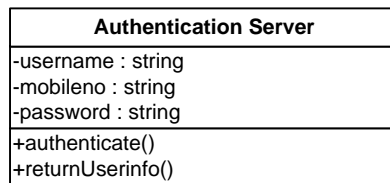
**Figure 40:** Mobile Phone Module Class Diagram.

The following diagram implements the functionality of the Mobile Based Authentication (MoBA) server. The MoBA Server class authenticates the users based on the OTP generated. It also handles the functionality of sending the OTP as SMS to the user so that he can authenticate himself by using it. The OTPGeneration class implements the functionality of generating a new OTP for every session. The CommunicationInterface class implements the functionality of handling the communication with the clients for authentication.



**Figure 41:** MoBA Server Class Diagram.

The following diagram shows the Authentication Server class diagram. Its functionality is to authenticate the users on the bases of stored static passwords. They also store the data of the user like name and their mobile numbers which can be used for the mobile based authentication for Linux workstation logon process in second stage.



**Figure 42:** Authentication Server Class Diagram.

## 6. Implementation

This chapter describes the implementation of the concepts of the proposed solution. It includes the deployment diagram of the system and the details of the components implementation are also discussed.

### 6.1 Deployment:

The deployment diagram shows that how the different software pieces of the system will run on different hardware and their relationship. Figure 43 shows the deployment diagram for the secure mobile authentication for Linux workstation logon process.

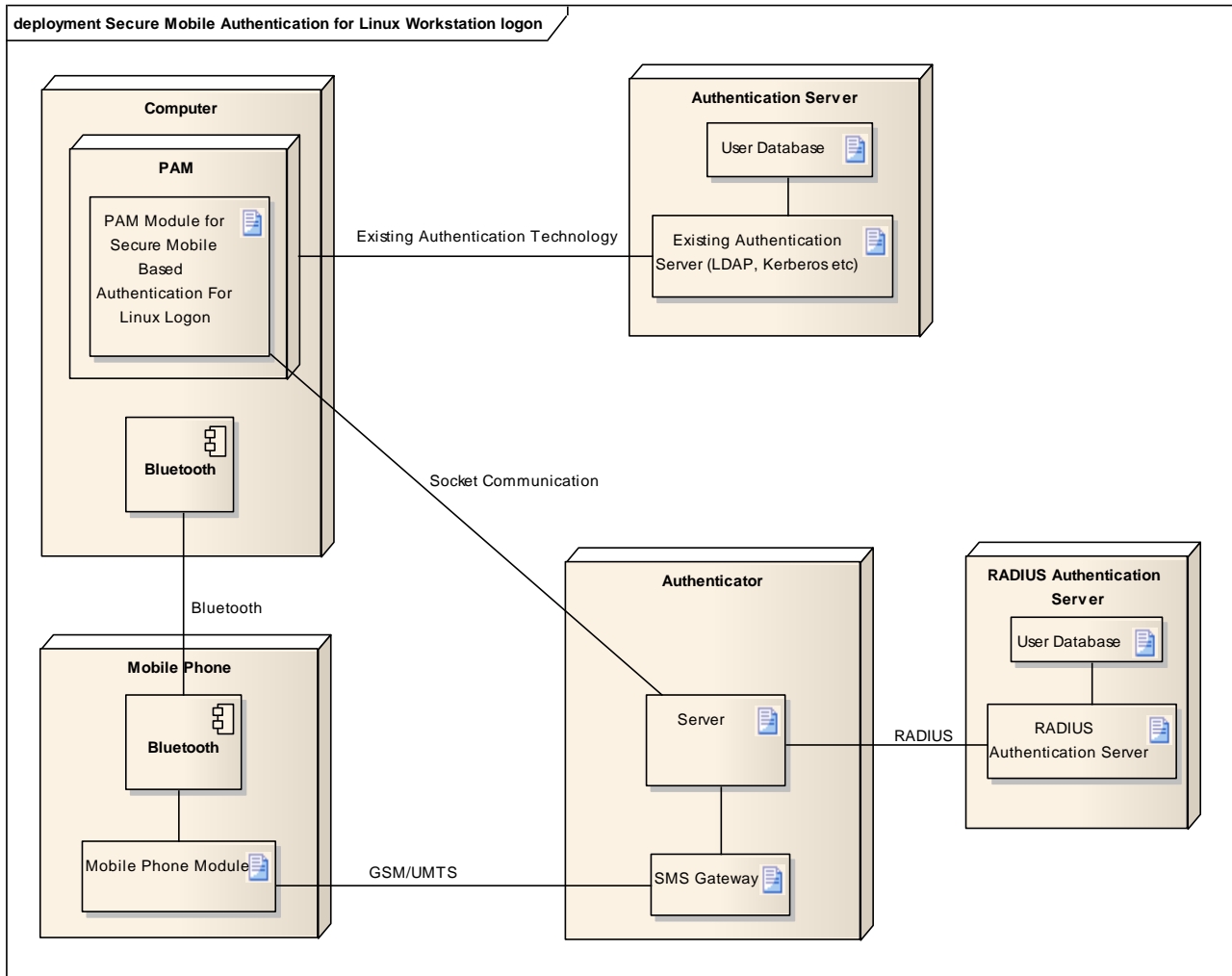


Figure 43: Secure MoBA for Linux Workstation Logon Deployment Diagram

### 6.2 Implementation of the components of the solution

The implementation of the components is as following:

#### 6.2.1 PAM Module

The PAM module has been installed on the client as PAM is used for authentication purpose in Linux environment. The PAM module has been implemented using C language. PAM makes the user free from the technology specific. It works as an intermediate API between the PAM functions and the underlying technology.

A PAM module has been implemented that calls the required functions for authentication and accessing data for the mobile based authentication from the existing enterprise authentication technology like LDAP, Kerberos etc. In section 2.5 it is explained how to write a PAM module. A sample code for writing a PAM module has been given in Appendix A as well.

The data of user is accessed with the *passwd* structure which has the following data members.

```
struct passwd {  
    char *pw_name; /* user name */  
    char *pw_passwd; /* user password */  
    uid_t pw_uid; /* user id */  
    gid_t pw_gid; /* group id */  
    char *pw_gecos; /* real name */  
    char *pw_dir; /* home directory */  
    char *pw_shell; /* shell program */  
};
```

The *pw\_gecos* data member is used for the mobile number of the user. The field is used as

*USER REAL NAME, MOBILE NUMBER*

E.g. george, 004746252374.

If the PAM module finds the mobile number it continues with the mobile based authentication as well otherwise it authenticates the user on the basis of the existing technology authentication method.

The other task that PAM Module performs is to interact with the Authenticator for the mobile based authentication. It used TCP socket communication for the communicating with the Authenticator. The PAM module also interacts with the user for getting the username and password. For mobile based Authentication it can accept the OTP either manually or by Bluetooth.

### **6.2.2 Authentication Server**

To make the solution work with the existing technology like LDAP, Kerberos etc, the existing enterprise authentication server has been used that will be used for user data like user name, mobile number, the static password and can be used for the static password authentication as now a day's used for authentication in an enterprise. This will help in security as one authentication of user will be done on the static password and other on the basis of mobile based authentication. The PAM API helps us to be free of any specific technology and there are functions available for the required functionalities.

### **6.2.3 Authenticator**

The Authenticator is implemented in C language. A multi process server has been implemented in C language to handle multiple users at a time. It works as a server and a connection point for PAM module and the RADIUS Server. It communicates with the PAM Module using TCP sockets. The other task that Authenticator performs is to generate a one time password (OTP). It generates OTP sends it to the SMS gateway so that OTP can be sent to the user as SMS. At the same time it saves the OTP in RADIUS server which can be used for authentication purpose. After the user receives the OTP and sends it back to the Authenticator the process waits. Upon receiving the OTP the Authenticator sends it to the RADIUS Server for authentication of user. If the user is authenticated by the RADIUS Server then user is allowed for login otherwise rejected and he has to do the authentication process from start.

#### **6.2.4 One Time Password (OTP)**

The OTP is used once. There are two methods implemented for the generation of the OTP.

In the first method SHA-384 is used. The following data is given as input to the algorithm

USERNAME | RANDOM NUMBER | TIME

The time part consists of day, month, year, hour, minute, second. It generates a hash of 48 bytes which is converted to hexadecimal values that can be human readable. When we change the hash value in hexadecimal values, we have then 96 hexadecimal characters.

The second method that we have implemented is using *dev/random* package of Linux for generating a random number which can be used as OTP. This package generates a random number based on the environment noise received through device drivers and other things [88].

We have used the second option as it gives us the freedom of selecting the length of OTP and it uses different characters like numbers, small and capital letters, special characters. We have used an OTP of 13 characters long, as it will help if the user has to enter the OTP manually.

#### **6.2.5 RADIUS Authentication Server**

The RADIUS Authentication Server is used for mobile based authentication of the user. It saves the OTP generated and do the authentication process.

#### **6.2.6 Mobile Phone Module**

A MIDlet is required to enable a mobile to send and receive the message through Bluetooth. We can configure the mobile so that when it receives the message from a certain number it will send it to required place using Bluetooth technology by using the PUSH to Register Service.

A MIDlet consists of a Java Archive (JAR) file containing the source code and a java application descriptor (JAD) file describing the MIDlet properties.

#### **6.2.7 SMS Gateway**

A web service is provided for SMS gateway. We have accessed the SMS gateway using the web service which has been provided.

### **6.3 Testing the Prototype**

The implementation has been tested to check the functionality of the system against the functional requirement that have been discussed in section 4.1.1.

#### **Environment of Testing**

There are 2 desktop, 1 laptop used for the testing purpose. Ubuntu 9.10 is installed on 2 computers and Ubuntu 8.10 is installed on the third computer. One computer is used as the server and the 2 are used as client who depends on the server for their authentication. Open LDAP has been installed so to have the Authentication Server. The PAM modules have been installed on client side and the server code of Authenticator is being run on the server side with RADIUS Server. Nokia 6230i has been used for the testing purpose.

#### **Testing Scenarios**

There are 2 users added to LDAP server for testing. The mobile number has been added to the *gecos* field of the database as discussed in section 6.2.1. In the given Figure 44 the users in the LDAP server have been displayed. The 2 users are George and Babar.

```

Applications Places System
usman@usman-desktop: ~
File Edit View Terminal Tabs Help
usman@usman-desktop:~$ ldapsearch -xLLL -b 'ou=people,dc=example,dc=com' cn
dn: ou=people,dc=example,dc=com

dn: uid=george,ou=people,dc=example,dc=com
cn: george

dn: uid=babar,ou=people,dc=example,dc=com
cn: babar

usman@usman-desktop:~$ █

```

**Figure 44:** Users in the LDAP Authentication Server

In the figure 45 the data of Babar user have been displayed. The gecos field is changed according to the requirement.

```

Applications Places System
usman@usman-desktop: ~
File Edit View Terminal Tabs Help
usman@usman-desktop:~$ sudo ldapfinger babar
dn: uid=babar,ou=people,dc=example,dc=com
objectClass: account
objectClass: posixAccount
cn: babar
uid: babar
uidNumber: 10002
gidNumber: 10000
homeDirectory: /home/babar
loginShell: /bin/sh
description: User account
userPassword:: e1NTSEF9SnLXS1hYRHRKYUI2SFpmWlhKSlRGsXFjTXhNQ25FVVg=
gecos: babar,4746252374

usman@usman-desktop:~$ █

```

**Figure 45:** User Information in LDAP Server

The server code at the Authenticator has been started so that it should be in waiting state for the client to be authenticated. Figure 46 shows the wait state.

```

Applications Places System
usman@usman-desktop: ~/Servercode
File Edit View Terminal Tabs Help
usman@usman-desktop:~$ cd Servercode
usman@usman-desktop:~/Servercode$ ./server
server: waiting for connections...
_

```

**Figure 46:** Server Waiting for Clients.

At the client side the client code has been executed. It asks for the username and the static password for the LDAP server authentication. After authentication of user with the static password, the data is send to the server for mobile based authentication. If user is not authenticated or mobile number is not found than the mobile based authentication is not done. Figure 47 shows the process at the client side

```
Applications Places System usman@usman-desktop: ~/Project
File Edit View Terminal Help
usman@usman-desktop:~$ cd Project
usman@usman-desktop:~/Project$ ./pam_moba_login
login:george
Password:
Authentication Successful      retval=0
IP=129.241.208.66
PORT=3490
USERNAME=george
Mobile#=4746252374
client: connecting to 129.241.208.66
MOBA Password:█
```

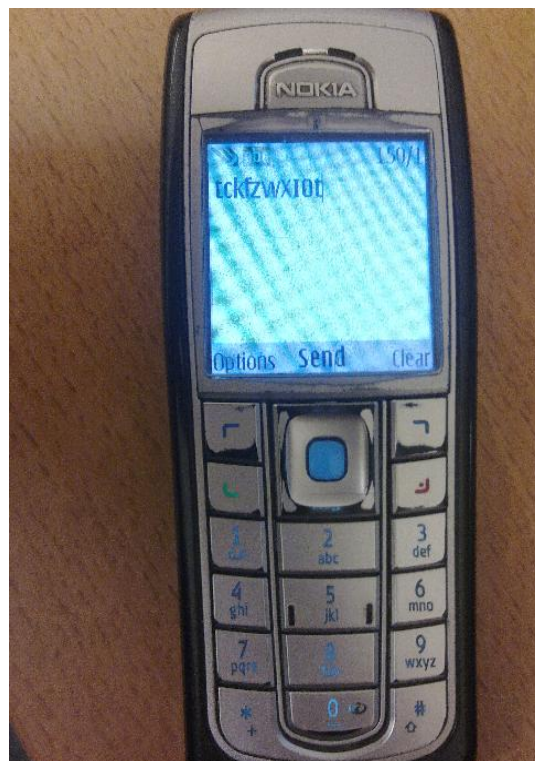
**Figure 47:** LDAP Authentication

With this the data is sent to the Authenticator for mobile based authentication. At the Authenticator the OTP is generated. At the same time the OTP is sent to the user through SMS. The following figure 48 shows the process.

```
Applications Places System usman@usman-desktop: ~/Servercode
File Edit View Terminal Tabs Help
usman@usman-desktop:~$ cd Servercode
usman@usman-desktop:~/Servercode$ ./server
server: waiting for connections...
server got connection from ::ffff:129.241.208.219:3490
Mobile#=4746252374
User Name Connected to Server: george
OTP=tckfzwxI0t
█
```

**Figure 48:** OTP Generation Snapshot

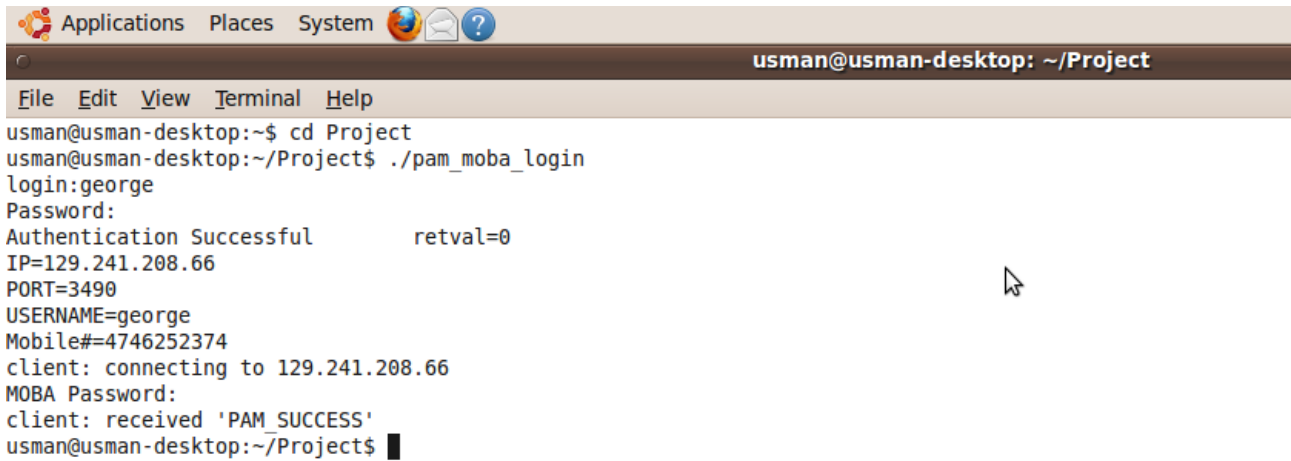
The OTP is received at the user mobile number as can be seen in the figure 49 below.



**Figure 49:** OTP received at User Mobile



The OTP received at the user mobile is typed by the user at the client console waiting for the MoBA password. If the password typed is correct the user will be authenticated otherwise the process will start again. The figure 50 given below shows the process



```
Applications Places System [Globe] [Envelope] [Question Mark]
usman@usman-desktop: ~/Project
File Edit View Terminal Help
usman@usman-desktop:~$ cd Project
usman@usman-desktop:~/Project$ ./pam_moba_login
login:george
Password:
Authentication Successful          retval=0
IP=129.241.208.66
PORT=3490
USERNAME=george
Mobile#=4746252374
client: connecting to 129.241.208.66
MOBA Password:
client: received 'PAM_SUCCESS'
usman@usman-desktop:~/Project$ █
```

**Figure 50:** User Authentication Snapshot.

## 7. Validation and Evaluation

In this chapter the functionality of the proposed solution is validated against the functional requirements. The security evaluation of the solution is also done which defines the different types of attacks and the countermeasures for the attacks.

### 7.1 Validating the Functionality against Requirements

The functional requirements of the system are

#### Secure Mobile Based Authentication

The solution provides secure mobile based authentication of users for Linux workstation logon in an enterprise. As can be seen in the proposed solution in section 3.3, the PAM module at client communicates with the Authenticator for the mobile based authentication. And the Authenticator then communicates with the RADIUS Authentication server and user through his mobile phone using SMS.

#### Mutual Authentication

The client identity can be validated as the clients are in an enterprise environment where administrator has the knowledge of the network. The server validity can be confirmed by the validation of the OTP generated by the Authenticator received at the user mobile. Thus this will confirm the mutual authentication.

#### Two Factor Authentication

The user is authenticated on the basis of two factors. One is the static password that the user selects and the other factor is the mobile based authentication using a 13 character long OTP. Thus the two factor authentication will make the solution more secure

#### Working with the Existing technology

The solution will work with the existing static password systems. The solution uses PAM module at the client side which allows us free of the existing authentication server technology for enterprises like LDAP, Kerberos etc. Thus we can use the solution with the current technologies without changes.

#### Strong Password (OTP) For Mobile Based Authentication

The OTP is generated with the use of *dev/random* package which generates random numbers (OTP) using the noises from the environment obtained through the device drivers and other things. The password length is 13 characters consisting of letters, numbers and special characters, thus making it a strong password. This solution will be suitable for those cases where most of the users don't have the facility of using the Bluetooth feature of the solution and has to enter password manually.

The other solution for OTP that has been implemented is using SHA-384 which produces a 48 byte hash which is converted in 96 hexadecimal characters. This solution of OTP can be used when the all the users have the facility of using the Bluetooth feature.

#### Usability

Besides creating a secure mobile based authentication scheme the important issue is usability. As by using the mobile based authentication method no extra hardware is required on the client side and as most people always carry their mobile phone with them, thus making the solution easy for users to use.

## Cost and Deployment

As the solution will use the already functional mobile and SIM with the user which will dramatically reduce the deployment time and which will also has great impact on cost as well. The deployment will be easy as the solution will work with the existing architecture making the solution less costly. A new user in this solution will be authenticated almost immediately as the administrator adds the record of the user.

## Reliability, Availability and Scalability

If due to some problem, the mobile based authentication system is down then the user can be authenticated with the existing static password based authentication method with making the required changes by the administrator. Thus, makes the availability and reliability of the solution much higher. On the other hand the GSM system is considered to be very reliable and it is the most deployed mobile network in the world. The other important factor is that the system must scale well with the increase in the number of users. The proposed solution can be easily distributed both physically and geographically to avoid the possible bottlenecks during the authentication of users.

## 7.2 Different Types of Attack

There are different types of attacks which are important to be considered for the security of the solution. Following are some attacks

**Eavesdropping:** In this type of attack, the attacker will monitor the transmission between the client and authenticator for getting the sensitive information.

**Dictionary attack:** This is a kind of brute force attack where the attacker tries to find the password from the list of passwords called dictionary. As most of the time the password that people use are single words or common things that can be found in the dictionary.

**Masquerade:** In this type of attack the attacker pretends to be someone else. E.g. the attacker may masquerade himself as client and authenticator as well. The main goal is to get some sensitive information from the user.

**Replay attack:** In Replay attack the attacker reuses the old authentication sequence to have an unauthorized access to the system.

**Modification:** In this type of attack the attacker make some changes in the message or the contents of a message to generate an unauthorized effect

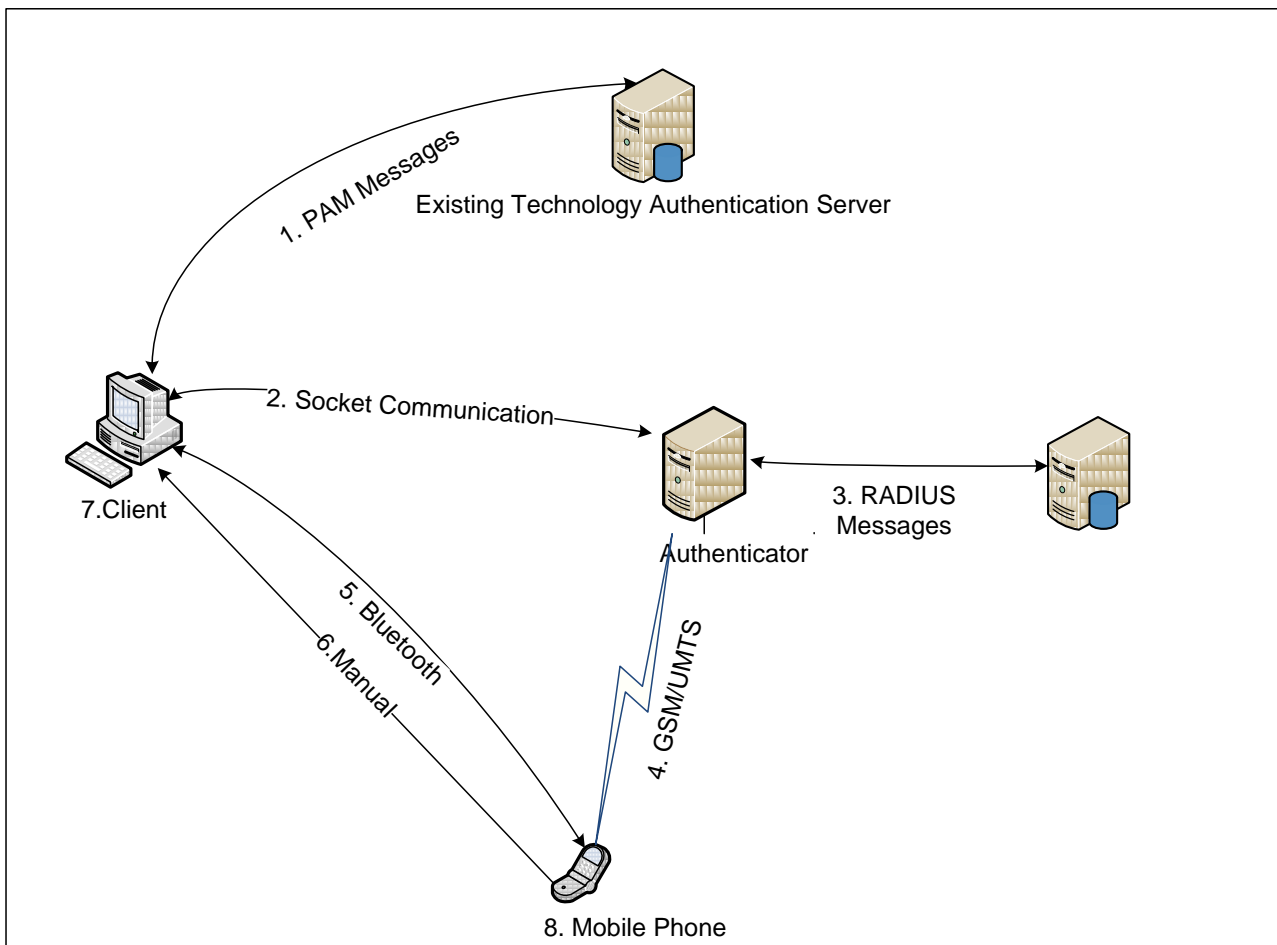
**Session hijacking:** In Session hijacking attack the attacker takes over a valid session that is already established to get an unauthorized access to the system.

**Phishing:** In this type of attack the attacker uses social engineering techniques to get the user secret information.

**Man-in-the-middle (MITM) attack:** In this kind of attack the attacker can read and modify all messages that are going between the client and authenticator without the knowledge of the user.

### 7.3 Security Evaluation of the Solution

The following diagram shows the components of the security environment of the solution.



**Figure 51: Component Threat Model**

#### 1. PAM Messages

The PAM messages are secured by the existing server technology. So this component will be difficult for the attacker to attack as the system already provides the security.

#### 2. Socket Communication

This component is very important part of the solution to be protected as the user information is communicated through this channel. The security can be provided by using Transport layer security (TLS), so the communication channel will be secured from modification and the session hijacking attack. Secondly as the proposed solution requires 2 factors, one is the static password which is authenticated with a secure channel and the second is the OTP which is communicated through the mobile. If the OTP has been somehow released to the attacker he will still need the static password of the user and the OTP can only be used for once, making the task difficult for the attacker.

#### 3. RADIUS Messages

The RADIUS messages contain the security as the RADIUS technology already have.

#### **4. GSM/UMTS**

As UMTS has the security mechanisms available as it provides mutual authentication, so we can depend on it. For GSM, it is vulnerable to the man in the middle attack (MITM) as the network is not authenticated which is weak link. This problem is solved by having the static password authentication. As if the attacker gets the OTP he will still need the static password for the authentication. Secondly the OTP is used for once so he cannot misuse it.

#### **5. Bluetooth**

Bluetooth is another weak link which can be used by an attacker. To secure link we can use the encryption so that the data can be made secure. We can use the pass key made during the pairing for encryption as discussed in section 2.9. And the static password is the other data that the attacker has to find out to use the OTP.

#### **6. Manual typing of OTP**

When the user is typing the OTP there may be a chance that he writes it or release the OTP. If even the attacker finds the OTP, he will still require the static password for authentication. And the OTP is useful for authentication once, thus makes the OTP useless for the attacker.

#### **7. Client Computer**

The client computer can be attacked by the attacker for getting some useful information. As the client computer is in the enterprise network where the environment can be controlled by the network administrator to not allow such malware like Trojan horse or worms which can cause leakage of information. Secondly the OTP used once is useless for the second time.

#### **8. Mobile Phone**

Mobile phone is another important component for the security of the system. If the attacker steals the mobile with the SIM of the user, it will still be useless for the attacker as he will require the static password of the user as well.

#### **9. One Time Password**

It is necessary to keep the OTP strong enough so that the attacker cannot guess it or cannot find it with dictionary attack. As in our solution we have used 13 random characters consisting of numbers, letters and special characters. The NIST considers a password of 80 bits entropy to be strong enough against the brute force attack. As we have used 13 random characters so its entropy is equal to 85 bits thus making it strong enough against the brute force attack. The OTP is generated by using the *dev/random* package that generates the random numbers from the environment noise obtained from the device drivers and other things [88], thus creating random numbers which will make the guessing of OTP difficult.

Secondly the attacker needs the static password as well for authentication, so the attacker will have many things to coup with.

## 8. Conclusion

In order to come up with a proposed solution for secure mobile based authentication system for Linux workstation logon, a study of different technologies that are used in mobile phone based authentication schemes is done. The overview of identity management has been discussed. A study of smart card security, smart card operating systems, identity management UMTS networks and how security is provided in UMTS networks has been carried out.

An overview of the different enterprise identity management solutions used currently is made. The RADIUS technology has been explored to get the idea of an enterprise authentication server. A detailed study of Bluetooth security is done. An analysis of different available mobile phone based authentication schemes is done to get some sense how the mobile based systems work. Then the mechanism of authentication and authorization in Windows, Linux, and MAC OS X has been analyzed for adding the required functionality to the logon process. In Linux operating system the Pluggable authentication module technology provides extendibility by allowing different applications to have different modules for authentication. The logon process in Linux is treated as the same way as any other process, has its own PAM configuration file and can use any appropriate module.

While designing a solution it is required to make the Linux workstation logon more secure, user friendly and can work with the existing technologies. At the client (user) side a Pluggable Authentication module (PAM) module has been implemented which makes us free from being technology specific. By using the PAM API and the functions we can access the user information and authenticate user at the server without taking care of the technology at the server side like LDAP, Kerberos etc. The PAM API converts the required messages to the technology, making us free from technology. An Authenticator has been implemented which communicates with the PAM module using socket communication. The Authenticator serves as the connection point for the client and the RADIUS Authentication server. The Authenticator is responsible for generating OTP of 13 characters and communicating with the SMS gateway for sending the OTP as SMS to the user.

When the user will receive the OTP as SMS, he can insert it in the client computer as manually or by using Bluetooth technology. By using the option of manually entering the password, the user who does not have mobile phone with Bluetooth facility or the mobile of the user uses software which our solution does not support will still be able to use the solution. The mobile authentication is done by the RADIUS Authentication Server through the Authenticator.

The Solution uses two factors for authentication, one is the static password and the second factor is the mobile based authentication using a 13 character long OTP. With adding the existing technology and giving the administrator the option of selecting the authentication method for user makes the solution more suitable for an enterprise. The administrator can select either the authentication based on static password or both static password and mobile based authentication method. By adding this facility we add more reliability and availability to the system as if there is some problem with the mobile based authentication the users can still be authenticated with the existing static password technology.

The detailed analysis and design of the proposed solution has been carried out in this thesis so that to come up with an implementation of the different parts of the solution. The validation and the security evaluation is also discussed as to check the different security threats and how to tackle with the different security attacks

The solution which is proposed for the mobile based authentication for Linux workstation logon process will make the logon process more secure. One of the big advantages of the proposed

solution is that it uses the infrastructure that is already available which will make the cost less and the deployment very easy. Thus makes the solution more suitable.

## **Future Work**

In this thesis the prototype implemented shows the main concept of the solution and has been tested with very few users and exhausting testing has not been carried out. For commercial use it is required to have fully developed implementation that should be tested with all the security features and real time environment to get a secure and reliable system.

New techniques should be explored to find a common mobile based authentication architecture for multiple platforms like windows, Mac OS, Solaris and Linux etc in an enterprise environment.

As in case of using Bluetooth for automatic transfer of data from mobile to computer ,different mobile phone companies have different operating system technologies like symbian, Andriod, iPhone OS, palm OS etc. The method and support for accessing the SMS in all technologies is different. Therefore it is required to have a solution which can access the SMS without being technology specific. This will allow the enterprise that will use this solution to use the automatic version without taking care of the user mobile technology. And companies have to bear the cost of mobile phone for its users and that is one of the reason that we have added the manual version of entering the password.

It is required to explore a secure method of using the secrets or identities in the Mobile Network operator for the mobile based authentication of users.

## References

- [1] Do Van Thanh, Ivar Jorstad, “The Ambiguity of Identity”, teletronikk 3/4.07, 2007, available at: [http://www.telenor.com/teletronikk/volumes/pdf/3\\_4.2007/Page\\_003-010.pdf](http://www.telenor.com/teletronikk/volumes/pdf/3_4.2007/Page_003-010.pdf)
- [2] Do Van Thuan, “Identity Management Demystified”, teletronikk 3/4.07, 2007, available at: <http://www.telenor.com/teletronikk/volumes/index.php?page=ing&id1=73&id2=200&id3=976&select=05-09>
- [3] I. Jorstad, Do Van Thanh, “The Mobile Phone as Authentication Token”, teletronikk 3/4.07, 2007.
- [4] “Offering SIM Strong Authentication to Internet Services”, available at: [http://www.ongx.org/SIM\\_STRONG\\_WHITEPAPER2.3\\_SMS.Screen.pdf](http://www.ongx.org/SIM_STRONG_WHITEPAPER2.3_SMS.Screen.pdf)
- [5] S. Hallsteinsen, I. Jorstad, Do Van Thanh, “Using the mobile phone as a security token for unified authentication”, International Conference on Signal Processing, Communication and Networking (ICSCN), August 2007
- [6] “Accenture Leadership in Customer Service: Building the Trust”, 2006, available at: [http://www.accenture.com/Global/Services/By\\_Industry/Government\\_and\\_Public\\_Service/PS\\_Global/R\\_and\\_I/BuildingtheTrustES.htm](http://www.accenture.com/Global/Services/By_Industry/Government_and_Public_Service/PS_Global/R_and_I/BuildingtheTrustES.htm)
- [7] A. Miriam B. Lips, J. A. Taylor, J. Organ, “IDENTITY MANAGEMENT, ADMINISTRATIVE SORTING AND CITIZENSHIP IN NEW MODES OF GOVERNMENT”, Information, Communication & Society, 12:5,715 — 734, 2009, available at: <http://dx.doi.org/10.1080/13691180802549508>
- [8] M. Crompton, “Proof of ID required? Getting Identity Management Right”, Australian IT Security Forum, 2004, available at: <http://www.privacy.gov.au/materials/types/speeches/view/6339>
- [9] “Identity”, Wikipedia, [cited] September 2009, available at: [http://en.wikipedia.org/wiki/Identity\\_\(philosophy\)](http://en.wikipedia.org/wiki/Identity_(philosophy))
- [10] “Identity Management Design Guide with IBM Tivoli Identity Manager”, 3<sup>rd</sup> edition, IBM Corp., April 2009, available at: <http://www.redbooks.ibm.com/redbooks/pdfs/sg246996.pdf>
- [11] “What is the data store”, available at: <http://technet.microsoft.com/en-us/library/cc787905%28WS.10%29.aspx>
- [12] D. Birch, “Digital Identity Management: Perspectives On The Technological, Business and Social Implications”, Gower Pub Co, 2007
- [13] “Sun Java System Directory Server Enterprise Edition 6.3 Evaluation Guide”, SunMicrosystems, 2008, available at: <http://dlc.sun.com/pdf/820-2766/820-2766.pdf>
- [14] M. Benantar, “Access Control Systems: Security, Identity Management and Trust Models”, Springer Science+Business Media, 2006.
- [15] “The Role of Kerberos in Modern Information Systems”, MIT Kerberos Consortium, 2008, available at: <http://www.kerberos.org/software/rolekerberos.pdf>
- [16] “The MIT Kerberos Administrator’s How-to Guide”, MIT Kerberos Consortium, 2008, available at: <http://www.kerberos.org/software/adminkerberos.pdf>
- [17] C. Neuman, T. Yu, S. Hartman, K. Raeburn, “The Kerberos Network Authentication Service (V5)”, RFC 4120, July 2005, available at: <http://tools.ietf.org/html/rfc4120>



- [18] “Recommended Practices for Deploying & Using Kerberos in Mixed Environments”, MIT Kerberos Consortium, 2008, available at: <http://www.kerberos.org/software/mixenvkerberos.pdf>
- [19] D. Todorov, ”Mechanics of user identification and authentication: Fundamentals of Identity Management”, Auerbach Publications, 2007
- [20] W. R. Stanek, “Windows Server 2008 insideout”, Microsoft Press, 2008
- [21] W. Soyinka, “Linux administration Beginner’s guide”, 5<sup>th</sup> edition, McGraw-Hill, 2008
- [22] T. Bialaski, Michael Haines, “LDAP in the Solaris Operating Environment: Deploying Secure Directory”, Prentice Hall, 2003
- [23] R. Harrison, “Lightweight Directory Access Protocol (LDAP): Authentication Methods and Security Mechanisms”, RFC 4513, June 2006, available at: <http://tools.ietf.org/html/rfc4513>
- [24] S. Huque “LDAP Weaknesses as a Central Authentication System”, University of Pennsylvania, 2007, available at: <http://www.huque.com/~shuque/doc/2007-10-LDAP-Authn.html>
- [25] “Investigating Single Sign-on”, Novell white paper, available at: [http://www.novell.com/rc/docrepository/public/37/basedocument.2007-08-07.2321076507/4622014\\_en.pdf](http://www.novell.com/rc/docrepository/public/37/basedocument.2007-08-07.2321076507/4622014_en.pdf)
- [26] “SunOpenSSO Enterprise 8.0 TechnicalOverview”, SunMicrosystems, March 2009, available at: <http://dlc.sun.com/pdf/820-3740/820-3740.pdf>
- [27] Jan De Clercq, “Single Sign-On Architectures”, Springer Berlin, 2002, available at: <http://www.springerlink.com/content/806c0atpq9ab0nx4/>
- [28] Dao Van Tran, Pål Løkstad, Do Van Thanh, ”Identity Federation in a Multi Circle-of-Trust Constellation”, Teletronikk, Telenor, 2007, Volume 103, pp. 103-117
- [29] “Security Assertion Markup Language (SAML) V2.0 Technical Overview”, OASIS Committee Draft, 2008, available at: <http://www.oasis-open.org/committees/download.php/27819/sstc-saml-tech-overview-2.0-cd-02.pdf>
- [30] “Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) V2.0”, OASIS standard, 2005, available at: <http://docs.oasis-open.org/security/saml/v2.0/saml-sec-consider-2.0-os.pdf>
- [31] K. Mayes, K. Markantonakis, “Smart cards, Tokens, Security and Applications”, Springer Science+Business Media, 2008.
- [32] B. Holcombe, “Government smart card handbook”, available at: <http://www.smartcard.gov/information/smartcardhandbook.pdf>.
- [33] W. Rankl, W. Effing, “Smart Card Handbook”, 3rd Edition, Wiley, November 2003.
- [34] “What Makes a Smart Card Secure?”, Smart Card Alliance Contactless and Mobile Payments Council White Paper, October 2008, available at: <http://www.smartcardalliance.org/pages/download>.
- [35] “MF1ICS70 Functional specification”, Rev. 4.1, January 2008, available at: [http://www.nxp.com/acrobat\\_download/other/identification/M043541\\_MF1ICS70\\_Fspec\\_rev4\\_1.pdf](http://www.nxp.com/acrobat_download/other/identification/M043541_MF1ICS70_Fspec_rev4_1.pdf).

- [36] J. Aussel, "Smart Cards and Digital Identity", *Teletronikk*, Telenor, 2007, Volume 103, pp. 66-79.
- [37] F. Vater, S. Peter, P. Langendorfer, "Combinatorial Logic Circuitry as Means to Protect Low Cost Devices Against Side Channel Attacks", Springer Berlin, 2007, available at: <http://www.springerlink.com/content/78k77t2m75731737/fulltext.pdf>.
- [38] "Common Criteria for Information Technology Security Evaluation", v.3.1, July 2009, available at: <http://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R3.pdf>
- [39] K.Boman, G.Horn, P.Howard, V.Niemi, "UMTS security", *Electronics & Communication Engineering Journal*, Volume 14, Issue 5, Oct. 2002 Page(s):191 - 204 October 2002, available at: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1088436&isnumber=23651>
- [40] M. Barbeau, Jean-Marc Robert, "Perfect identity concealment in UMTS over radio access links", *Wireless And Mobile Computing, Networking And Communications*, 2005. (WiMob'2005), IEEE International Conference on, Volume 2, 22-24 Aug. 2005 Page(s): 72 - 77 Vol. 2, available at: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1512853&isnumber=32399>
- [41] Abdul Bais, W.T. Penzhorn, P. Palensky, "Evaluation of UMTS security architecture and services" *Industrial Informatics*, 2006 IEEE International Conference on, 16-18 Aug. 2006 Page(s):570 – 575, available at: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4053451&isnumber=4053336>
- [42] J. A. Audestad, "Technologies and systems for access and transport networks", Artech House, 2008.
- [43] "UMTS", Wikipedia, available at: <http://en.wikipedia.org/wiki/Umts>
- [44] P. Hernberg, "User Authentication HOWTO", 02-05- 2000, available at: <http://www.faqs.org/docs/Linux-HOWTO/User-Authentication-HOWTO.html#AEN59>
- [45] "AIX: root group and system group ", [Cited] November 2009, available at: <http://www.linuxquestions.org/questions/aix-43/aix-root-group-and-system-group-751399/>
- [46] "The wheel Group", [cited] November 2009, available at: <http://administratosphere.wordpress.com/2007/07/19/the-wheel-group/>
- [47] "Hardening Linux authentication and user identity", September 23, 2004, available at: <http://www.linux.com/archive/articles/39114>
- [48] "The Linux Login Process", [cited] November 2009, available at: [http://www.comptechdoc.org/os/linux/howlinuxworks/linux\\_hlogin.html](http://www.comptechdoc.org/os/linux/howlinuxworks/linux_hlogin.html)
- [49] "Access Control lists in Linux", Administration guide, 18<sup>th</sup> July 2003, available at: [http://www.suse.de/~agruen/acl/chapter/fs\\_acl-en.pdf](http://www.suse.de/~agruen/acl/chapter/fs_acl-en.pdf)
- [50] A. Grunbacher, "POSIX Access control lists on Linux", USENIX Annual Technical Conference, San Antonio, Texas, June 2003, available at: <http://www.suse.de/~agruen/acl/linux-acls/linux-acls-final.pdf>
- [51] J. A. Martinez, M. Strasser, A. Tapaninen, T. Sirainen, L. Rousseau, "PAM-PKCS11 User Manual", [cited] December 2009, available at: [http://www.opensc-project.org/doc/pam\\_pkcs11/pam\\_pkcs11.html#introduction](http://www.opensc-project.org/doc/pam_pkcs11/pam_pkcs11.html#introduction)
- [52] Kenneth Geisshirt, "Pluggable Authentication Modules: The Definitive Guide to PAM for Linux SysAdmins and C Developers", Packt Publishing, 2007.

- [53] J. M. Johansson, "Windows Server 2008 Security Resource Kit", Microsoft Press, 2008
- [54] M. E. Russinovich, D. A. Solomon, "Microsoft Windows Internals: Microsoft Windows Server 2003, Windows XP, and Windows 2000", 4<sup>th</sup> ed., Microsoft Press, 2005
- [55] D. Todorov, "Mechanics of user identification and authentication: Fundamentals of Identity Management", Auerbach Publications, 2007
- [56] "Access Control Model", MSDN Library, 2009, available at: <http://msdn.microsoft.com/en-us/library/aa374876%28VS.85%29.aspx>
- [57] "Access Control Lists", MSDN Library, 2009, available at: <http://msdn.microsoft.com/en-us/library/aa374872%28VS.85%29.aspx>
- [58] "Access Mask Format", MSDN Library, 2009, available at: <http://msdn.microsoft.com/en-us/library/aa374896%28VS.85%29.aspx>
- [59] "How DACLs Control Access to an Object", MSDN Library, 2009, available at: <http://msdn.microsoft.com/en-us/library/aa446683%28VS.85%29.aspx>
- [60] "Privileges", MSDN Library, 2009, available at: <http://msdn.microsoft.com/en-us/library/aa379306%28VS.85%29.aspx>
- [61] "How Interactive Logon Works", Windows TechNet Library, 2009, available at: [http://technet.microsoft.com/en-us/library/cc780332%28WS.10%29.aspx#w2k3tr\\_intlg\\_how\\_tpxs](http://technet.microsoft.com/en-us/library/cc780332%28WS.10%29.aspx#w2k3tr_intlg_how_tpxs)
- [62] M. E. Russinovich, D. A. Solomon, A. Ionescu, "Windows Internals: Including Windows Server 2008 and Windows Vista", 5<sup>th</sup> ed., Microsoft Press, 2009
- [63] R. Morimoto, M. Noel, O. Droubi, R. Mistry, C. Amaris, "Windows Server 2008 Unleashed", Sams, 2008
- [64] "Application Compatibility: Session 0 Isolation", MSDN Library, 2009 available at : <http://msdn.microsoft.com/en-us/library/bb756986.aspx>
- [65] D. Griffin, "Create Custom Login Experiences With Credential Providers For Windows Vista", MSDN magazine, 2007, available at: <http://msdn.microsoft.com/en-us/magazine/cc163489.aspx>
- [66] "Understanding Logon and Authentication", 2005, available at: <http://technet.microsoft.com/en-us/library/bb457114.aspx>
- [67] L. Zhu, B. Tung, "Public Key Cryptography for Initial Authentication in Kerberos (PKINIT)", RFC 4556, 2006, available at: <http://tools.ietf.org/html/rfc4556>
- [68] A. Nirmalanathan, "The Smart Card Cryptographic Service Provider Cookbook ", Microsoft Corporation, 2002, available at: <http://msdn.microsoft.com/en-us/library/ms953432.aspx>
- [69] "Windows Vista Smart Card Infrastructure ", Microsoft Corporation, available at: <http://msdn.microsoft.com/en-us/library/bb905527.aspx>
- [70] "What's New in Smart Cards", 2009, available at: <http://technet.microsoft.com/en-us/library/dd367851%28WS.10%29.aspx>
- [71] Dave," How Mac OS X Implements Password Authentication, Part 2", and April 2006, available at: [http://www.dribin.org/dave/blog/archives/2006/04/28/os\\_x\\_passwords\\_2/](http://www.dribin.org/dave/blog/archives/2006/04/28/os_x_passwords_2/)

- [72] “Understanding and Using NetInfo”, [cited] November 2009, available at: [http://download.info.apple.com/Apple\\_Support\\_Area/Manuals/software/UnderstandingUsingNetInfo.PDF](http://download.info.apple.com/Apple_Support_Area/Manuals/software/UnderstandingUsingNetInfo.PDF)
- [73] “Authorization”, MAC OS X reference Library, [cited] November 2009, available at: [http://developer.apple.com/mac/library/documentation/Security/Conceptual/authorization\\_concepts/02authconcepts/authconcepts.html#//apple\\_ref/doc/uid/TP30000995-CH205-TPXREF9](http://developer.apple.com/mac/library/documentation/Security/Conceptual/authorization_concepts/02authconcepts/authconcepts.html#//apple_ref/doc/uid/TP30000995-CH205-TPXREF9)
- [74] “Mac OS X 10.4: Enabling smart card login”, MAC OS X reference Library, [cited] December 2009, available at: [http://support.apple.com/kb/TA24244?viewlocale=en\\_US](http://support.apple.com/kb/TA24244?viewlocale=en_US)
- [75] “RADIUS”, Wikipedia, [cited] 18<sup>th</sup> May, 2010, available at: <http://en.wikipedia.org/wiki/RADIUS>
- [76] Jon Edney, Willam A. Arbaugh, "Real 802.11 Security: WiFi Protected Access and 802.11i", Addison Wesley, 2003.
- [77] “Simple Pairing Whitepaper”, v.10r00, Core Specification Working Group, 2006, available at: [http://www.bluetooth.com/NR/rdonlyres/0A0B3F36-D15F-4470-85A6-F2CCFA26F70F/0/SimplePairing\\_WP\\_V10r00.pdf](http://www.bluetooth.com/NR/rdonlyres/0A0B3F36-D15F-4470-85A6-F2CCFA26F70F/0/SimplePairing_WP_V10r00.pdf)
- [78] K. Scarfone, J. Padgette, “Guide to Bluetooth Security”, National Institute of Standards and Technology (NIST), NIST Special Publication 800-121, September 2008, available at: <http://csrc.nist.gov/publications/nistpubs/800-121/SP800-121.pdf>
- [79] “National Information Assurance Glossary”, available at: [http://www.cnss.gov/Assets/pdf/cnssi\\_4009.pdf](http://www.cnss.gov/Assets/pdf/cnssi_4009.pdf)
- [80] “Extensible Authentication Protocol Method for Global System for Mobile Communications (GSM) Subscriber Identity Modules (EAP-SIM)”, RFC 4186, available at: <http://tools.ietf.org/html/rfc4186>
- [81] “S/Key”, Wikipedia, [cited] November 2009, available at: <http://en.wikipedia.org/wiki/S/KEY>
- [82] “HOTP: An HMAC-Based One-Time Password Algorithm”, RFC 4226, available at: <http://tools.ietf.org/html/rfc4226>
- [83] “TOTP: Time-based One-time Password Algorithm”, Internet-Draft, available at: <http://tools.ietf.org/html/draft-mraihi-totp-timebased-03>
- [84] “OCRA: OATH Challenge-Response Algorithms”, Internet-Draft, available at: <http://tools.ietf.org/html/draft-mraihi-mutual-oath-hotp-variants-09>
- [85] “Component (UML)”, [Cited] 21<sup>st</sup> April, 2010, available at: [http://en.wikipedia.org/wiki/Component\\_UML](http://en.wikipedia.org/wiki/Component_UML)
- [86] Donald Bell, “UML basics: The class diagram”, [cited] 3<sup>rd</sup> May, 2010, available at: <http://www.ibm.com/developerworks/rational/library/content/RationalEdge/sep04/bell/>
- [87] “UML 2 Package Diagrams”, [Cited] 3<sup>rd</sup> May, 2010, available at: <http://www.agilemodeling.com/artifacts/packageDiagram.htm>
- [88] “Random”, [Cited] 15<sup>th</sup> May, 2010, available at: <http://linux.die.net/man/4/random>

## Appendix A

### Sample Code for PAM Module

```
#include <security/pam_appl.h>
#include <security/pam_misc.h>
#include <stdio.h>

//Conversation function
static struct pam_conv conver = {
    misc_conv,
    NULL
};

// main function
int main(int argc, char *argv[])
{
    pam_handle_t *pamhandler=NULL;
    int retvalue;
    const char *user="username";

    // starting PAM session and initializing the PAM data structure
    retvalue = pam_start("check", user, &conver, &pamhandler);

    //check if the PAM session is created
    if (retvalue == PAM_SUCCESS)
        retvalue = pam_authenticate(pamhandler, 0); /*Authenticating user */

    //check if user is authenticated
    if (retvalue == PAM_SUCCESS)
        retvalue = pam_acct_mgmt(pamhandler, 0); /* permitted access? */

    /* This is where we have been authorized or not. */

    if (retvalue == PAM_SUCCESS) {
        fprintf(stdout, "User is Authenticated\n");
    } else {
        fprintf(stdout, "User is not Authenticated\n");
    }

    if (pam_end(pamhandler,retvalue) != PAM_SUCCESS) { /* close Linux-PAM */
        pamhandler = NULL;
        fprintf(stderr, "check_user: failed to release authenticator\n");
        exit(1);
    }
    // indicate success or failure

    return ( retvalue == PAM_SUCCESS ? 0:1 );
}
//////////////////////////////////Code END//////////////////////////////////
```

## Compilation of PAM Module

For compiling the PAM Module the following libraries files should be added. The command will be like

```
gcc filename.c -lpam -lpam_misc -o outputfile
```

For creating .so files which can be used than in PAM configuration files for accessing the service we should do the following steps

```
$ gcc -fPIC -c filename.c  
$ ld -x --shared -o filename.so filename.o  
$ sudo cp filename.o /lib/security
```

## **Appendix B**

The source code for the secure mobile based authentication prototype is attached as a ZIP file. The attachment has the following data

1. Source code for the PAM module for the client side
2. Source code for the Authenticator
3. A “read me” file explaining how to use the prototype