SECURITY REQUIREMENTS IN CLOUD SYSTEM

A Dissertation submitted in partial fulfilment of the requirement for the award of degree of

Master of Technology

In

Software Engineering

Submitted by

Puneet Meerwal

Roll No.- 2K11/SWE/09

Under the guidance of

Prof. (Dr.) Daya Gupta

DEPARTMENT OF COMPUTER ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

BAWANA ROAD, DELHI

2011-2013

CERTIFICATE

DELHI TECHNOLOGICAL UNIVERSITY
(Govt. of National Capital Territory of Delhi)
BAWANA ROAD, DELHI – 110042

This is to certify that dissertation entitled “***Security Requirements in Cloud System***” has been completed by **PUNEET MEERWAL** (Roll Number: **2K11/SWE/09**) for partial fulfillment of the requirement of **Master of Technology** degree in **Software Engineering**. This work is carried out by him under my supervision and has not been submitted earlier for the award of any degree or diploma in any university to the best of my knowledge.

Date: ___________

Prof. (Dr.) DAYA GUPTA
HOD & Project Guide
Department of Computer Engineering ,
DTU, Delhi

ACKNOWLEDGEMENT

This dissertation is the results of study in "***Security Requirements in Cloud System***" which was conducted in the fourth semester at the Department of Computer Engineering in Delhi Technological University (formerly Delhi College of Engineering), Delhi.

Foremost I would like to express my sincere gratitude and indebtedness to my learned supervisor **Prof. (Dr.) DAYA GUPTA**, **HoD, Department of Computer Engineering, Delhi Technological University, Delhi** for her invaluable guidance, encouragement and patient reviews. Her continuous inspiration only has made me complete this dissertation. She provided the conceptions and theoretical background for this study as well as suggested the rational approach.

Besides my supervisor, I am also highly thankful to **Ms. Shruti Jaiswal** who enlightened me at every step by giving helpful direction and guidance.

(PUNEET MEERWAL)
M. Tech (Software Engineering)
Dept. of Computer Engineering
DTU, Delhi

ABSTRACT

Cloud computing offers immediate availability and massive scalability with low cost as major benefits, but it introduces new risks and vulnerabilities too. In practice security considerations are usually incorporated in the later stages of development, resulting in rise of vulnerabilities. To help address these issues in Cloud computing, this study presents an approach to elicitate true Security Requirements early in the development process by considering functional requirements of actors and possible threats. We also conduct an in depth analysis of some available cloud storage services to prove the need of this approach for a secure Cloud environment.

The results of this thesis could potentially encourage the Security Requirement engineers to follow a structured elicitation approach and to consider various viewpoints, which help’s to develop secure Cloud applications, thus, contributing to secure Cloud environment. It also enables the organizations to gain a better understanding on how the security could be implemented to best effect.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

“This page is intentionally left blank”

Chapter 1

INTRODUCTION

‘Security Engineering’ is emerging as a new challenging research area for Software Engineering community with the evolution of new techniques and software. In the last two decades, reports of software security failure have become common and these security failure incidents are growing day by day [1]. The major source of these failures is due to the tolerance of Security Requirements during the development of the system. Results from various surveys and reports also shows that ‘Security’ is a top most concern [2] in Cloud computing and should consequently be subject to careful Security Requirements analysis and decision making early in the application development process related to Cloud environment. As Cloud computing is itself an amalgamation of various technologies resulting in a complex architecture, hence it needs a special emphasis on Security Requirements early during the application development process.

1.1 General Concepts

Cloud systems are more vulnerable to threats that bring various types of damages resulting in endanger of human life and serious loss to major economic infrastructures. These threats can range from errors destroying database integrity to natural forces like fires destroying entire computer data centers. All these threats and attacks gave birth to computer security which is defined as a measures and controls that ensures confidentiality, integrity & availability of the information or data processed and stored by the computer [3].

Software intensive systems are neither perfect nor invulnerable [4] as they usually fail due to hardware breakdown, software glitches, accidental misuse and intentional abuse. They are also deliberately attacked by malicious hackers, criminals & terrorists, industrial spies and even foreign government or military agents. Sometimes we are not even aware that whether we are safe or not when we logon to our computer systems, as there may be a possibility that someone is tracking our activities and private data. Similarly overlooking Cloud system security is not a viable option since society heavily relies upon them. Nowadays we use Cloud system for various activities like storage, computation, development, business applications related to financial matters, which are so critical that if they got attacked by intruders then they can make a serious impact on organizations as well as persons who are using them. Thus, Security Engineering is nowadays becoming an essential component of system engineering.

'Security Engineering' is a relatively new and emerging field which as discipline focuses on security aspects of processes, tools and methods to design & test the complete system. Security Engineering basically deals with following [3]:

- Security Requirements specification and management
- Implementing Security Requirements while designing a system
- Implementing specific mechanism and algorithms to make a system acceptable in real world environment.

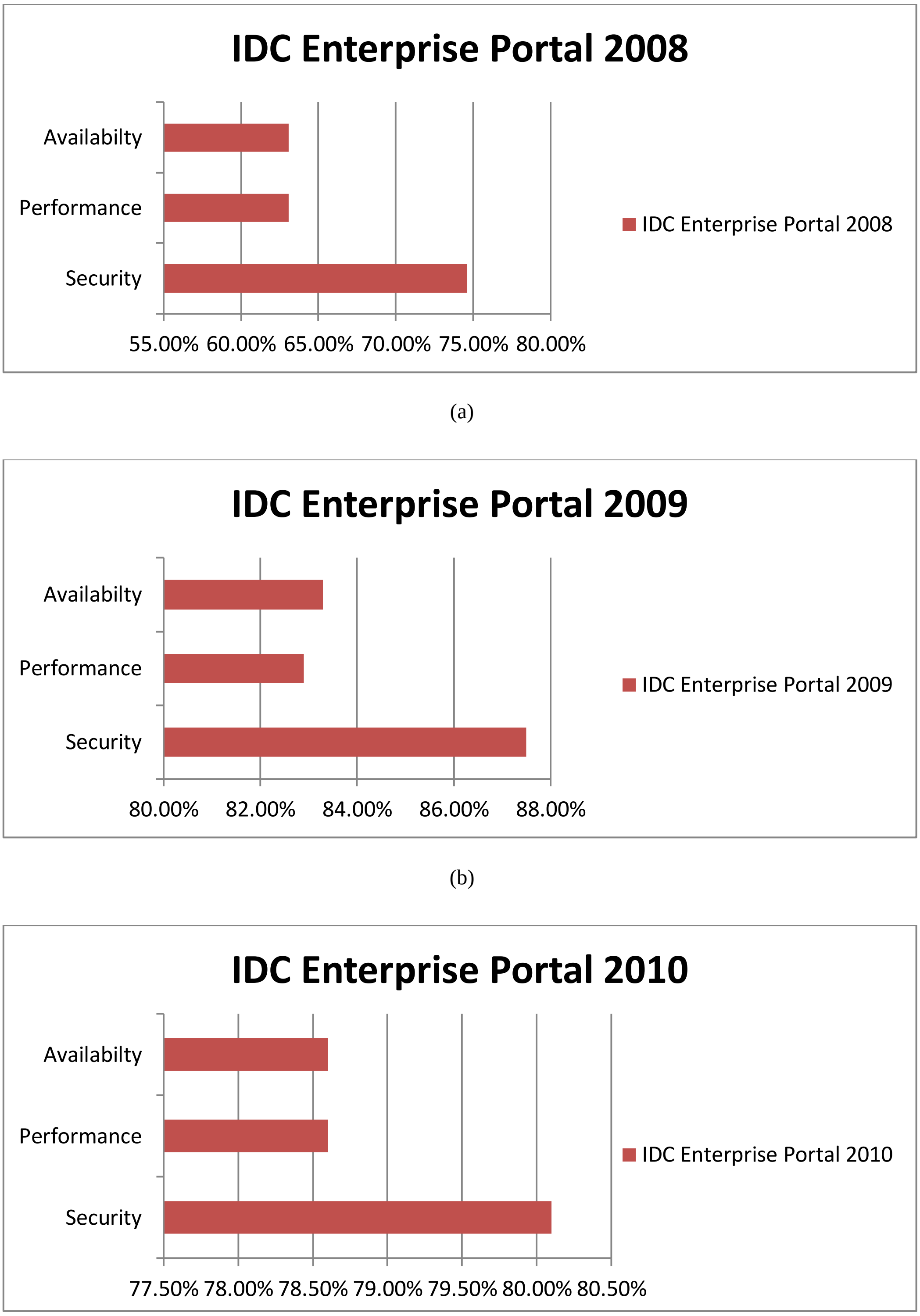
Cloud computing basically provides the computing facility as a service over the internet [5] and due to its complex architecture, development in Cloud environment requires a strong commitment to a secure software development life cycle, including requirement analysis, design, testing, deployment and disposal.

1.2 Motivation

Cloud computing is nowadays becoming an integral part of major business organizations and individuals, as they are shifting their business operations and critical data to Cloud systems. Hence it is necessary to protect this critical data and operations from the opposition and attackers and to ensure that the information within Cloud system is secure.

During the process of developing a computer based system the first and most important step is the gathering of requirements. Gathering requirements is the hardest part [6] of developing as it deals with deciding precisely what to build. As any fault or errors in this process can lead to the system which can fail under certain circumstances or are not up to the customer expectations. Similarly elicitation and specification of Security Requirements is also a difficult task as they specify what is to be prevented and how. So these all mandatory requirements must be gathered in early phases of SDLC so as to build a reliable and good quality system.

As per the reports released by data giant IDC [2], the top most concern in Cloud computing is ‘Security’, as shown for three consecutive years in Figure 1. This statistics is the driving force of this project, which aims to build a secure Cloud environment. Software engineering community feels that to implement security, first security requirements should be gathered, analyzed and then implemented [7,8,9,10]. Also our research shows that vary few proposals exist in literature for Security Requirements engineering in Cloud system. Hence there is a need to address this issue. This motivated us to explore the work on elicitation of Security Requirements related to Cloud system.



(a)

(b)

(c)

Figure 1: Survey data on issues in Cloud computing [2]:

(a) IDC Survey 2008 (b) IDC Survey 2009 (c) IDC Survey 2010

1.3 Related Work

In last decade researchers have been working on elicitation of Security Requirements in different domain such as banking, railway reservation and health care sector etc. Our research could find only few proposals in the domain of Cloud where ‘Security’ is a great concern. If proper measures are not taken for security in the early stages it may lead to an inefficient system or may result in a failure.

Security basically deals with protecting CIA (confidentiality, integrity and availability). But Firesmith [3] has defined twelve different Security Requirements covering CIA as well as identification, authorization, immunity etc. He also distinguished between Security Requirements and architectural constraints so that true Security Requirements can be identified which can lead to cost effective secure system.

In literature we can find different elicitation techniques like abuse cases [20], common criteria [21,22], misuse cases [23,24], security use cases [6], attack trees [26], secure tropos [27] and intentional anti model [28]. There are also proposals for Security Requirements Engineering where Security Requirements are elicited, analyzed and prioritized [7,8]. Then the proposals for design decision [9,10] and Security testing [11] are found for proper implementation and testing of Security Requirements.

Cloud computing researchers have also recognized the need to address these requirements but only few proposals have focused on Security Requirements and their elicitation related to Cloud System. Following are some work done by researchers in the field of security requirements elicitation for Cloud system.

Hanna [12] proposed a security analysis process which capture and analyze Security Requirements in Cloud Computing. His proposed method first identifies the assets that need to be protected and attacks that could be implemented on them and lastly identifies the countermeasure. His proposed process prevents or mitigates threats posed by external misusers to the Cloud; he gives little consideration to threats posed by internal misusers like persons who are authorized to access.

Iliana & Maya [13] also identified Security Requirements related to Cloud computing and classified those identified requirements into nine sub-classes. But they have not proposed or used any framework based on which they have identified the security requirements.

1.4 Problem Statement

From the foregoing section we can conclude that there are not concrete proposals for Security Requirements Engineering in the domain of Cloud. Existing approaches do not consider all possible security concerns. In addition there is no framework or formal method for elicitation, analysis and documentation.

Hence the problem of thesis is

"Propose a structured framework for Security Requirements Engineering for Cloud system and apply it to develop a secure system for Cloud Storage-as-a-service model."

The framework has well defined steps which describe how Security Requirements are elicited in an efficient way. First high level Security Requirements as proposed by Firesmith [3] are identified, and then they are extended as low level Security Requirements.

This proposal on Security Requirements will be useful for Security Requirement engineers to determine the structured way through which elicitation can be performed on Cloud system. The low level Security Requirements also helps Design engineers in taking design decision to implement Security criteria. Not only by Software engineers this approach can also be used by various researchers to gain better understanding of Security Requirements & its elicitation and give their best to enhance the approach by applying it to other domains for creating a secure society.

1.5 Scope of Work

The real motive of proposal is to provide a well defined way for identifying Security Requirements in Cloud system. Earlier proposals by researchers are very limited and also unable to consider every possible security concern in Cloud. Their proposed approaches are also not based on any process, method or framework. So we wish to extend the framework for Security Requirements Engineering presented by Agarwal & Gupta [7] in the domain of Cloud system.

In this project we adapt the Security Requirements classification of Firesmith [3] in the domain of Cloud System. We then define low level Security Functionalities in the same domain [29]. These Security Functionalities are nothing but are low level Security Requirements which helps the design engineer to take optimal decision in implementing security. Our process consists of identifying different Actors, their functionalities, identifying Security Requirements and then maps them to Security Functionalities. We then finally illustrate our approach for Cloud Storage-as-a-service.

Hence our work will cover:

- Eliciting different Security Requirements related to Cloud system.

- Defining Security Functionalities i.e. low level Security Requirements.
- Presenting a step by step method to elicit Security Requirements.
- Perform a case study on real world Cloud Storage services to ensure security.

1.6 Organization of Thesis

The rest of this thesis is organized as follows:

Chapter 2 provides a basic overview of Cloud computing by first giving definition and the history associated with Cloud computing. Then it describes the various models available in Cloud computing with their merits and demerits. And lastly it discusses the advantages of Cloud with some available issues.

Chapter 3 gives the overview of our study on Security Engineering. It first gives a brief idea about Security Engineering activities. Then it introduces Security Requirements Engineering with various elicitation methods proposed during last two decades. At last it presents Security Design Framework and Security Testing Framework proposed by researchers.

Chapter 4 contains our proposal of Security Requirement Engineering for Cloud system. It describes the five steps involved in our approach i.e. identification of actors in Cloud, identification of functionalities for each actor, identification of threats related to functionality, determination of Security Requirements with Security Functionalities.

Chapter 5 covers our analysis done on two real worlds Cloud Storage services. It first discusses their available features with functionalities and the threats possible on them. Then it proves the feasibility of our approach for developing similar services.

Chapter 6 contains our implementation by first giving instructions to configure and then shows some snapshots of tool developed to elicit Security Requirements.

Chapter 7 finally concludes the thesis.

Chapter 2

CLOUD COMPUTING OVERVIEW

In this chapter we first discuss the basic definition and history of Cloud computing. Then the various models of Cloud system with their merits and demerits are given for better understanding and finally we discuss the advantages of Cloud services with their available issues.

2.1 Cloud Definition

In most of the literatures the name Cloud computing relates to the images of clouds that are representing networks and the Internet. Cloud computing was implemented into the real world because of the pressure on IT to save money and now it became future of next generation IT. Basically, Cloud computing makes data and applications available through the Internet to the Cloud users. Cloud computing is not a new technology or a new device but it is a use of existing technology and devices in a new way. A standard definition of Cloud computing given by NIST is

*"**Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction**."* [14]

Cloud computing architecture basically consists of hierarchical levels as shown in Figure 2.

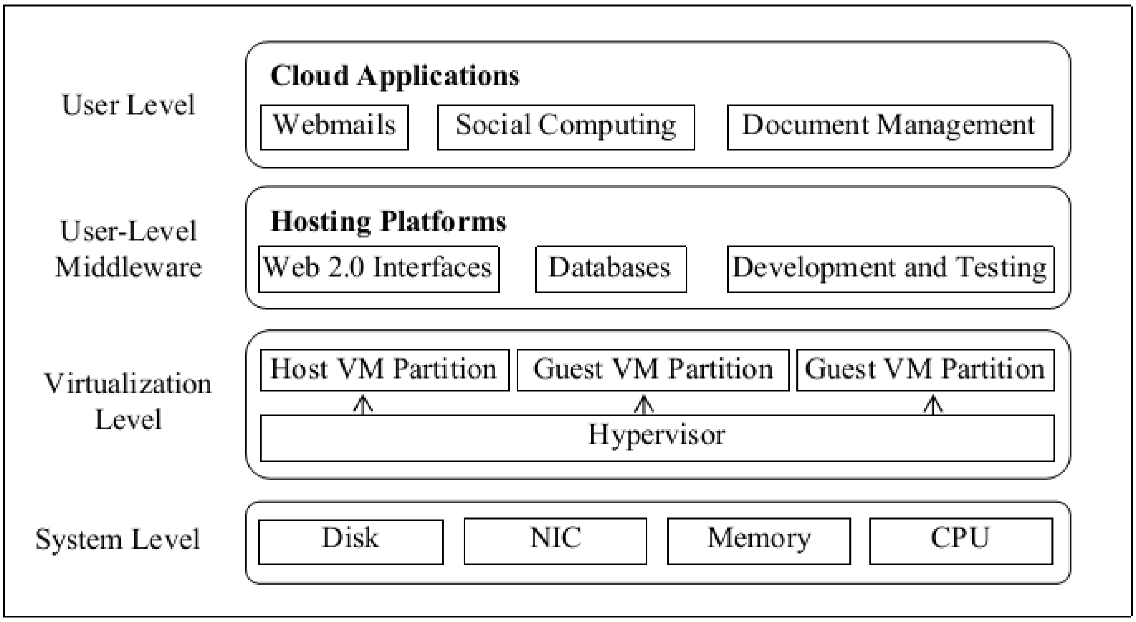


Figure 2: Cloud computing architecture

2.2 History of Cloud

John McCarthy in 60’s said that “*computation may someday be organized as a public utility*” and today Cloud computing really seems to be breaking through. In 60’s and 70’s companies had large and expensive mainframe computers which provide the services to workers who access them through dumb terminals. These mainframe computers store all information and did all calculations.

Then in 80’s these mainframes are replaced by computers for the users due to decrease in the price of personal computers. In 90’s with the advancement of Internet technology the fashion of many computers accessing one big server came again. At that time requirement of web servers arises with plenty of powers to resolve requests that were made from the Internet. Since from that time to today numerous services are offered in Internet with more storage capacity and massive computation requirements are solved by dedicated service providers. In this way large users can share the common infrastructure maximizing efficiency and minimizing the cost.

At the finish of 90's, normally all data centers were using only less than 10% of their capabilities [5] as they wanted to reserve the rest in case of occasional peaks. At this time Amazon made a great effort to solve this problem by adding capabilities on demands by the users.

In 1999 Salesforce.com began to deliver services to organizations by their own website and initiated the concept of software as a service. In 2002 Amazon launched AWS suite that includes storage, computation with other services. Again in 2006 Amazon launched EC2 for small companies and users to let them run their own computer applications in Cloud. In 2008 Eucalyptus was launched, which was the first open source AWS API compatible platform for deploying private Clouds. In 2009 Google began to offer enterprise applications as Google AppEngine. And today all large companies like Microsoft, IBM, Oracle and HP offers Cloud computing with various services.

2.3 Cloud Models

Based on the underlying infrastructure and the services offered to users, Cloud systems are classified as Cloud deployment models and Cloud service models which further consists various sub-models [14].

2.3.1 Deployment Models

In Cloud deployment models the available sub-models are public, private, hybrid and community which are distinguished by their architecture, location of datacenter and the needs of the Cloud customer.

- **Public Cloud**

  Public Cloud offers the computing resources to general public over the internet via Web applications or Web browsers either on free or on pay-per-use license policy [14]. It is advantageous as the customer does not have to buy any equipment and the resources are shared among different customers at a time. Public Cloud's physical infrastructure is owned by a CSP. Its limitation is the less control over the hardware.

- **Private Cloud**

  It offers the infrastructure to be used by only one organization which can be located on the premises of the CSP. These are used in private networks and hence restrict the unwanted public access to the data that is used by the organization [5]. Advantage of this model is the total control over the hardware by an organization. It is also more secure than the traditional public Cloud. Its limitation is the high cost.

- **Hybrid Cloud**

  Hybrid Cloud consists of combination of public and private Cloud. So through its implementation an organization can benefits from the advantages of both Clouds. For example an organization can runs all applications in private Cloud and uses public Cloud when private Cloud lacks from certain features.

- **Community Cloud**

  Community Cloud offers to share the resources and hardware between organizations that have similar needs. So it is a private Cloud for a community, where community consists of organizations fewer than public Cloud and more than private Cloud. Security of data is compromised in this model.

Hence, for deciding which type of Cloud to deploy in organization, the business managers need to assess each Cloud deployment model from multiple points of view like cost, economy, availability etc. A comparison between various Cloud deployment models with their merits and demerits are summarized in Table 1.

Table 1: Deployment models comparison

Cloud	Merits	Demerits
Public	Efficient use of hardware No need to buy hardware	Data stored off-premise
Private	Control over hardware Control over data	High Cost Hardware has to be bought
Hybrid	Critical information can stay on premise	Less efficient than public cloud
Community	Cost can be spread Efficient use of hardware	Less efficient than public cloud

2.3.2 Service Models

Similar to deployment models, the service models are also sub-divided mostly into four categories as Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS) and Storage as a Service (StaaS) [14].

- **Software as a Service (SaaS)**

  In SaaS model the CSU access the application software installed and maintained by CSP at providers end. In this the implementation and deployment is abstracted from the user and only limited set of configuration control is made available by provider. Its main benefit is the reduction in hardware cost and software development and maintenance cost. Examples of SaaS are MS Office 365, Quickbooks online and Salesforce.com.

- **Platform as a Service (PaaS)**

  The CSP provides the computing platform as on demand service on which applications can be developed and deployed. In addition to computing platform a solution stack consists of operating systems, programming language environment, databases and web servers is also provided to customers. This model is mostly suitable for developers. Its purpose is to reduce cost and complexity of buying and managing underlying hardware and software components. Examples of PaaS are GAE, Force.com and Windows Azure Compute.

- **Infrastructure as a Service (IaaS)**

  In IaaS model the computing infrastructure like servers, network equipment's and software are provided as on-demand services where the customers can install operating system images with applications to create their own customized environment. The CSP owns the hardware and is responsible for housing and maintaining them. Examples of IaaS are Rackspace Cloud, Amazon EC2, Google Compute Engine and GoGrid.

- **Storage as a Service (StaaS)**

  In StaaS model the provider provides the storage services on their own infrastructure. Cloud storage system can be considered as a network of distributed data servers which use cloud computing features like virtualization and provide some kind of interface for storing the customer data. Basic features of Cloud storage services are copy, backup, synchronization and file sharing. Examples of StaaS are Dropbox, Mozy, and Cloud One etc.

In terms of efficiency and cost, the best suitable service model is SaaS but IaaS is best related to control over hardware and data as shown in Figure 3.

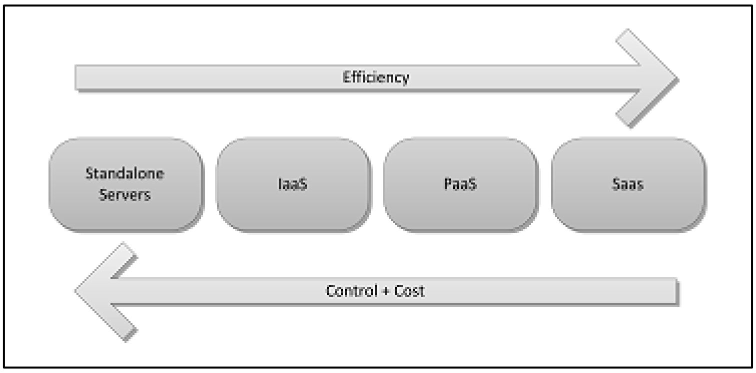


Figure 3: Efficiency & cost related to Cloud service models

2.4 Advantages of Cloud Computing

While Cloud computing is undoubtedly beneficial for mid-size to large organizations, it is not without its downsides especially for smaller companies. If used with care and to the extent necessary, working with data in Cloud can immensely benefit for all types of businesses [5].

- **Cost Efficient**

  Cloud computing is probably one of the most cost efficient method to use. Traditional desktop software with multiple users costs companies a lot in terms of economy. The Cloud, on the other hand, is available at much cheaper rates and hence, can drastically lower the company’s IT expenses. Besides this the pay-as-you-go and other scalable options available makes it very reasonable for the business in use.

- **Almost Unlimited Storage**

  Storing information in the Cloud gives us almost unlimited data storage capacity. Hence, we no more need to think about running short of storage space or increasing our current storage space capacity.

- **Backup & Recovery**

  Since all our critical data is stored in the Cloud, so backing it up and restoring the same is relatively much easier in Cloud than storing the same on a physical device. Hence, this makes the entire process of backup and recovery much simpler than other traditional methods of data storage.

- **Automatic Software Integration**

  In Cloud, software integration is usually something that occurs automatically. It means that we do not need to take additional efforts of customizing and integrating our applications as per our preferences. Not only this, Cloud computing allows us to customize our options with great ease. One can easily handpick just those services and applications that they think will best suit their particular enterprise.

- **Easy Access to Information**

  Once we register ourselves in the Cloud, we can access our information from anywhere, where an Internet connection is available. This convenient feature lets us to move beyond time zone and geographic location issues.

- **Quick Deployment**

  Lastly and most importantly, Cloud computing give us the advantage of quick deployment. Once we opt for this method of functioning, our entire system can be fully functional within few minutes.

2.5 Issues in Cloud Computing

Concerns related to Cloud computing are given and surveyed by various researchers from Cloud domain, some of them are [15]:

- **Security**

  As Cloud computing is gaining popularity, concerns about the security issues introduced through adoption of this new model. However, in Cloud, your data will be distributed regardless of where your base repository of data is ultimately stored. Industrious hackers can invade virtually any server, and there are the statistics that show that one-third of breaches result from hackers and attackers [16].

- **Privacy**

  Cloud computing uses the virtual computing technology, users personal data may be scattered in various virtual data center rather than stay at the same physical location, even across the countries borders, hence data privacy protection will face the controversy of different countries legal systems [17].

- **Availability**

  Like any external service cloud computing also requires a high degree of availability to prevent an adverse impact on business operations. When using a Cloud computing for any business critical operations, consumers must evaluate the risk associated with loss of connectivity to the CSP.

- **Integrity**

  Integrity in simple terms refers to the fact that data cannot be modified by unauthorized person. An organization doesn't want its customers to see data of other customers stored in same server but it is also not wanted that they can alter data uploaded by customers.

- **Reliability**

  The Cloud servers also have the same problem as our own resident servers; they experience downtimes and slowdowns, what the difference is that users have a higher dependent on CSP in the Cloud computing model.

- **Legal Issues**

  Various legal issues like trademark infringement, security concerns and sharing of proprietary data resources arises with Cloud computing. The question comes that ‘who is in possession of the data’ and what happens if provider-customer relationship ends?

- **Vendor Lock-In**

  Most Cloud platforms and services are proprietary, means they are built on the specific standards, tools and protocols developed by a particular vendor for its particular Cloud offering [18]. This will make migrating off a proprietary Cloud platform prohibitively complicated and expensive.

- **Compliances**

  Many regulations pertain to storage and usage of data require regular auditing and reporting trails, so Cloud providers must enable their customers to comply appropriately with these regulations. The data centers maintained by Cloud providers must also be subject to compliance requirements.

Chapter 3

SECURITY REQUIREMENTS ENGINEERING OVERVIEW

3.1 Security Engineering

A framework for Security Engineering Process (SEP) is proposed by [9,10]. That consists of four activities that are Security Requirements Engineering, Security Design Engineering, Security Requirements implementation and Security Requirements testing as shown in Figure 4.

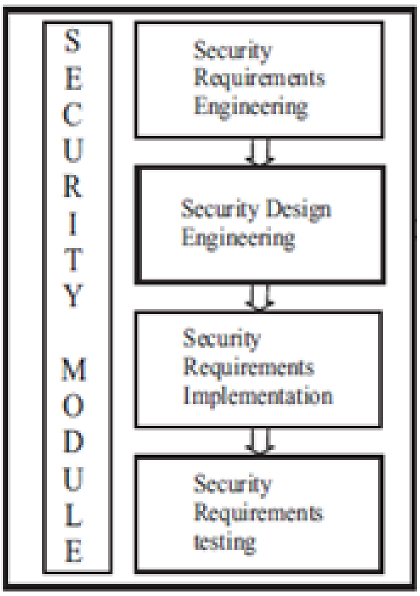

SECURITY MODULE
Security Requirements Engineering
Security Design Engineering
Security Requirements Implementation
Security Requirements testing

Figure 4: Framework for Security Engineering Process [9]

- ***Security Requirements Engineering***

The main aim of this phase is to identify all Security Requirements early during the development process. It consists of four different stages: Security

Requirement Elicitation, Security Requirement Analysis, Security Requirement Prioritization and Security Requirement Management.

- ***Security Design Engineering***

  In this phase Security design decisions are taken which include mapping of Security Requirements with cryptographic services such as authentication, confidentiality, etc., and then mapping attacks to prioritized threats. After that design decisions are taken which consider environmental constraints such as memory, encryption speed, energy etc. and security design attributes such as throughput, target platform, cost, etc. It then generates Security Design Template (SDT) that guides security engineer to finalize design decisions that specify which cryptographic technique is best suited for particular environment and design constraints.

- ***Security Requirements Implementation***

  In this phase all functionalities are implemented incorporating design decisions of the system. This includes implementing specific techniques that are suggested in the design phase of the Security Engineering process.

- ***Security Requirements Testing***

  This phase involves evaluating the system security and determining the adequacy of security mechanisms, assurances and other properties to enforce system security policies. It consists of four different stages: identify the attacker and possible attacks, find vulnerable points, security mechanism verification & validation and generate a test report template.

Our main emphasis in the thesis is on Security Requirements Engineering which is the first phase of Security Engineering framework and also the most important one because it feeds to other phases. We will discuss these phases in detail in the following sections.

3.2 Security Requirements Engineering

If Security Requirements are not properly elicited, analyzed and managed they result in a system that can fail. So they are even more important than all the requirements described above. These requirements are generally called as Security Requirements because they are responsible for the security of the system. Security Requirements Engineering is defined as a process of eliciting, analyzing and documenting the Security Requirements.

Security Requirements Engineering process consists of four sub processes which are Security Requirement Elicitation, Security Requirement Analysis, Security Requirements prioritization and Security Requirement management as shown in Figure 5.

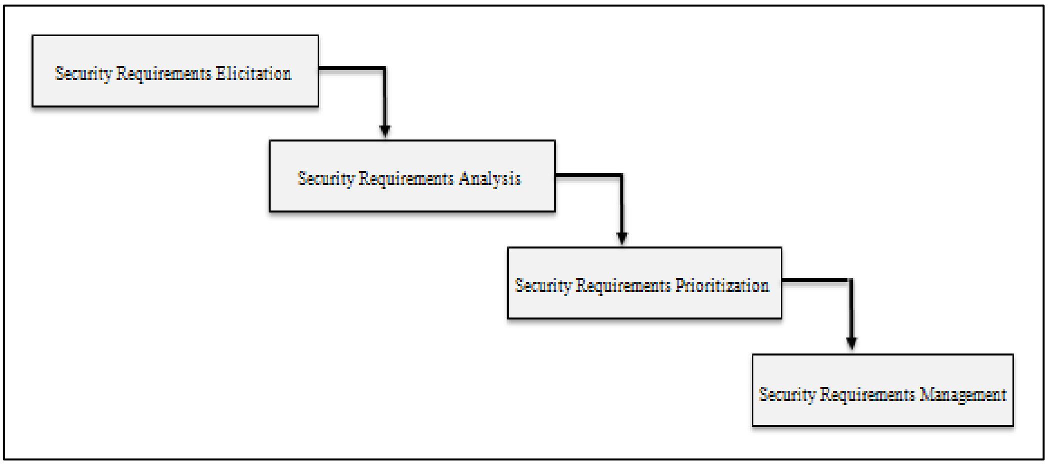

Security Requirements Elicitation
Security Requirements Analysis
Security Requirements Prioritization
Security Requirements Management

Figure 5: Security Requirements Engineering process

Important terminologies related to Security Requirements Engineering [19] are given in Table 2.

Table 2: Security related terms

Terms	Description
Threat	Circumstances that have potential to cause harm
Asset	Something valuable that needs to be protected. The asset may be the data or the software itself
Exposure	Loss or harm to a computing system. This can be loss or damage to data
Vulnerability	Weakness of a system that may be exploited to cause harm
Attack	Exploitation of systems vulnerabilities. Attack may be accidental or deliberate

3.3 Security Requirements

Security Requirements are defined as a **"high level requirement that gives detailed specification of the system behavior that is unacceptable like every user can only access data for which they are properly authorized [3]."**

Security Requirements are also known as '*shall not*' requirements that define unacceptable system behavior. Some important points that differentiate Security requirements are:

- Functional requirements are derived from goals of a system where as Security Requirements are objective resulting from threats on functionality or confidential data.
- Security requirements are related to non functional requirements like interoperability, feasibility and correctness etc. For example NFR like

correctness if implemented covers to some extent the integrity security requirement.

There are various types of Security Requirements in literature but D. G. Firesmith [3] defined them properly for a computer based system. Twelve Security Requirements given by Firesmith are:

- ***Identification Requirements***

Identification requirement specifies the extent to which a computer based system identifies external entities before interacting with system. These external entities may be a human actor or an external application.

- ***Authentication Requirements***

Objective of authentication requirement is to ensure that external entities are actually who or what they claim to be.

- ***Authorization Requirements***

It specifies that only authenticated external entities can access specific applications or information only if they are explicitly authorized to do so by the admin of the application.

- ***Immunity Requirements***

Immunity requirements specify the extent to which an application shall protect itself from infections caused by viruses, worms or Trojan etc. Its objective is to prevent any undesirable programs from destroying or causing harm to data and applications.

- ***Integrity Requirements***

Integrity requirement specifies the extent to which an application shall ensure that its data does not get intentionally corrupted or modifies through unauthorized creation, modification and deletion. Its objective is to ensure that data and communication can be trusted.

- ***Intrusion Detection Requirements***

This security requirement specifies that if an application or component is attacked by unauthorized individuals or programs then that can be detected and recorded so that the security personnel can properly handle them.

- ***Non repudiation Requirements***

This requirement specifies that a party should not deny its participation after interacting with the application or a business. Its objective is to ensure that adequate records are kept to prevent parties from denying interactions that have taken place earlier.

- ***Privacy Requirements***

This security requirement specifies that the application or business should keep its critical data and communications private from unauthorized individuals and programs. It says that concern persons are able to access only their data.

- ***Security Auditing Requirements***

This security requirement specifies the extent to which an application or organization shall allow security personnel to audit the use and status of the security mechanism.

- ***Survivability Requirements***

This security requirement specifies that an application should provide basic functionality in degraded mode even if some destruction has been there in the application. Its objective is to ensure that an application or center either fails gracefully or provide functionality even though certain components have been intentionally damaged.

- ***Physical Protection Requirements***

This security requirement specifies that an application or data center shall protect itself from physical assault. Its objective is to ensure that applications or data centers are protected against any physical damage or theft done intentionally or unintentionally.

- ***System Maintenance Requirements***

Security requirements in this category specify that an application or data center shall prevent authorized modification from accidentally defeating the security mechanism.

3.4 Security Requirements Elicitation Methods

Security Requirements elicitation is important in the early stages of the SDLC because it determines whether the system when exposed to real world survives or not. However generally, Security Requirements are discovered after a product is developed which may invite some possible attacks [3]. The elicitation methods in this section are discussed in chronological order of their proposals from various researchers.

3.4.1 Abuse Cases

Abuse case proposed by McDermott [20] represents a specification of complete interaction between the system and its environment, where the interaction can cause harm to system. The complete abuse case defines an interaction that results in harms to the resource associated with one of the actors, stakeholders or the system itself. An abuse can be accomplished by gaining total control over target machine through modification of system software or firmware. Abuse cases are described using the same strategy as for use case where abuse cases are kept separate and should describe the abuse of privilege used to complete the abuse case. First step in developing abuse cases is to find malicious actors and when we have actors, identify the abuse cases by determining how they might interact with the system. Figure 6 shows the abuse cases with symbols when applied on college management system.

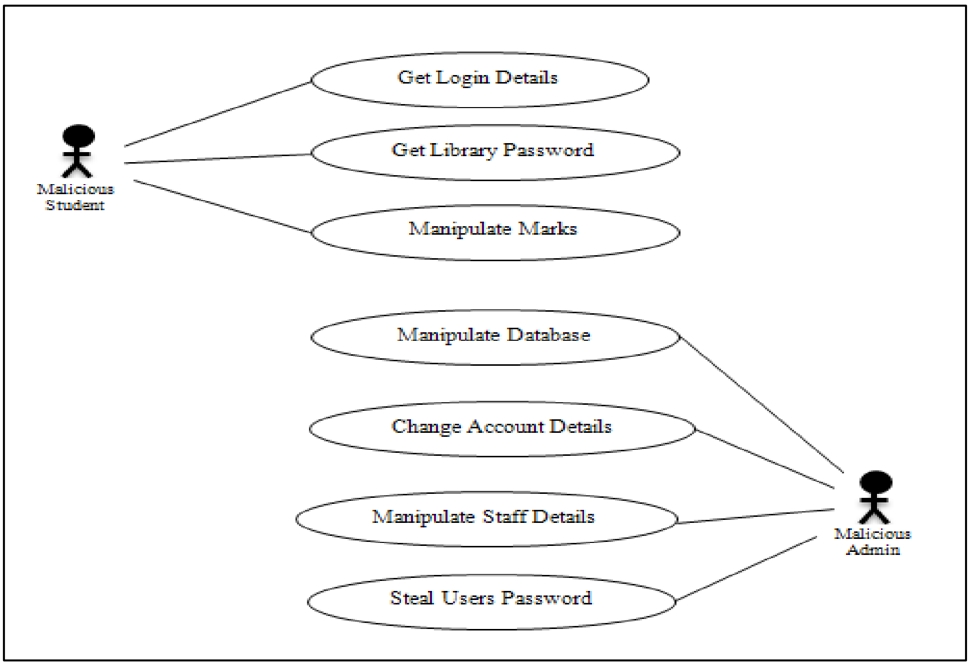


Figure 6: Abuse cases example for college management system

3.4.2 Common Criteria

This approach given by Ware et al. [21,22] relates the common criteria standard with the use case diagrams for the elicitation of Security Requirements. In this approach each actor is defined more formally than traditional practices with seven fields:

- Actor: represent the name of the actor.
- Use Case: name of associated use case with actor.
- Type: this can be human, cooperative or autonomous.
- Location: specifies the location of actor which can be local or remote.
- Private Exchange: it can be true or false depending on exchange of private info.
- Secret Exchange: it can also be true or false depending on information confidentiality needs.
- Association: specifies the association with use case.

Table 3: Actor profile in common criteria

Actor:	Name of actor
Use Case:	Associated Use Case
Type:	Type of actor (Human/Co-operative/Autonomous)
Location:	Actor Location (Local/Remote)
Private Exchange:	Information flowing is private or not (True/False)
Secret Exchange:	Confidentiality required for information (True/False)
Association:	Association with Use Case (Read/Write/Read_Write/Ask/Answer/ Ask_Answer)

Now based on actor profile data as shown in Table 3, threats are derived based on the relationship between actor and use case. After threats are identified, Security Requirements to counter each threat are established and Security Functionalities are specified to fulfill each objective.

3.4.3 Misuse Cases

Researchers [23,24] describe misuse cases as a 'special kind of use case, describing behavior that the system owner does not want to occur'. Hence misuse case is an extension of use case diagram in which misuse cases and mis-actors (one who initiate misuse cases) are shown in inverted format. The association between a misuse case and a use case shown in same diagram can either be a 'threaten' or 'mitigate'. The goal of misuse case finally is to prevent a threat from occurring or to mitigate the impact if it occurs [25]. Figure 7 shows the symbols and concept of misuse cases by taking an example of college management system.

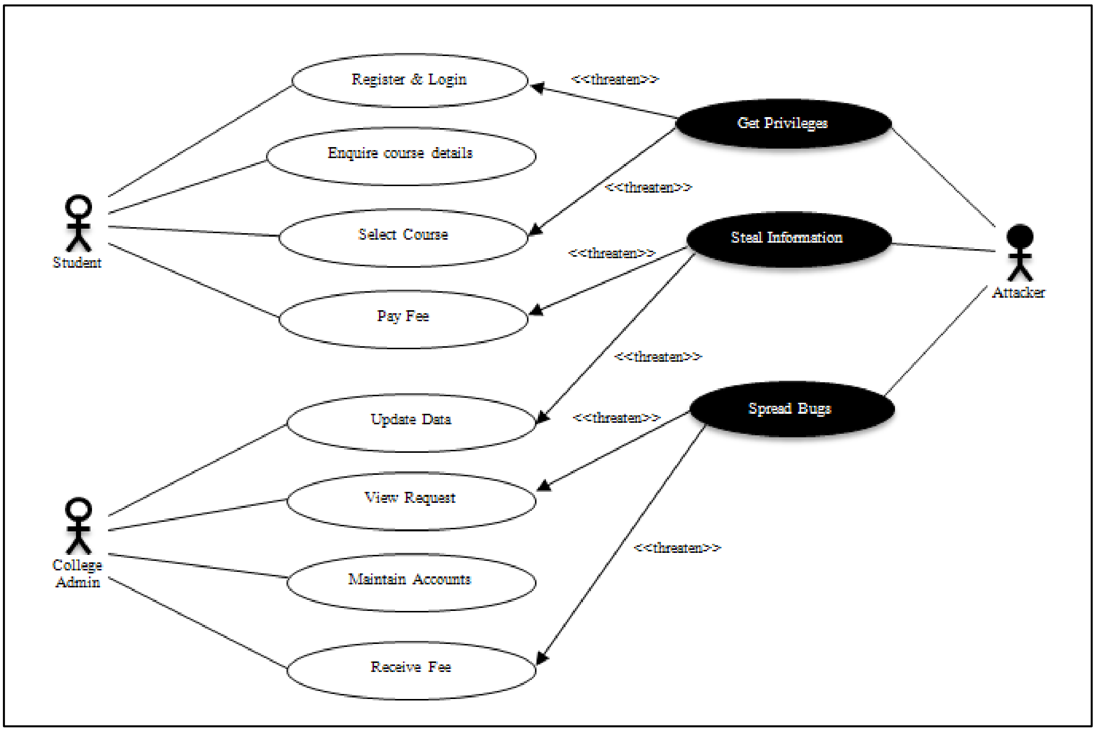


Figure 7: Misuse case example for college management system

3.4.4 Security Use Cases

According to Firesmith [6] misuse cases are effective way of analyzing threats but they are inappropriate during analysis and specification of Security Requirements, because the success criteria for misuse case is a successful attack against an application while the security use cases specifies requirements that the application shall successfully protect itself from relevant security threats. In this each Security Requirements like access control, integrity, non-repudiation etc. has its own security use case which provide natural organization to the use cases. An example of security use cases for college registration system is shown below for illustration. Figure 8 showing the concept of security use cases with an example of college management system.

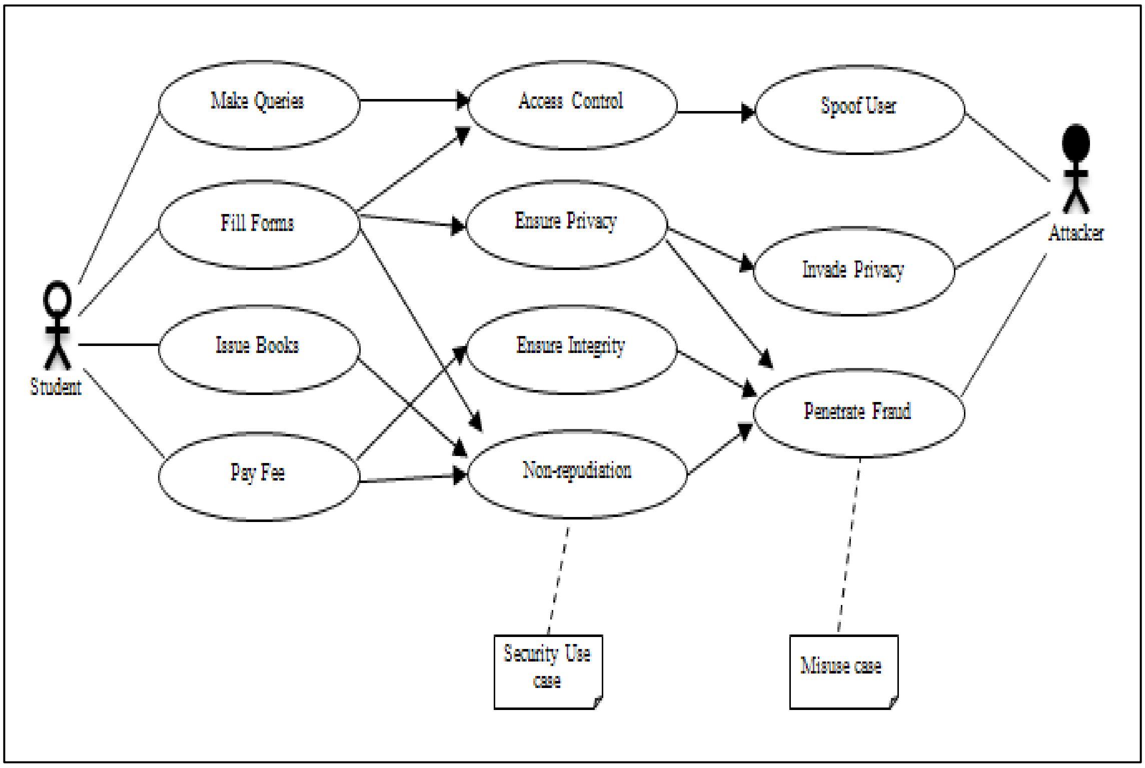


Figure 8: Security use case example for college management system

3.4.5 Attack Trees

Attack trees are introduced by Robert [26] to provide an approach to describe the security of a system. These are similar to threat trees, except that they enumerate all possible attacks of a single threat and so the whole system may have multiple attack trees with one for each threat present in a system. Attack trees uses the most widely used data structure 'Trees' for representation, where the root node of the tree is a goal the attacker wishes to achieve i.e. the 'threat'. And each node on tree, other than root node is a way to achieve the goal of its parent node as shown in Figure 9. This process can be done iteratively till an end leaf is reached which represents a single attack method to accomplish the goal of its parent. Satisfying a tree node means either satisfying all leaves (AND) or satisfying a single leaf (OR). The value of attack tree analysis is derived from the attributes associated with each node. Figure 9 shows the concept of attack tree by taking an example of college management system.

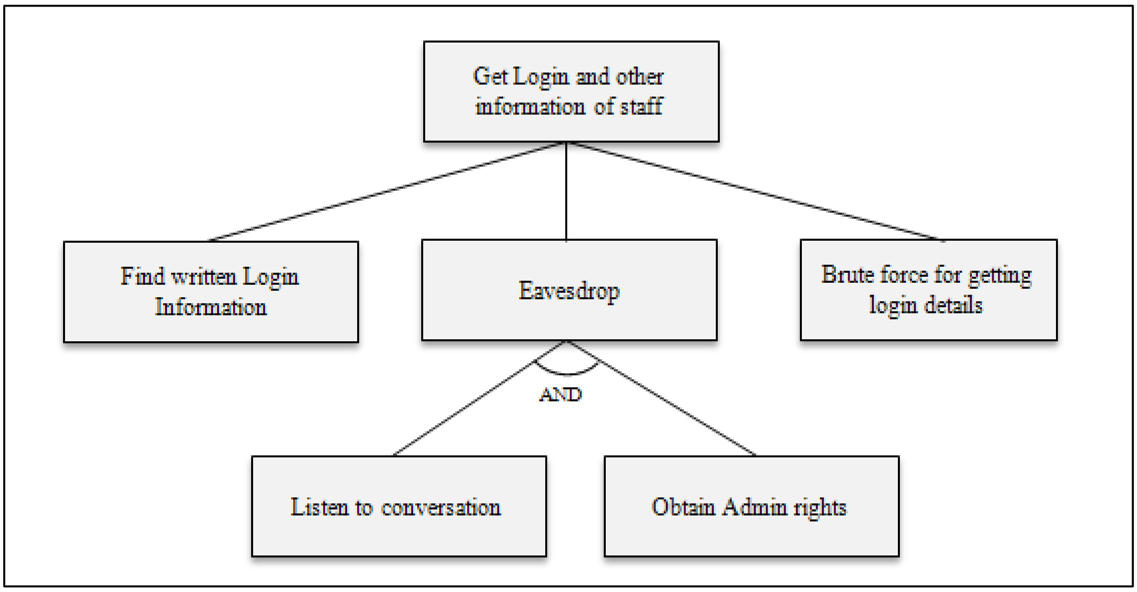

Get Login and other information of staff
Find written Login Information
Eavesdrop
Brute force for getting login details
AND
Listen to conversation
Obtain Admin rights

Figure 9: Attack tree example for college management system

3.4.6 Secure Tropos

Secure Tropos [27] is an extension of 'Tropos' methodology with security oriented features in it. Through this a developer can identify Security Requirements during software development process by employing various modeling activities. Secure Tropos includes following features:

- *Actors*: entities having strategic goals.
- *Soft-Goal*: goals which have no clear criteria for satisfaction.
- *Tasks*: is an abstract way of doing something.
- *Resources*: is a physical or informational entity.
- *Security Constraint*: are constraints like integrity, privacy, availability etc. of system under development.
- *Intentional Dependencies*: specifies that actors are interdependent to achieve some specific goal.
- *Secure Entities*: it involve any secure goal, tasks and resources of a system.
- *Secure Goal*: introduced to achieve every security constraints that are imposed on actors.
- *Secure Diagram*: constructed after analyzing security requirements of the system and its environment.
- *Secure Task*: a task that represents a way to achieve a secure goal.

Secure Tropos is an iterative process in which modeling activities are used to produce different kinds of actors and goal diagrams such that the diagrams produced by one activity are used as an input for other activities. Various activities involved in secure tropos for building a secure model of the system are:

- First Modeling: it's an actor modeling in which different stakeholders are identified.
- Second Modeling: it's a dependency modeling which consist of identifying actors that depends on one another for achieving a goal.
- Third Modeling: consist of goal modeling which focuses on actors and goal analysis. It includes three reasoning techniques i.e. means-end-analysis, contribution analysis and decomposition.
- Fourth Modeling: is a plan modeling which focuses on plan, similar to goal modeling it uses goal diagram for graphical representation.
- Last Modeling: is a capability modeling in which capabilities of sub-systems are specified.

3.4.7 Intentional Anti Model

Lamsweerde [28] defines anti-model which leads to the generation of more subtle threats and robust Security Requirements as countermeasure to security threats. He used 'anti-models' which are used to capture attackers, their goals, software vulnerability and attacks that satisfy their goals with 'anti-goals' which represents attackers own goal. This process derives an threat tree systematically through anti goal refinement until leaf nodes are reached that are either vulnerabilities exploited by attacker or anti requirement used by this attacker. A systematic procedure for building intentional anti-model is:

- Find initial anti-goal by negating relevant confidentiality, privacy, integrity and availability.

- For each identified anti-goal, find potential attackers agents that might own the goal.
- For each anti-goal and attackers identified, elicit the attacker's higher level anti-goals.
- Now form anti-goal AND/OR graph by refining anti-goals along alternative branches for deriving terminal anti-goals that are realized by the attacker agents.
- Derive the object and agent anti-models from anti-goal specifications.
- AND/OR operationalize all the anti requirements in terms of capabilities of the attacker agent.

Figure 10 shows how a threat tree is generated based on a threat related to college management system.

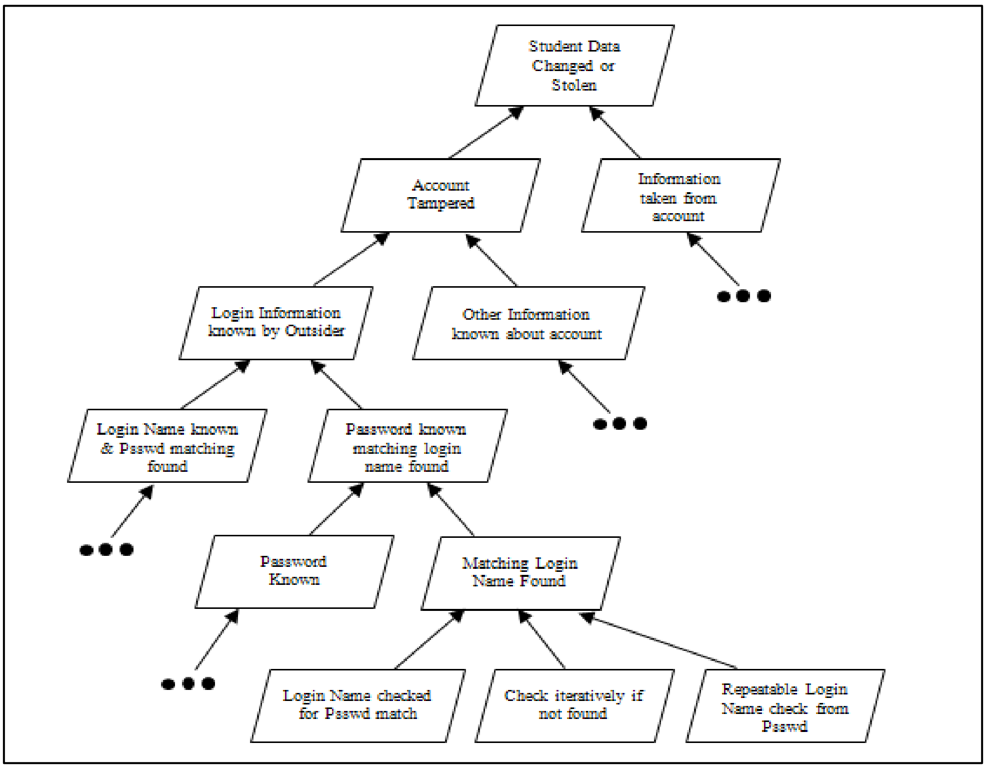

Student Data Changed or Stolen
Account Tampered
Information taken from account
Login Information known by Outsider
Other Information known about account
Login Name known & Psswd matching found
Password known matching login name found
Password Known
Matching Login Name Found
Login Name checked for Psswd match
Check iteratively if not found
Repeatable Login Name check from Psswd

Figure 10: Intentional anti model example for college management system

3.4.8 *View Point oriented Security Requirement elicitation process (VOSREP)*

VOSREP process defined by Agarwal & Gupta [7] helps in elicitation of Security Requirements with various techniques for activities like requirement discovery, analysis with prioritization and management. Various activities included are discussed below:

- *Requirement discovery and definition*: this is the initial activity which consists of following sub-activities:
    - Identification of Stakeholders using view point analysis as shown in Figure 11.
    - Identification of functionalities and non functional requirements related to each actor
    - Identification of Threats associated with functional requirements and assets.
    - Identification of Security Requirements to mitigate threats
- *Analysis and Prioritization of Requirements*: Analyze identified Security Requirements by removing ambiguities and prioritize them based on the risk associated with threats.
- *Management of Requirements*: Keep trace of each Security Requirement and its associated attribute as they are also subject to changes during the project.

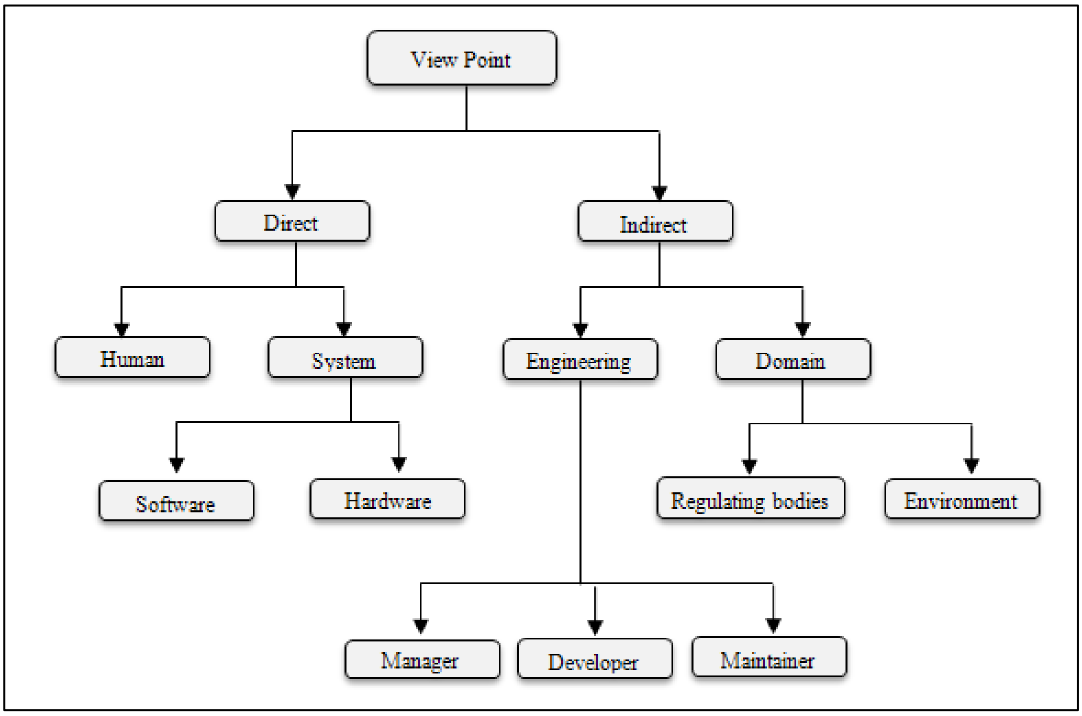


Figure 11: Various stakeholders based on viewpoints

3.5 Integration of Security Requirements Engineering in Conventional Spiral Model

As proposed by [7,8] integration of security mechanism in Software engineering process as shown in Figure 12 is modified version of conventional spiral model.

As in the conventional spiral model the process starts from the requirement discovery & definition where the developer will meet the client and collect all the information related to project development then analyze them and check feasibility of the project and if it is feasible. Then he will proceed and plan the further phases of the development. And follow all the successive steps accordingly. But in this process there is no consideration of Security Requirement in the early phase that is the requirement engineering phase that is the first phase it will consider all the constraint related to security at the end of development process. So handling of these at later phase will cost more and will sometime lead to over budgeting of the project.

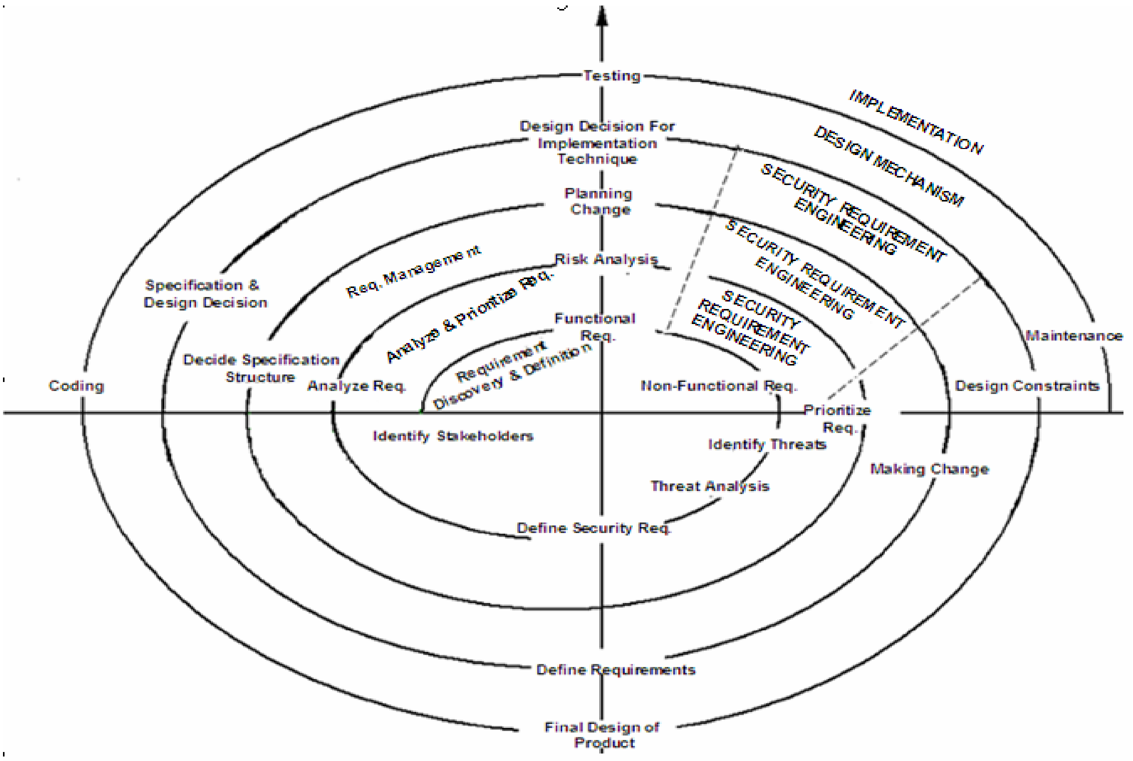


Figure 12: Different tasks in Software Engineering process [7]

So it will be better to incorporate Security Requirement in the beginning of the project. So [7,8] have modified the conventional spiral model to incorporate Security Requirement in it.

3.6 Security Design Engineering Framework

After Security Requirements Engineering the next phase in Security Engineering is Security Design Engineering according to the framework proposed by [9,10]. It consists of total four sub-steps as shown in Figure 13 which identifies the cryptographic services needed to mitigate the threats after the identification of Security Requirements. Various sub-steps involved are briefly discussed here for better understanding.

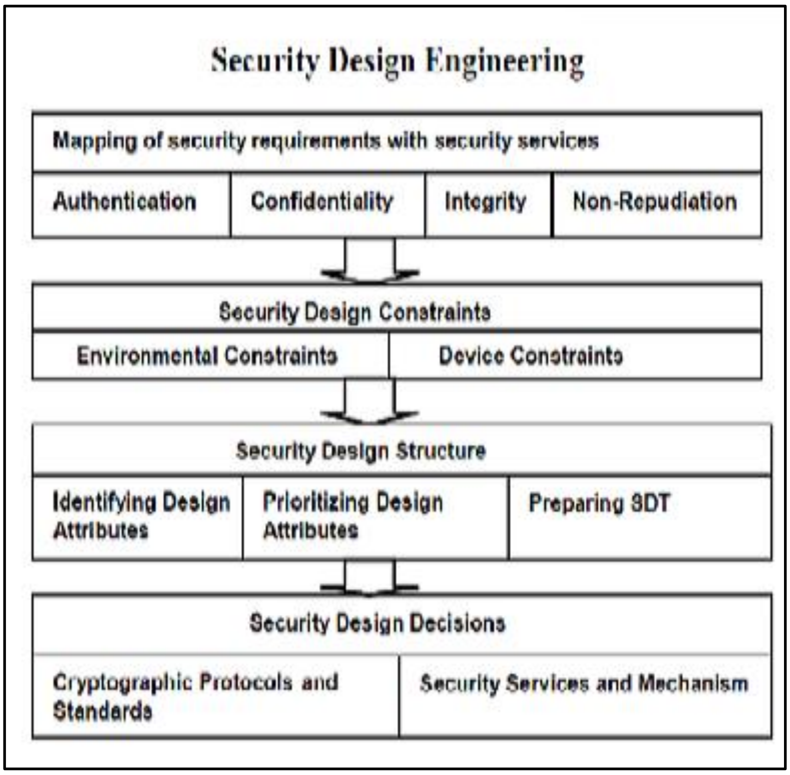

Security Design Engineering
Mapping of security requirements with security services
Authentication
Confidentiality
Integrity
Non-Repudiation
Security Design Constraints
Environmental Constraints
Device Constraints
Security Design Structure
Identifying Design Attributes
Prioritizing Design Attributes
Preparing SDT
Security Design Decisions
Cryptographic Protocols and Standards
Security Services and Mechanism

Figure 13: Security Design Framework [9]

- **Mapping of Security Requirements with Security Services**

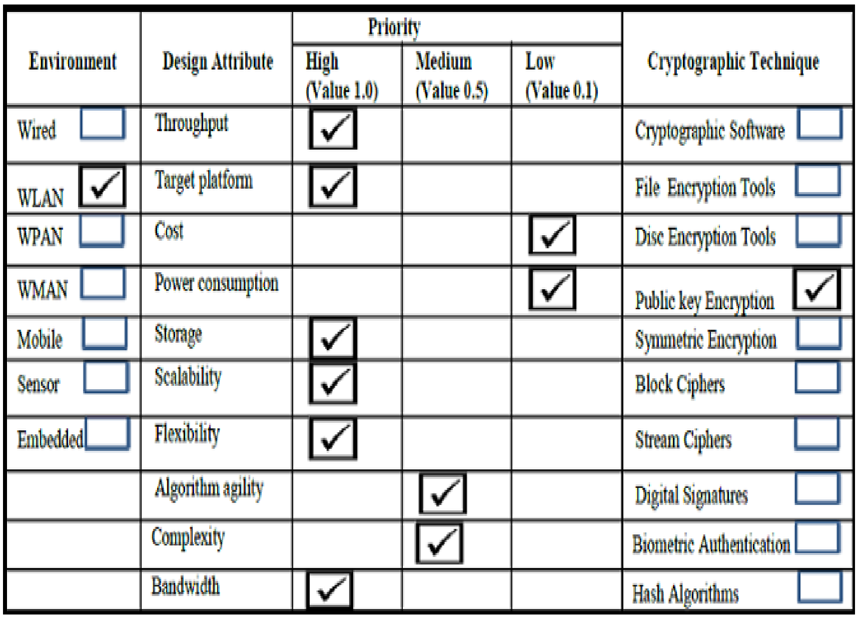
The prioritized Security Requirements obtained from Security Requirements Engineering phase are correspondingly mapped with the security services provided by cryptography. This would help in specifying which cryptography technique is suitable in particular scenario.

- **Security Design Constraints**

In this activity various constraints like environmental and device constraints are identified which are going to affect the selection of cryptographic techniques. For instance in web based system various constraints are power, bandwidth, memory etc.

➢ Security Design Structuring

Identification of different design attributes which affects the cryptographic protocols selection is performed in this step. First the design attributes are identified by observing the target platform and environments, after which they are prioritized on the basis of devices used as their priorities are different for Low-end devices and High-end devices. Secondly a Security Design Template (SDT) is prepared as shown in Figure 14 which contains specification of each design constraints and design attributes of specific environment.



Environment	Design Attribute	Priority			Cryptographic Technique
		High (Value 1.0)	Medium (Value 0.5)	Low (Value 0.1)	
Wired □	Throughput	☑			Cryptographic Software □
WLAN ☑	Target platform	☑			File Encryption Tools □
WPAN □	Cost			☑	Disc Encryption Tools □
WMAN □	Power consumption			☑	Public key Encryption ☑
Mobile □	Storage	☑			Symmetric Encryption □
Sensor □	Scalability	☑			Block Ciphers □
Embedded □	Flexibility	☑			Stream Ciphers □
	Algorithm agility		☑		Digital Signatures □
	Complexity		☑		Biometric Authentication □
	Bandwidth	☑			Hash Algorithms □

Figure 14: Security Design Template (SDT) [9]

➢ Security Design Decision

It is the final step in which the optimum security protocol is selected for system under consideration. This selection is made from a repository of various cryptographic techniques with their analytic attributes.

3.7 Security Testing Framework

The final phase of Security Engineering is the Security Testing which is done to test the proper implementation of Security Requirements which protects the system against all possible attacks. Gupta et al.[11] proposed an Security Testing framework consisting of four sub-steps as shown in Figure 15. Various sub-steps in this framework are discussed below in this section.

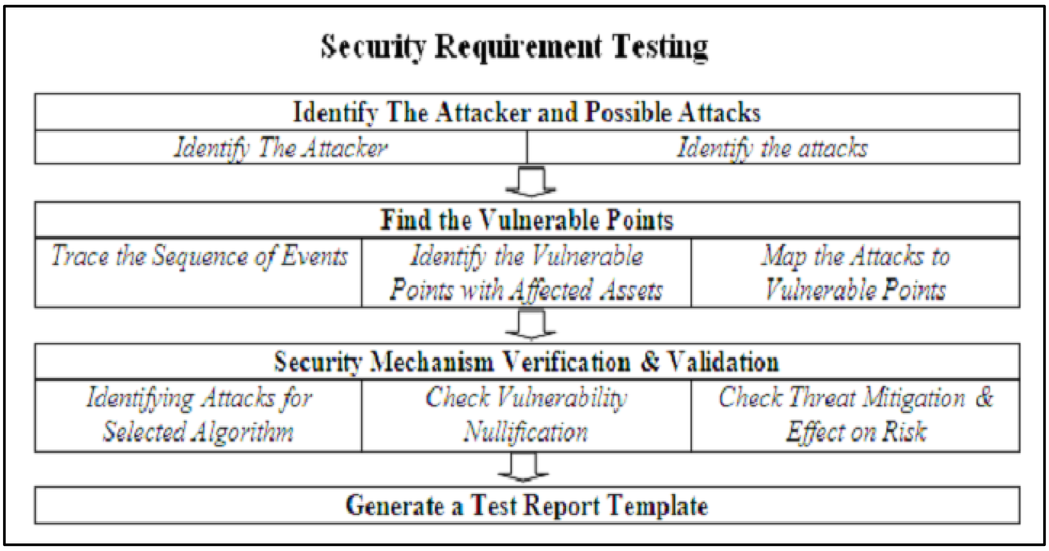

Security Requirement Testing
Identify The Attacker and Possible Attacks
Identify The Attacker
Identify the attacks
Find the Vulnerable Points
Trace the Sequence of Events
Identify the Vulnerable Points with Affected Assets
Map the Attacks to Vulnerable Points
Security Mechanism Verification & Validation
Identifying Attacks for Selected Algorithm
Check Vulnerability Nullification
Check Threat Mitigation & Effect on Risk
Generate a Test Report Template

Figure 15: Security Testing Framework [11]

➢ Identify the attackers and possible attacks

In this step first the attackers are identified for a system which may cause any kind of harm, and then possible attacks are identified from these attackers.

➢ Find Vulnerable points

The main objective of this step is the identification of vulnerable points in a system which an attacker uses. The first sub-step in this is to trace the sequence of events. Second sub-step is the identification of vulnerable points and the assets that may be affected in case of attack. Third sub-step is the mapping of attacks from step 1 to the identified vulnerable points in a system.

- **Security Mechanism Verification & Validation**

The cryptographic algorithms identified in Security Design phase are now tested to check whether they are able to protect against attack or threats possible due to the presence of vulnerable points. It consists of various activities like identifying attacks for selected cryptographic algorithms, checking for the vulnerability nullification and checking for threat mitigation.

- **Generate Test Report Template**

The final step in Security Testing is the generation of test report template which contains all information related to testing activities. This will further help the developer in selecting future activities. The test report template consist of various fields like Test Case ID, Name, Attacker with their attacks, Vulnerable point etc.

Chapter 4

FRAMEWORK FOR SECURITY REQUIREMENTS ENGINEERING IN CLOUD SYSTEM

After establishing the foundation of Security Requirements Engineering we now present our Security Requirements elicitation framework for Cloud system which is based on the following observations concluded from the forgoing sections:

- Security issues are the top most concern in Cloud and should be considered early during SDLC.
- Security Requirements are driven from functionalities and data which are accessed and used by user of the system which may be internal or external to the system.
- Security Requirements are related to each other. For e.g. authorization requirements requires existence of identification and authentication requirements both.

We have considered the widely used 'Storage-as-a-Service' model of Cloud computing as basis for Security Requirements elicitation process as Cloud itself offer various type of services in which the architecture and roles involved varies from model to model. So before we dwell into the approach, a concise overview of Cloud Storage-as-a-service is given to help a novice.

In Cloud Storage-as-a-service model the service provider rents the digital storage available on their own data centers accessible over an internet, on subscription basis.

Basic features available in Cloud Storage services are copy, backup, synchronization and file sharing. Some popular available Storage services are Google Drive, DropBox, CloudMe, Mozy etc. The most basic view of Cloud storage service is shown in Figure 16.

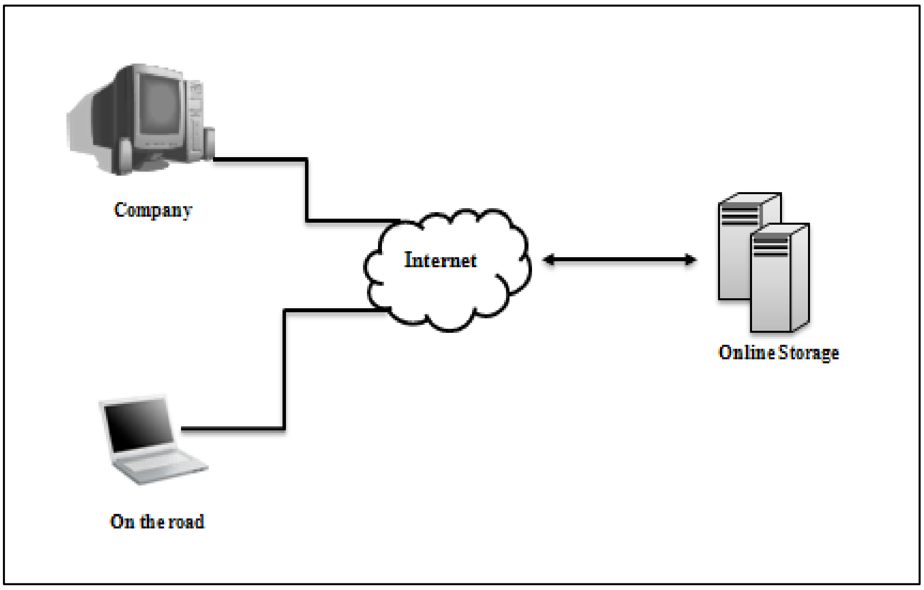


Figure 16: Basic view of Cloud Storage service

Security Engineering process consists of four activities [9,10] i.e. Security Requirements Engineering, Security Design Engineering, Security Requirements Implementation and Security Requirements Testing and we are focusing only on the first phase i.e. Security Requirements Engineering phase in our framework, as we feel that it is the most important phase of Security Engineering as it feeds to other phases and any mistake in this phase may introduce vulnerabilities in the system which can be exploited by attackers.

In the rest of this chapter we are going to discuss in detail our proposed framework for Security Requirements Engineering in step-by-step manner for clear understanding.

4.1 Framework Overview

This framework is an extension of View point oriented Security Requirements elicitation approach [7] which is applied on a Cloud system. Basically our framework consists of total five steps which need to be executed in sequential order as shown in Figure 17 for eliciting true Security Requirements with their Security functionalities in Cloud system.

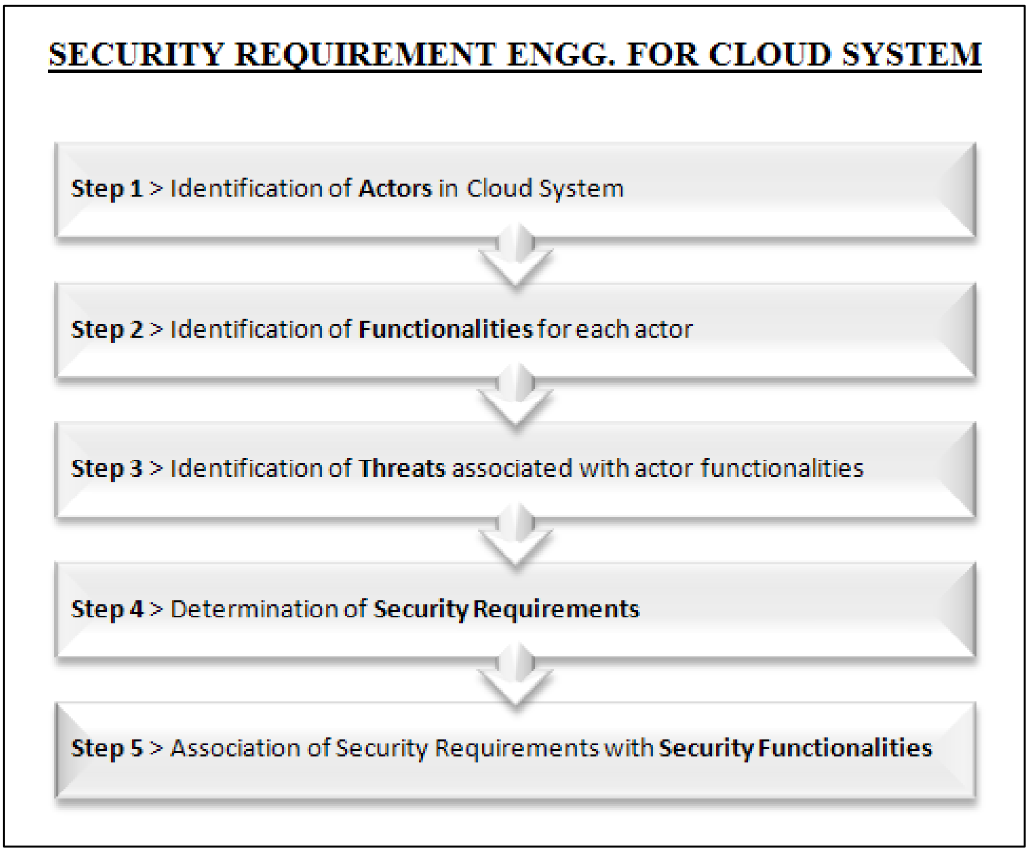

SECURITY REQUIREMENT ENGG. FOR CLOUD SYSTEM
Step 1 > Identification of Actors in Cloud System
Step 2 > Identification of Functionalities for each actor
Step 3 > Identification of Threats associated with actor functionalities
Step 4 > Determination of Security Requirements
Step 5 > Association of Security Requirements with Security Functionalities

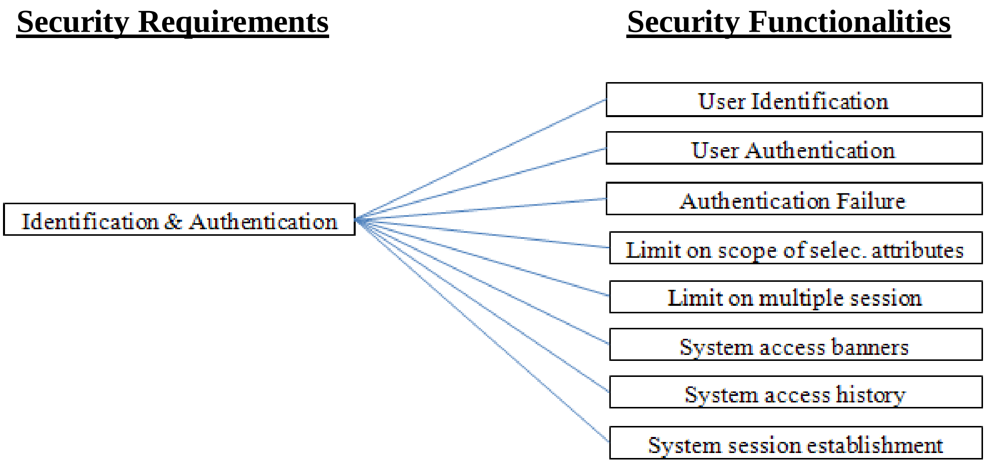
Figure 17: Framework for Security Requirements Engineering in Cloud System

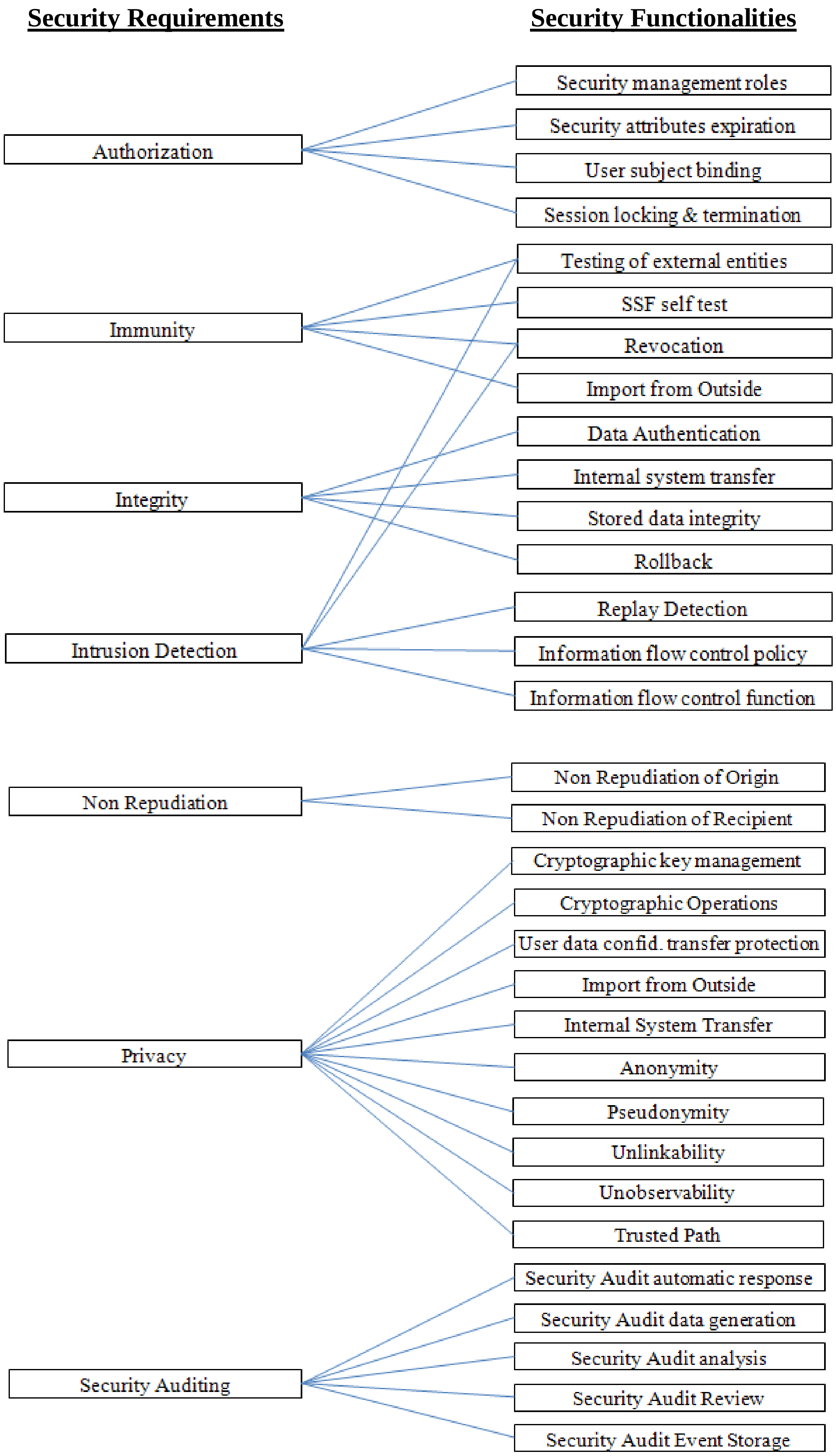
The first step in our framework is the identification of ‘Actors’ in Cloud system. These actors can be either direct actors (one who directly interact with the system) or indirect actors (one who regulates the application domain). In second step the functionalities (functional requirements) are identified for each actor and correspondingly an Actor profile is created which consists of seven fields. Now in third step the 'Threats' possible in Cloud system are identified based on the actor profile created in step two. After that in step four we determine the possible Security Requirements needed to mitigate the above identified threats. Then finally in step five we associate each Security Requirements with the Security Functionalities defined by Common Criteria [29].

Now these identified Security Requirements can be prioritize & managed [7,8] and finally passed over to the Security Design phase so that proper design decisions based on these Security Requirements should be taken early in the SDLC to create a secure Cloud System.

4.2 Security Requirements with associated Functionalities

Before we explain each step of our framework in detail, we first show the broad association as shown in Figure 18 and then discuss various Security Requirements and their corresponding Security Functionalities involved. These set of Functionalities have been borrowed from the set of Common Criteria [29] which helps Security Design Engineers to take better decision on Security Requirements.




Security Requirements
Security Functionalities
Authorization
Security management roles
Security attributes expiration
User subject binding
Session locking & termination
Immunity
Testing of external entities
SSF self test
Revocation
Import from Outside
Integrity
Data Authentication
Internal system transfer
Stored data integrity
Rollback
Intrusion Detection
Replay Detection
Information flow control policy
Information flow control function
Non Repudiation
Non Repudiation of Origin
Non Repudiation of Recipient
Privacy
Cryptographic key management
Cryptographic Operations
User data confid. transfer protection
Import from Outside
Internal System Transfer
Anonymity
Pseudonymity
Unlinkability
Unobservability
Trusted Path
Security Auditing
Security Audit automatic response
Security Audit data generation
Security Audit analysis
Security Audit Review
Security Audit Event Storage

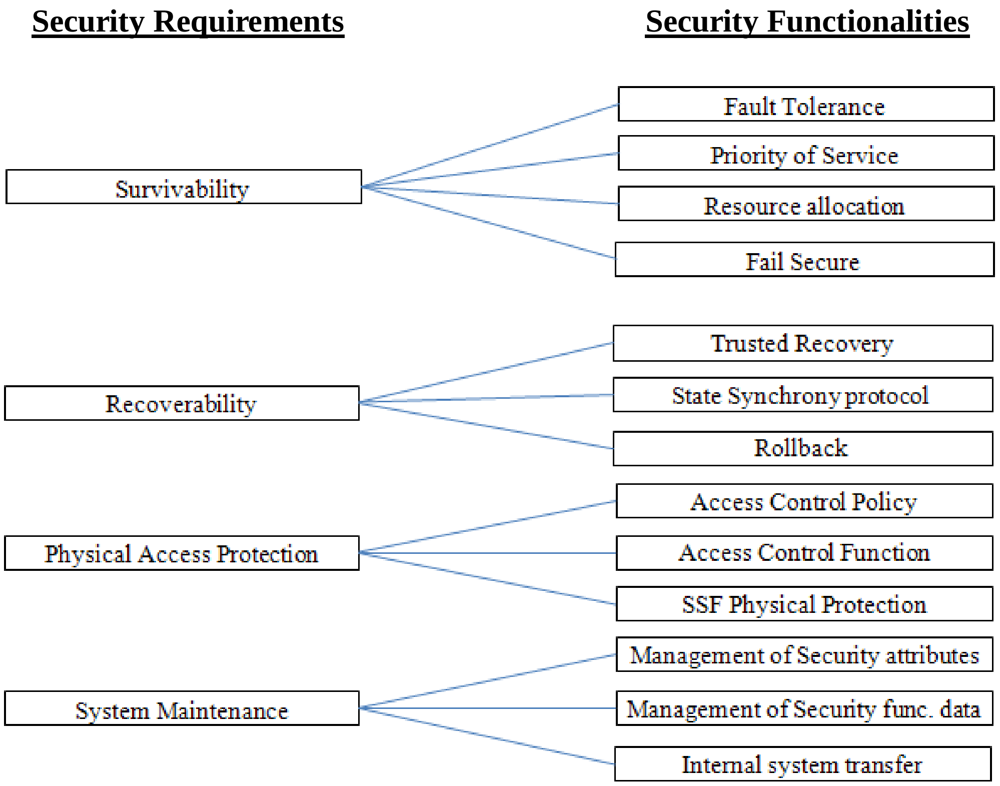


Figure 18: Security Requirements and Functionalities association

The detailed description about these Security Requirements with the associated Security Functionalities is given below:

➢ Identification & Authentication

Identification requirement specifies the extent to which a Cloud system shall identify its external users before interacting them, like applying turing test etc. Similarly authentication requirement is used to verify the identity of externals what they claims to be before interacting by using attributes like User ID or Password etc.

Associated functionalities with this Security Requirement are:

- *User Identification (UID)*: It defines conditions based on which users are required to identify themselves before performing any action which demands user interaction. E.g.
  - 'Cloud Storage shall apply turing test on customers and users.' Or
  - 'Cloud Storage shall use multidimensional attributes for identification.'
- *User Authentication (UAU)*: It defines the user authentication mechanism and required attributes on which the mechanism must be based. E.g.
  - 'Cloud Storage shall use combination of UserID and Password for Authentication.' Or
  - 'Cloud should use biometrics attributes for Authentication.'
- *Authentication Failures (AFL)*: This requirement deals with defining some limit on unsuccessful authentication attempts or necessary action when authentication fails. E.g.
  - 'Cloud Storage should not allow more than three consecutive wrong authentication attempts.' Or
  - 'Cloud shall notify the security administrator in case of consecutive wrong attempts.'
- *Limitation on scope of selectable attributes (LSA)*: This requirement applies a limit on the scope of security attributes related to a session that a user select during a session establishment. E.g.
  - 'The Cloud Storage shall not allow three months old password to gain access.'
  - 'Cloud shall regularly change the key size used for encrypt transfer.'

- *Limitation on multiple concurrent sessions (MCS)*: This Security Requirement applies a limit on the number of session occurring concurrently that belongs to a same user. E.g.
    - 'Cloud Storage shall only allow single session at a time from same user.'
    - 'Cloud shall record locations used during multiple session from same user.'
- *System access banners (TAB)*: This requirement specifies the need to display advisory warning related to system use before a session to the users. E.g.
    - 'Cloud shall regularly notify customers to change login details.'
    - 'Cloud shall warn customers about remaining unsuccessful attempts.'
- *System access history (TAH)*: This requirement deals with the need to display a history of successful and unsuccessful attempts to user's account after the establishment of a session. E.g.
    - 'Cloud Storage shall display last login attempts to all customers.'
    - 'Cloud Storage should maintain archives of last 1 month login attempts details.'
- *System session establishment (TSE)*: It deals with accepting or denying a users request for session establishment based on certain attributes. E.g.
    - 'Cloud Storage shall temporarily suspend services for specific customer in case of doubtful login attempt.'
    - 'Cloud Storage allow session establishment only after verifying every security attribute related to customer.'

➢ Authorization

Authorization requirement specifies the extent to which the Cloud system verifies the usage privileges and access restrictions of authenticated users and application. This requirement prevents unauthorized users from obtaining access to inappropriate data or services.

Associated functionalities with this Security Requirement are:

- *Security management roles (SMR)*: It deals with the control over the assignment of different roles to various users in a system. E.g.
    - ‘Cloud Storage shall have predefined roles for employees.’
    - ‘Cloud shall impose restrictions based on roles.’
- *Security Attribute Expiration (SAE)*: This requirement deals with the assignment of time limits for the validity of various security attributes.
    - ‘Cloud Storage shall terminate session if unattended for 10 minutes’
- *User-Subject Binding (USB)*: This requirement creates and associate user’s security attributes, totally or partially to a subject which acts on user’s behalf. E.g.
    - ‘Cloud Storage shall relate the device used to gain access with UserID.’
    - ‘Cloud shall verify customers device also before establishing sessions.’
- *Session locking and termination (SSL)*: This requirement deals with the capability of user initiated locking, unlocking and termination of session.
    - ‘Cloud Storage should provide function to temporarily suspend services to customers.’

- **Immunity**

Immunity requirement specifies the extent to which a Cloud system shall protect itself from undesirable programs like viruses, worms etc. from destroying or damaging the data and system applications.

Associated functionalities with Immunity Security Requirement are:

- *Testing of external entities (TEE)*: This requirement allows the SSF to test one or more external entities (like applications running on system, hardware, software etc.). E.g.
  - ‘Cloud Storage shall test critical devices before communication.’
  - ‘Cloud shall report security administrator if error detected.’
- *SSF self test (TST)*: It deals with the self testing of SSF by the system, which can be performed at startup, periodically or at request from authorized user. E.g.
  - ‘Cloud Storage should test its virus chest for updates periodically.’
  - ‘Cloud shall test connection between devices after startup.’
- *Revocation (REV)*: This requirement deals with the security attributes revocation for variety of entities within a system. E.g.
  - ‘Cloud Storage should delete information of customers who unsubscribe services.’
  - ‘Cloud Storage shall scrap left over employees access credentials.’
- *Import from Outside (ITC)*: This requirement defines the mechanism for either protecting security attributes or not for a user data when importing into the system from outside. E.g.

- ‘Cloud Storage shall allow customers to select encryption method at client side before uploading their data.’

➢ **Integrity**

Integrity requirement specifies that a Cloud shall protect its data and communication from any unauthorized modification or deletion. Its main objective is to ensure that the communication and data can be trusted.

Associated functionalities with Integrity Security Requirement are:

- *Data Authentication (DAU)*: Data authentication means that an entity is responsible for the authenticity of information. This requirement specifies a method to verify the authenticity of static data, that the content has not been forged. E.g.
  - ‘Cloud Storage shall verify the owner of data before storing.’
  - ‘Cloud shall immediately remove unauthentic data.’
- Internal System Transfer (ITT): It specifies the extent to which user data is protected when it is transferred between various parts of a system through internal channels. E.g.
  - ‘Cloud Storage should use reliable channel for internal transfer of data.’
- *Stored Data Integrity (SDI)*: It specifies the protection of user critical data while it is stored within the boundary of a system. It differs from ITT which protects the integrity of user data while being transferred within a system. E.g.

  - ‘Cloud Storage shall regularly check the stored customer data for any unauthorized modification.’

- *Rollback (ROL)*: This requirement provides the ability to undo the effect of an operation in a system to ensure the integrity of user data. E.g.
  - ‘Cloud Storage shall preserve the status of data in timely manner.’ Or
  - ‘Cloud Storage shall restore the previous state of customer data if unauthorized modification detected.’

➢ Intrusion Detection

Intrusion requirement specifies the extent to which a Cloud shall perform detection and recording of intrusions or modifications by unauthorized or authorized persons. It may include potential response activities like security alarms etc. in case of intrusion.

Associated functionalities with this Security Requirement are:

- Replay detection (RPL): This requirement deals with the detection and prevention of replay actions from various types of entities. E.g.
  - ‘Cloud Storage shall discard multiple requests within short time to prevent Denial of Services.’
- Testing of external entities (TEE): This requirement allows the SSF to test one or more external entities (like applications running on system, hardware, software etc.). E.g.
  - ‘‘Cloud Storage shall test customer device after session start.’
  - ‘Cloud shall immediately discard device if error detected.’

- Revocation (REV): This requirement deals with the security attributes revocation for variety of entities within a system. E.g.
  - ‘Cloud Storage shall discard access credentials of employees after they left.’
- Information flow control policy (IFC): This requirement identifies the information flow control policy and also defines the scope of control for each information flow control by identifying the subject, information and operation under control of policy. For e.g.
  - ‘Cloud Storage shall select reliable path for information flow.’
- *Information flow control functions (IFF)*: It describes the rules related to specific function that implements the information flow control policy identified in IFC. E.g.
  - ‘Cloud Storage shall test all nodes before actual transfer begins.’

➢ Non Repudiation

Non repudiation security requirement specifies that Cloud shall prevent the sender and receiver from denying their involvement in communication at later stage. Its objective is to maintain records about critical involvement of customers to prevent them from denying.

Associated functionalities with Non-Repudiation Security Requirements are:

- *Non-repudiation of Origin (NRO)*: This requirement specifies that the originator of information cannot deny after sending the information. It requires a method to provide evidence of the origin to receiver. E.g.
  - ‘Cloud Storage shall store information of every data upload action.’

- *Non-repudiation of Recipient (NRR)*: It specifies that the recipient of information cannot deny after receiving information. It requires a method to provide evidence of receipt to the sender. E.g.
    - ‘Cloud Storage shall store recipient for every data downloaded.’

➢ Privacy

Privacy security requirement specifies the extent to which Cloud system shall protect the customer’s critical data stored on its server or during communication from any unauthorized person or attackers. Its objective is the protection customer’s data, identity and actions so that it became unobservable to others.

Associated functionalities with Privacy Security Requirement are:

- *Cryptographic Key Management (CKM)*: These requirements specify that the cryptographic keys must be properly managed throughout its life cycle like key generation, distribution and destruction. E.g.
    - ‘Cloud Storage shall transfer the encryption keys to customer through secure channel.’
- *Cryptographic Operations (COP)*: It specifies that the cryptographic operations must be implemented in accordance with a specified algorithm and key size. Various cryptographic operations are encryption / decryption, digital signature verification, checksum generation and verification etc. E.g.
    - ‘Cloud Storage should not public the algorithm used for encryption.’

- *User data Confidentiality Transfer Protection (UCT)*: This requirement ensures the confidentiality of user data when it is transferred from a system to another product using an external channel. E.g.
  - ‘Cloud Storage should always send encrypted data to customers.’
- *Import from Outside (ITC):* This requirement defines the mechanism for either protecting security attributes or not for a user data when importing into the system from outside. E.g.
  - ‘Cloud Storage shall allow customers to select encryption method at client side before uploading their data.’
- *Internal System Transfer (ITT)*: It specifies the extent to which user data is protected when it is transferred between various parts of a system through internal channels. E.g.
  - ‘Cloud Storage should also encrypt data when transferred internally.’
- *Anonymity (ANO)*: It ensures that a user without disclosing its identity may use a resource or services in a system. E.g.
  - ‘Cloud Storage shall verify identity in encrypted form.’
- *Pseudonymity (PSE)*: This requirement specifies that a user without disclosing its identity may use a resource or service, but can still be accountable for that. E.g.
  - ‘Cloud Storage shall record all usage details of customers in encrypted form.’
- *Unlinkability (UNL)*: It ensures that multiple use of resources and services are allowed to a user, such that others are unable to link these uses together. E.g.
  - ‘Cloud Storage should encrypt the service usage links of customers.’

- *Unobservability (UNO)*: This requirement ensures that a user may use resources and services, such that others are not able to observe this utilization. E.g.
  - ‘Cloud Storage should hide the customer usage pattern from others.
- *Trusted Path (TRP)*: This requirement expresses the need to implement and maintain a trusted communication between user and the system. E.g.
  - ‘Cloud Storage shall selected secure nodes in a path to customer.’

➢ **Security Auditing**

Auditing security requirement specifies the extent to which Cloud system shall allow security auditors to inspect the system behavior and status from security point of view.

Associated functionalities with Security Auditing Requirement are:

- *Security Audit Automatic Response (ARP)*: This requirement defines the response that should be taken in case of security violation. E.g.
  - ‘Cloud Storage should terminate session in case of violation.’
- *Security Audit Data Generation (GEN)*: This requirement defines the need for recording the occurrence of security events by the system. E.g.
  - ‘Cloud Storage shall continuously record all unsuccessful attempts.’
- *Security Audit Analysis (SAA)*: It defines the need for automated monitoring of system activities and audit data for identifying security violation in system. E.g.
  - ‘Cloud Storage shall regularly monitor access details of customers.’

- *Security Audit Review (SAR)*: It specifies the need of audit tools that should be available to authorized users only to help them in reviewing audit data. E.g.
    - 'Cloud Storage shall have automated tools for analyzing large audit data.'
- *Security Audit Event Storage (STG)*: It says that the system should be able to create and maintain a secure audit trail which guarantees availability of audit data. E.g.
    - 'Cloud Storage shall have feature to generate audit data when needed by Auditor.'

➢ **Survivability**

Survivability requirement specifies that a Cloud system shall provide the basic functionalities or either fails gracefully even when some components or devices have been destroyed intentionally or naturally. Its objective is to survive the intentional component destruction.

Associated functionalities with Survivability Security Requirement are:

- *Fault Tolerance (FLT)*: This requirement ensures that the system will provide basic functionality in the event of failure also. E.g.
    - 'Cloud Storage have backup ready for providing basic services.'
- *Priority of service (PRS)*: It specifies that resources with high priority will always be accomplished without any delay caused by low priority activities. E.g.

    - ‘Cloud Storage shall protect customer data first in the event of security attack.’
- *Resource allocation (RSA)*: This requirement allows the system to control the resource utilization such that denial of service will never occur. E.g.
    - ‘Cloud Storage shall not allow full load on its data servers.’
- *Fail Secure (FLS)*: This requirement ensures that the system will always enforce its security requirements in the event of failure as identified in the SSF. E.g.
    - ‘Cloud Storage shall immediately notify the security administrator to replace device if it fails.’

➢ **Recoverability**

Recoverability security requirement specifies the extent to which Cloud system shall recover the data and system after the failure happens. Data recoverability deals with data correction after authorized or unauthorized modification whereas system recoverability specifies that a system recovers to a secure state after failure or modification.

Associated functionalities with Recoverability Security Requirement are:

- *Trusted recovery (RCV)*: This requirement specifies that the SSF can successfully recover the system after discontinuity of operations. E.g.
    - ‘Cloud Storage shall have proper mechanism to recover customer data in the event of failure.’

- *State synchrony protocol (SSP)*: This requirement ensures that various parts of a system have properly synchronized their states after some security related action in a Cloud system. E.g.
    - ‘Cloud Storage shall synchronize their devices after recovery from security attack.’
- *Rollback (ROL)*: This requirement provides the ability to undo the effect of an operation in a system to ensure the integrity of user data. E.g.
    - ‘Cloud Storage restores data to its previous state if unauthorized modification is detected.’

➢ **Physical Access Protection**

Physical access protection requirement specifies that a Cloud system shall protect its data centers and itself from unauthorized physical access, damage, theft, hardware replacement or sabotage.

Associated functionalities with Access Protection Security Requirement are:

- *Access Control Policy (ACC)*: This requirement identifies the access control policy and defines the scope of control of the policies on the object. E.g.
    - ‘Cloud Storage shall cover every employee in access control policy to data centers.’
- *Access Control Function (ACF)*: This requirement describes the rules related to specific functions that are implemented by the access control policy defined by ACC. E.g.

  - ‘Cloud Storage shall allow only limited employees to enter inside data centers.’

- *SSF physical protection (PHP)*: This requirement specifies the restriction applied on unauthorized physical access and physical modification to the SSF. E.g.
  - ‘Cloud Storage shall immediately raise alarm if security breach is detected in data centers.’

➢ **System Maintenance**

This security requirement specifies the extent to which Cloud system shall protect itself from any accidental authorized modification during maintenance or updates. It also includes management of security features and attributes.

Associated functionalities with System Maintenance Security Requirement are:

- *Management of Security Attributes (MSA)*: This requirement allows the control over the management of security attributes by an authorized users like modifying attributes or viewing etc. E.g.
  - ‘Cloud Storage shall not allow any employee to view customer data.’
- *Management of Security Functionality Data (MTD)*: It allows the control over the management of security functionality data like audit information and configuration parameters by an authorized user. E.g.
  - ‘Cloud Storage shall encrypt audit data and allow only authorized Auditors to decrypt them.’

- *Internal System Transfer (ITT)*: It specifies the extent to which user data is protected when it is transferred between various parts of a system through internal channels. E.g.
    - ‘Cloud Storage should also encrypt data when transferred internally.’
    - ‘Cloud shall distribute keys to authorized persons to decrypt security functionalities data.’

Now these Security Requirements will be going to act as our repository from which we have to elicit true Security Requirements based on the actor profile. In the rest of this chapter we are going to discuss in detail our proposed framework for Security Requirements Engineering in step-by-step manner when applied on Cloud-Storage-as-a-Service model for clear understanding.

4.3 Framework for Security Requirements Engineering in Cloud System

4.3.1 Identification of Actors in Cloud System *(Step 1)*

This is the foremost step of our framework which involves identification of Actors. According to VOSREP [7] process actors can be classified as direct and indirect actors. Direct actors are those who directly interact with the system like humans, software system and hardware etc. whereas indirect actors are personnel who regulate the application domain.

Direct actors for Storage-as-a-Service model are Cloud Customer, Cloud User, Cloud Service Provider and Cloud Service Integrator as shown in Figure 19:

- ***Cloud Customer***: Cloud customer is the one who stores data over the Cloud and pay for the services provided by CSP.

- ***Cloud User***: Cloud user is the one who have access to view the data shared by Cloud customer. It can be a subscriber or non subscriber of Cloud services.
- ***Cloud Service Provider (CSP)***: CSP is the one who owns, manage and operate the Cloud system to deliver services. It also receives the payment from Cloud customers for service provided.
- ***Cloud Service Integrator***: Integrator supplies business and IT services to others by integrating Cloud and other services in a transparent way, regardless of where those services are coming from.

Indirect actors for Storage as a Service model are ***Security Administrator*** (who maintains security related functions) and ***Auditor*** (who manages the audit and log details).

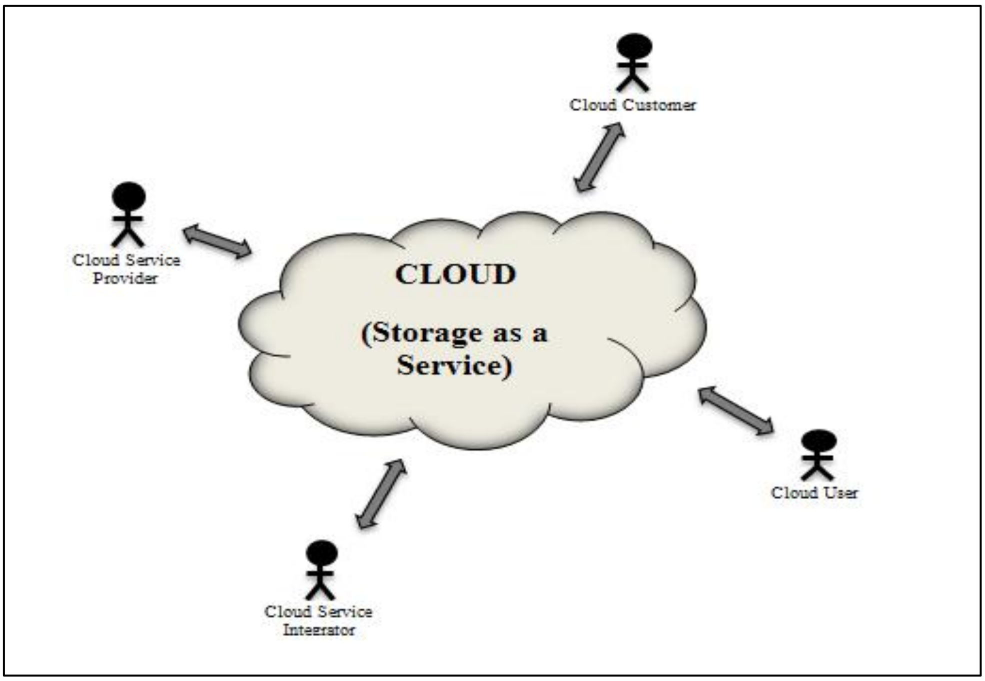


Figure 19: Various actors in Storage as a service

4.3.2 Identification of Functionalities for each Actor *(Step 2)*

In this step functionalities related to each actor identified in previous step are defined which finally helps in the elicitation of true Security Requirements. For e.g. functionalities related to Cloud Customer are: store data into Cloud, download data from Cloud, make payment etc.

Also the actor profile for each identified actor is populated which consists of seven fields i.e. Actor name, functionality, type, location, private exchange, secret exchange and association as shown in Table 4, which further helps in identification of threats related to each actor.

Table 4: Actor profile according to VOSREP [7]

Actor:	Name of actor
Functionality:	Functional Requirement
Type:	Type of actor (Direct/Indirect)
Location:	Actor Location (Local/Remote)
Private Exchange:	Information flowing is private or not (True/False)
Secret Exchange:	Confidentiality required for information (True/False)
Association:	Association with System (Read/Write/Read_Write/Ask/Answer/Ask_Answer)

Hence the identified functionalities related to direct actors (Cloud customer, Cloud users, Cloud service provider and Cloud service integrator) and indirect actors (Security Administrator and Auditor) and listed in tabular form as shown in Table 5.

Table 5: Identified functionalities of Actors

Actors	Functionality
Cloud Customer	1. Registration & Login. 2. Update Login details 3. Store data into cloud. 4. Manage automatic backup. 5. Manage sharing with cloud users. 6. Download data stored in cloud. 7. Select storage location. 8. Make payment for services used.
Cloud Users	1. Registration & Login. 2. View shared data based on permission. 3. Submit request to join group. 4. Unjoin a group.
Cloud Service Provider	1. Manage Cloud customer's account. 2. Manage Customer data. 3. Manage cloud hardware's & softwares. 4. Receive cloud usage payment. 5. Maintain SLA.
Cloud Service Integrator	1. Registration & Login. 2. Combines cloud based & in-house services.
Security Administrator	1. Take action on security breaches. 2. Apply security patches to software. 3. Update system with antiviruses and firewalls.
Auditor	1. Maintain the audit and log details. 2. Release audit reports.

4.3.3 Identification of Threats associated with Actor functionalities *(Step 3)*

For the identification of threats in Storage-as-a-Service model we have extended the predefined repository [7] by adding threats related to Cloud system. Hence various categories of threats related to Cloud System as identified in this step are shown in Table 6.

Table 6: Various threats in Cloud system

Sl. #	Threats	Description
1	Data_Theft	Stealing of data for personal use
2	Disclose_Data	Disclosing of information to unauthorized user while storing or processing
3	DoS_DDoS	Result in unavailability of services due to large number of requests
4	Human_Error	Is an accidental deletion or modification of data by employees
5	Impersonate	Making unauthorized access by impersonating an authorized user
6	Insider	An authorized user may gain unauthorized access to data
7	Malicious_Code	It includes execution of viruses, worms and Trojan horses etc.
8	Multilocation_DataPlacement	Customer critical data is stored at multiple locations in various countries
9	Natural_Disaster	It can be a force of nature like earthquake, flood, fire etc.
10	Outsider	Individual who is external and not authorized may gain access to data centers
11	Psswd_Cracking	Process of reverse calculating a password by brute-force, guessing or dictionary attacks
12	Change_Data	Deliberate act to modify the data
13	Repudiate_Receive	Entity may deny after receiving data
14	Repudiate_Send	Entity may deny that it has send data
15	Sabotage	Deliberate action to weaken another entity by destruction or disruption
16	Scavenging	Acquisition of left over data from residue
17	Sniffing	Using devices or programs to monitor data travelling over the network
18	Social_Engineering	Using social skills to manipulate people into revealing vulnerable information
19	Technological_Obsolescence	Use of antiquated & outdated technologies
20	VM_Threats	Related to virtualization and hypervisor vulnerabilities in cloud

These identified threats are going to act as our repository from which threats are retrieved and mapped to actors based on their functionalities and other parameters as mentioned in actor profile created during step 2. Hence the threats related to each actor after the associations are listed in tabular form as shown in Table 7.

Table 7: Threats association with functionalities

Actors	Functionality	Threats
Cloud Customer	1. Registration & Login	Password_Cracking; Impersonate; Sniffing
	2. Update Login details	Change_Data; Social_Engg.
	3. Store data into Cloud	Change_Data; Disclose_Data; Malicious_Code; Sniffing; Repudiate_Send
	4. Manage automatic backup	Data_Theft
	5. Manage sharing with Cloud users	Disclose_Data; Human_Error
	6. Download data stored in Cloud	Repudiat_Receive; Data_Theft; Impersonate
	7. Select storage location	Multilocation_Dataplacement
	8. Make payment for services used	Sniffing
Cloud Users	1. Registration & Login	Password_Cracking; Impersonate; Sniffing
	2. View shared data based on permission	Disclose_Data
	3. Submit request to join group	Impersonate; DoS_DDoS
	4. Unjoin a group	Impersonate; DoS_DDoS
Cloud Service Provider	1. Manage Cloud customers account	Data_Theft; Insider
	2. Manage Customer data	Insider; Data_Theft; Multilocation_DataPlacement; Change_Data
	3. Manage Cloud hardware's & softwares	Natural_Disaster; VM_Threats; Outsider; Technological_Obsolescence; Scavenging
	4. Receive Cloud usage payment	Repudiate_Receive
	5. Maintain SLA	Human_Error
Cloud Service Integrator	1. Registration & Login	Impersonate; Password_Cracking; Sniffing
	2. Combines Cloud based & in-house services	VM_Threats; Sabotage; Malicious_Code
Security Administrator	1. Take action on security breaches	Data_Theft; Human_Error; Outsider
	2. Apply security patch to software	Malicious_Code; Data_Theft
	3. Update system with antiviruses and firewalls	Technological_Obsolescence
Auditor	1. Maintain the audit and log details	Data_Theft; Change_Data
	2. Release audit reports	Disclose_Data

4.3.4 Determination of Security Requirements *(Step 4)*

After the identification of threats in previous step, we have to determine the Security Requirements, which are basically high level natural language solution to mitigate threats. Security Requirements are goals and constraints which affect the integrity, confidentiality and availability of applications and data [7].

We have already defined and discussed the various Security Requirements related to Cloud system earlier in this chapter, which are also listed below:

- Identification & Authentication
- Authorization
- Immunity
- Integrity
- Intrusion Detection
- Non Repudiation
- Privacy
- Security Auditing
- Survivability
- Recoverability
- Physical Access Protection
- System Maintenance

Now these Security Requirements are retrieved from the repository and mapped to mitigate the previously identified threats. Hence the results after the mapping with threats are shown in Table 8.

Table 8: Threats association with Security Requirements

Sl. #	Threats	Security Requirements
1	Data_Theft	Identification & Authentication
		Authorization
		Intrusion Detection
		Privacy
		Physical Access Protection
2	Disclose_Data	Authorization
		Privacy
3	DoS_DDoS	Identification & Authentication
		Intrusion Detection
		Survivability
		Recoverability
4	Human_Error	Security Auditing
		Recoverability
5	Impersonate	Identification & Authentication
		Intrusion Detection
6	Insider	Authorization
		Privacy
		Physical Access Control
7	Malicious_Code	Immunity
		Survivability
8	Multilocation_Dataplacement	Privacy
9	Natural_Disaster	Survivability
		Recoverability
10	Outsider	Identification & Authentication
		Privacy
		Physical Access Control
11	Password_Cracking	Identification & Authentication
12	Change_Data	Authorization
		Integrity
		Recoverability
13	Repudiate_Receive	Non Repudiation
		Security Auditing
14	Repudiate_Send	Non Repudiation
		Security Auditing
15	Sabotage	Physical Access Control
		Survivability
		Recoverability
16	Scavenging	Privacy
17	Sniffing	Intrusion Detection
		Privacy
18	Social_Engineering	Privacy
19	Technological_Obsolescence	Immunity
20	VM_Threats	Immunity
		Survivability
		Recoverability

4.3.5 Association of Security Requirements with Functionalities *(Step 5)*

In this last step of our framework we have associated the identified Security Requirements from previous step to the set of functionalities borrowed from Common Criteria [29] approach as given in Table 9. This association with functionalities will help the Security design engineers to gain better understanding of Security Requirements and aid them in selecting suitable security design mechanism.

Table 9: Security Requirements and Security Functionalities association

Sl. #	Security Requirements	Security Functionalities
1	Identification & Authentication	User Identification (UID)
		User Authentication (UAU)
		Authentication Failures (AFL)
		Limitation on scope of selectable attributes (LSA)
		Limitation on multiple concurrent sessions (MCS)
		System access banners (TAB)
		System access history (TAH)
		System session establishment (TSE)
2	Authorization	Security management roles (SMR)
		Security Attribute Expiration (SAE)
		User-Subject Binding (USB)
		Session locking and termination (SSL)
3	Immunity	Testing of external entities (TEE)
		SSF self test (TST)
		Revocation (REV)
		Import from Outside (ITC)
4	Integrity	Data Authentication (DAU)
		Internal System Transfer (ITT)
		Stored Data Integrity (SDI)
		Rollback (ROL)
5	Intrusion Detection	Replay detection (RPL)
		Testing of external entities (TEE)
		Revocation (REV)
		Information flow control policy (IFC)
		Information flow control functions (IFF)
6	Non-Repudiation	Non-repudiation of Origin (NRO)
		Non-repudiation of Recipient (NRR)
7	Privacy	Cryptographic Key Management (CKM)
		Cryptographic Operations (COP)
		User data Confidentiality Transfer Protection (UCT)

		Import from Outside (ITC)
		Internal System Transfer (ITT)
		Anonymity (ANO)
		Pseudonymity (PSE)
		Unlinkability (UNL)
		Unobservability (UNO)
		Trusted Path (TRP)
8	Security Auditing	Security Audit Automatic Response (ARP)
		Security Audit Data Generation (GEN)
		Security Audit Analysis (SAA)
		Security Audit Review (SAR)
		Security Audit Event Storage (STG)
9	Survivability	Fault Tolerance (FLT)
		Priority of service (PRS)
		Resource allocation (RSA)
		Fail Secure (FLS)
10	Recoverability	Trusted recovery (RCV)
		State synchrony protocol (SSP)
		Rollback (ROL)
11	Physical Access Protection	Access Control Policy (ACC)
		Access Control Function (ACF)
		SSF physical protection (PHP)
12	System Maintenance	Management of Security Attributes (MSA)
		Management of Security Functionality Data (MTD)
		Internal System Transfer (ITT)

Hence we have finally elicited the true Security Requirements based on our proposed framework for a Cloud system as shown in Table 9. Now these Security Requirements can be prioritized on the basis of risk measures [8] and further managed to handle the changes in requirements [7].

And finally these Security Requirements are passed over to the next phase of Security engineering i.e. Security Design Engineering, so that appropriate design decision can be taken by Security Design Engineers to create a secure and reliable Cloud Storage-as-a-Service model.

Chapter 5

CASE STUDY ON CLOUD (STORAGE-AS-A-SERVICE)

Usage of Cloud storage services basically means uploading critical data on third party storage servers where no prior relationship has been established based on trust. Cloud customers who upload their personal data on cloud want to be sure that only authorized and limited people are able to access it which also excludes the provider, because the disclosure of customer critical data or business secrets poses a severe threat to individual's identity or company's business.

Hence in this chapter we are going to discuss our analysis done on some popular Storage-as-a-service available in the market for general users. Our analysis is based on the study from online computer magazines and dedicated websites, which first gives a basic overview about their features and costs and then it focuses some light on their available features from security point of view which uncovers the vulnerabilities. And finally we shows that how our framework can mitigate those threats early in the development process.

We have done our analysis on two popular Cloud storage services which are:

- ***Cloud Me***
- ***Dropbox***

5.1 CLOUDME

- ***Overview***

CloudMe [30] is operated by Sweden based Xcerion. CloudMe can be accessed by their Web Desktop supported in IE7 or they provide different tools like Easy Upload, Web Desktop and CloudMe Lite. These tools are used for managing various functionalities like storing data, downloading data and recovery etc. Creating backup functionality is not available in CloudMe.

CloudMe provides various ways to share data like:

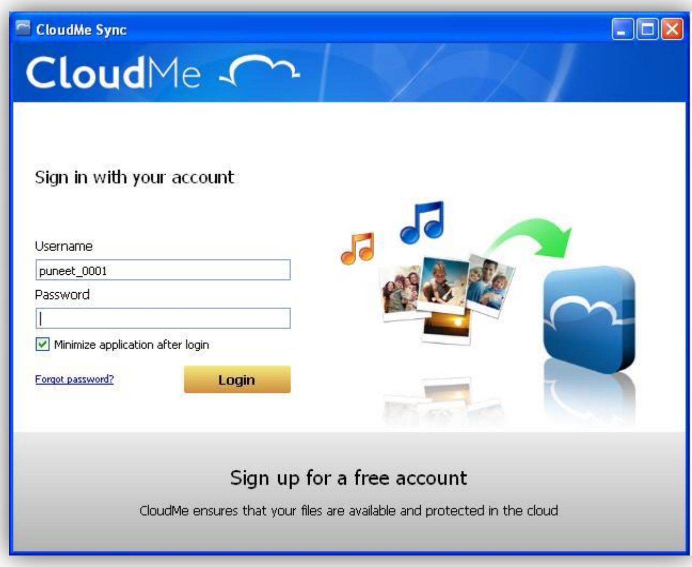
- *Sharing data with other subscribers*: Customer can choose various access rights (read, write, delete etc.) for others when sharing data.
- *Sharing with everyone*: There is a folder 'Public' which automatically publishes the data shared by the customer at URL link http://my.cloudme.com/UserID/webshare.

- ***Subscription Cost***

Subscription	Storage Space	Price
Free	3 GB	Null
Paid_1	25 GB	$ 49.99 per year
Paid_2	100 GB	$ 99.99 per year

- ***Registration and Login***

For registration on CloudMe, users have to provide desired username, password, first name, last name, country and email. The password should be of at least six characters, if someone enters short password then registration cannot be completed with CloudMe. It does not validate the users e-mail account used in registration, inviting an 'impersonate' threat. It also displays the availability or non-availability of User ID during registration to users, which enables information gathering and due its weak password strength 'password cracking' threat is possible on CloudMe.



Sign in with your account

Username: puneet_0001

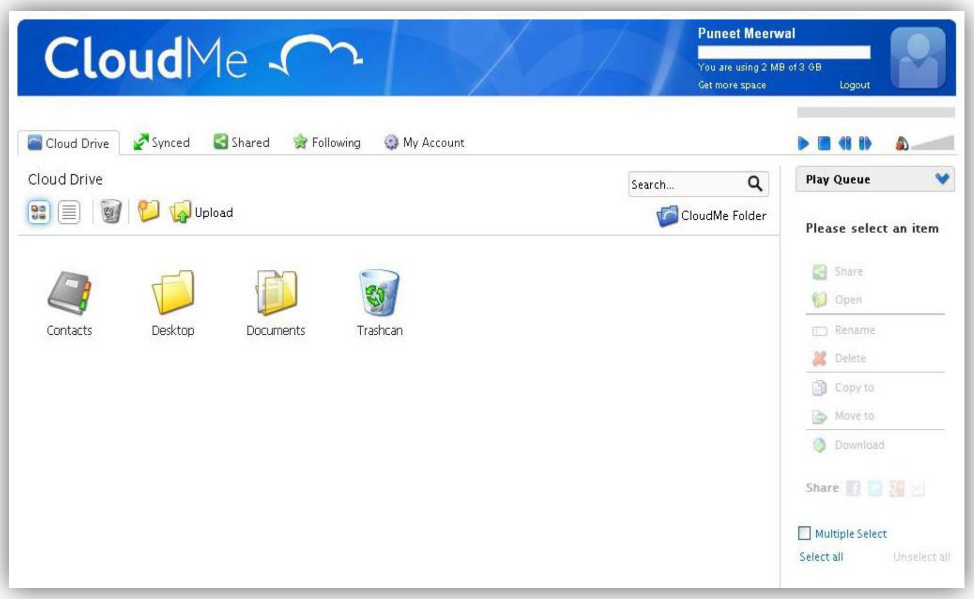
Password:

☑ Minimize application after login

Sign up for a free account

CloudMe ensures that your files are available and protected in the cloud

Figure 20: Cloud Me client application interface



Puneet Meerwal

You are using 2 MB of 3 GB

Cloud Drive

Please select an item

Figure 21: Cloud Me simple web based interface

- ***Communication Security***

CloudMe does not provide encryption of customer critical data when transmitted between client and server, allowing ‘sniffing’ of data possible.

- ***Data Encryption***

CloudMe also does not provide encryption of customer data stored on its servers, which make various threats possible on it like change data and disclose data.

- ***Data Placement***

All the data centers owned by CloudMe are located in Sweden and its does not offer customer to choose server location.

5.2 DROPBOX

- ***Overview***

Dropbox [31] is operated by US based Dropbox Inc. which uses Amazon Web Services (AWS) for storage. It allows access from web interface for account management and data access, as well as provides client application for Windows, Linux and MacOS.

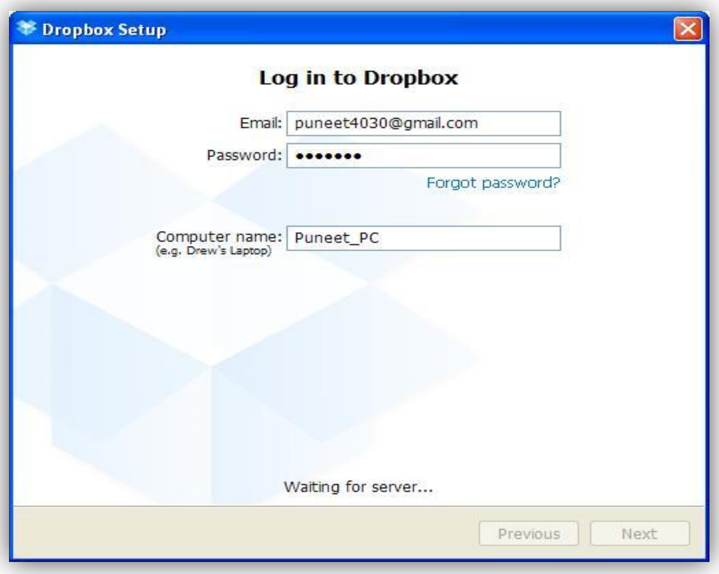
During client application installation, it creates Dropbox folder such that all files and data placed in this folders are automatically uploaded on its servers. Additionally it also provide upload & download feature through the web interface. Similar to CloudMe, creating backup functionality is also not available in Dropbox.

Dropbox allows sharing of data in many ways like:

- *Sharing data with other subscribers*: It allows sharing by inviting other subscribers by entering their Username or email address.
- *Sharing data with everyone*: It allows sharing with everyone by copying data to specific 'Public_Share' folder which is then mapped to a URL like http://dl.dropbox.com/UserID/xyz, where xyz is a unique number.

➢ ***Subscription Cost***

Subscription	Storage Space	Price
Free	2 GB	Null
Paid_1	50 GB	$ 9.99 per month
Paid_2	100 GB	$ 19.99 per month



Email: puneet4030@gmail.com

Password: •••••••

Computer name: Puneet_PC
(e.g. Drew's Laptop)

Waiting for server...

Figure 22: Dropbox client application interface

➢ ***Registration and Login***

Dropbox provide TLS secure communication channel for both registration and login process. Customers are allowed to enter first and last name, email address and desired password during the registration process. Unlike CloudMe, email address is used to login into Dropbox and password length should be of six characters. It shows the already registered e-mail address error warning to users during registration process which in addition to weak password strength makes 'password cracking' threat easy. It also does not send any activation emails after the registration to customers, resulting in 'impersonate' threat possible on it.

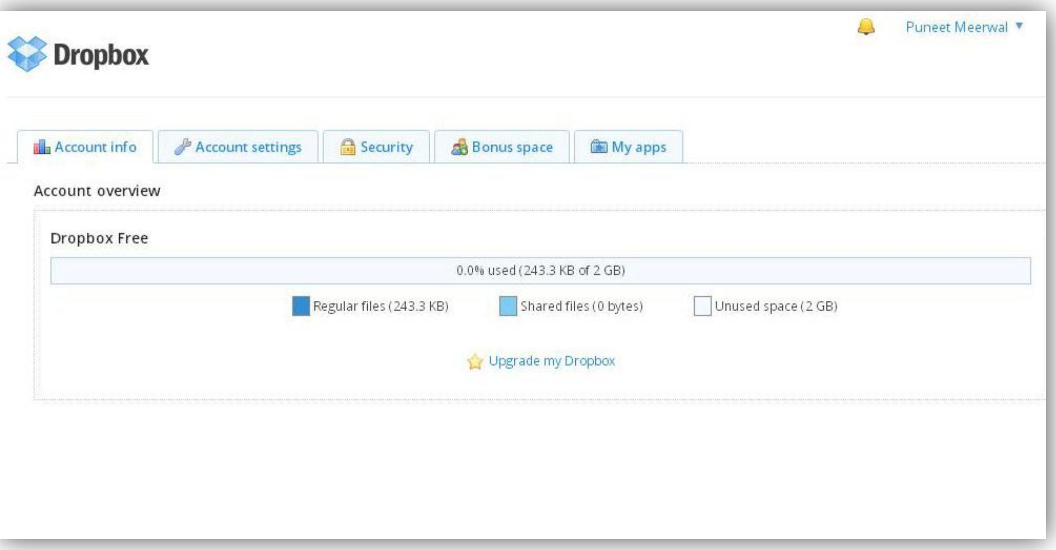


Figure 23: Dropbox web based interface

➢ *Communication Security*

Dropbox uses TLS and HTTPS to encrypt the data communication between the client and server.

➢ *Data Encryption*

It uses AES-256 encryption algorithm for the encryption of customer data stored on its servers, but only at server side using its own encryption key of which the client is unaware. Hence, a 'change data' threat applies on Dropbox.

- ***Data Placement***

It does not offer data placement choice to customers, as it uses Amazon S3 servers located in United States.

Hence based on our analysis, there are some vulnerabilities presents in above mentioned popular Cloud storage services which makes a compromise with the security of client's critical data either transmitted or stored on their servers. Table 10 summarizes the possible threats discussed above in our analysis.

Table 10: Threats possible on popular Cloud storage services

	Cloud Me	***Dropbox***
Registration & Login	Impersonate Password Cracking	Impersonate Password Cracking
Communication Security	Sniffing	-
Data Encryption	Change Data Disclose Data	Change Data Disclose Data
Data Placement	-	-

Hence we can say that all these analyzed threats are the results of improper elicitation and handling of Security Requirements during the development process of Cloud storage services. But these threats are not possible if some structured or proper mechanism for Security Requirements elicitation was used early during the development process like ours as discussed in chapter 4.

Hence to prove the feasibility of our proposed approach to develop secure Cloud system, we apply our framework on two functionalities i.e. 'Registration & Login' and 'Store data on Cloud'. Finally we compare our obtained results with the analysis done.

5.3 APPROACH APPLIED ON CLOUDME/DROPBOX

We are now going to use our proposed framework in step by step manner for eliciting true Security Requirements for Cloud Storage service similar to CloudMe and Dropbox.

5.3.1 Functionality 1 - "Registration & Login"

Step 1) Identification of Actors

Three main direct actors we consider for CloudMe/Dropbox services are '*CloudMe/Dropbox Customer*' who stores its personal data on Cloud, '*CloudMe/Dropbox Service Provider*' who provide and manage the Cloud services and '*Cloud Users*' with whom the customer share its data.

- *CloudMe/Dropbox Customer*
- *CloudMe/Dropbox Service Provider*
- *Cloud User*

Step 2) Identification of functionalities

As their functionalities are numerous, but here for analysis purpose we are interested and consider only the '*Registration & Login*' functionality of CloudMe/Dropbox Customer.

Step 3) Identification of 'Threats' associated with actor functionalities

For identification of threats we use our previously defined threat repository from Chapter 4. Hence based on functionality '*Registration & Login*' the threats possible are 'Password_Cracking', 'Impersonate' and 'Sniffing'.

Step 4) Determination of Security Requirements

Security Requirements are also retrieved from the repository as defined in chapter 4. Hence based on our approach, the Security Requirements used for mitigating identified threats are shown in Table 11.

Step 5) Association of Security Requirements with Functionalities

The final output obtained after associating the Security Requirements to set of Security Functionalities is also shown in Table 11.

Table 11: Security Requirements for 'Registration & Login'

Actor	Functionality	Threats	Security Requirements	Security Functionalities
CloudMe/ Dropbox Customer	Registration & Login	Impersonate	Identification & Authentication	UID, UAU, AFL, LSA, MCS, TAB, TAH, TSE
			Intrusion Detection	RPL, TEE, REV, IFC, IFF
		Password_Cracking	Identification & Authentication	UID, UAU, AFL, LSA, MCS, TAB, TAH, TSE
		Sniffing	Intrusion Detection	RPL, TEE, REV, IFC, IFF
			Privacy	CKM, COP, UCT, ITC, ITT, ANO, PSE, UNO, UNL, TRP

5.3.2 Functionality 2 - "Store data into Cloud"

Step 1) Identification of Actors

As we have already identified that the three main actors are: '*CloudMe/Dropbox Customer*', '*CloudMe/Dropbox Service Provider*' and '*CloudMe/Dropbox User*'.

Step 2) Identification of functionalities

We only consider the 'Store data into Cloud' functionality of CloudMe/Dropbox Customer here for analysis purpose.

Step 3) Identification of 'Threats' associated with actor functionalities

Based on our previously defined repository, the threats possible on functionality '*Store data into CloudMe/Dropbox*' are 'Change_Data', 'Disclose_Data', 'Malicious_Code', 'Sniffing' and 'Repudiate_Send'.

Step 4) Determination of Security Requirements

Based on our approach, the Security Requirements elicited to mitigate identified threats are shown in Table 12.

Step 5) Association of Security Requirements with Functionalities

The final output obtained after the association of Security Requirements with Security Functionalities is also shown in Table 12.

Table 12: Security Requirements for 'Store data into Cloud'

Actor	Functionality	Threats	Security Requirements	Security Functionalities
CloudMe/ Dropbox Customer	Store Data into CloudMe/ Dropbox	Change_Data	Authorization	SMR, SAE, USB, SSL
			Integrity	DAU, ITT, SDI, ROL
			Recoverability	RCV, SSP, ROL
		Disclose_Data	Authorization	SMR, SAE, USB, SSL
			Privacy	CKM, COP, UCT, ITC, ITT, ANO, PSE, UNO, UNL, TRP
		Malicious_Code	Immunity	TEE, TST, REV, ITC
			Survivability	FLT, PRS, RSA, FLS
		Sniffing	Intrusion Detection	RPL, TEE, REV, IFC, IFF
			Privacy	CKM, COP, UCT, ITC, ITT, ANO, PSE, UNO, UNL, TRP
		Repudiate_Send	Security Auditing	ARP, GEN, SAA, SAR, STG
			Non Repudiation	NRO, NRR

5.4 COMPARISON

Through our analysis on real world Cloud storage services, we identify that some threats are still possible on the popular Storage services which most of the people uses around the globe to store their personal and critical data. These threats are possible only because of improper elicitation or handling of Security Requirements during their development process.

However if we apply our proposed structured framework for developing similar Cloud Storage service, we uncover these possible threats early in the development process so that proper Security Design decision can be taken early to mitigate those threats as per the identified true Security Requirements.

.

Chapter 6

IMPLEMENTATION

6.1 Tools Used

We have used the following tools during the development of our project.

- ***Java Platform Standard Edition 7 Development Kit (JDK 7)***:- JDK 7 provides tools and other utilities that help to develop, execute, debug, and document programs written in the Java programming language. It can be downloaded from Sun Microsystems website on free license basis.
- ***Eclipse IDE Juno Service Release 2***:- The standard distribution of Eclipse IDE (Integrated Development Environment) gives everything to develop Java SE applications, web applications, and Java EE enterprise applications, which makes programming enterprise applications and web services much simpler. It’s open source software which can be downloaded from Eclipse website (www.eclipse.org).
- ***Microsoft Access***: -Microsoft access has been used to make relations for our development tool. The main theme to use access is it is a very light weight database and provides all the basic database utilities that we need in our project. We do not want any security feature to the database hence we have used this database.

6.2 Running the Code

First you need to create a DSN connection so that database can be accessed through JDBC (Java Database Connectivity). To create DSN connections follow the steps:

- Go to Control Panel.
- Open Administrative Tools.
- Select and open ODBC.
- Click DSN Tab.
- Click on ADD button and select Microsoft Access Driver (*.mdb) from available list of drivers.
- Give suitable name to DSN and select the location where relations are stored.
- Press the FINISH button.

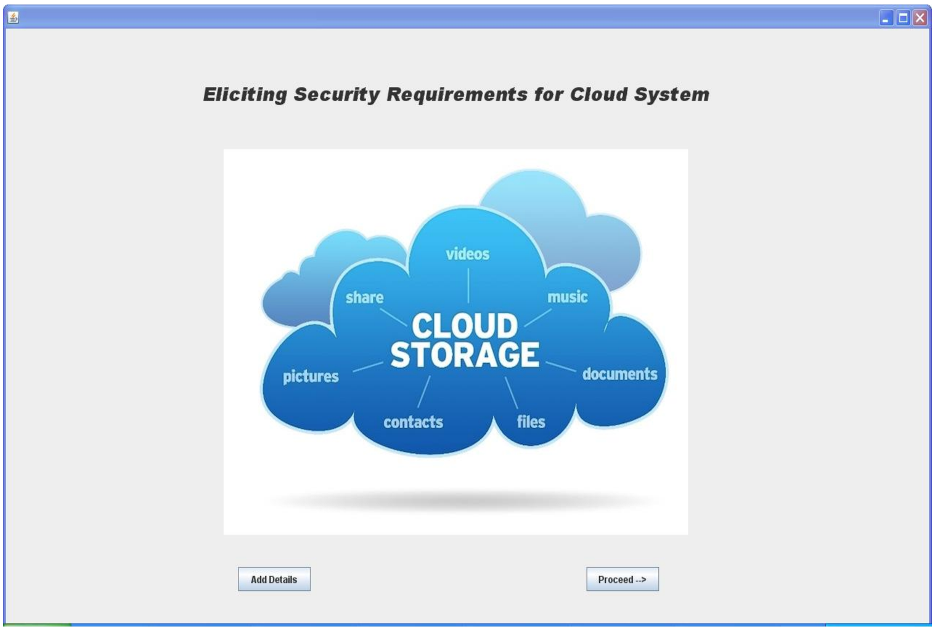
In order to run the code you need to have JDK 7 installed on the system on which you want to execute the code. To execute the code under windows environment, open the command prompt and do the following steps:

- Select directory location where all files are present.
- Then to compile the file type "**javac *filename*.java**"
- After compiling, to run the code type "**java *filename*"**

OR

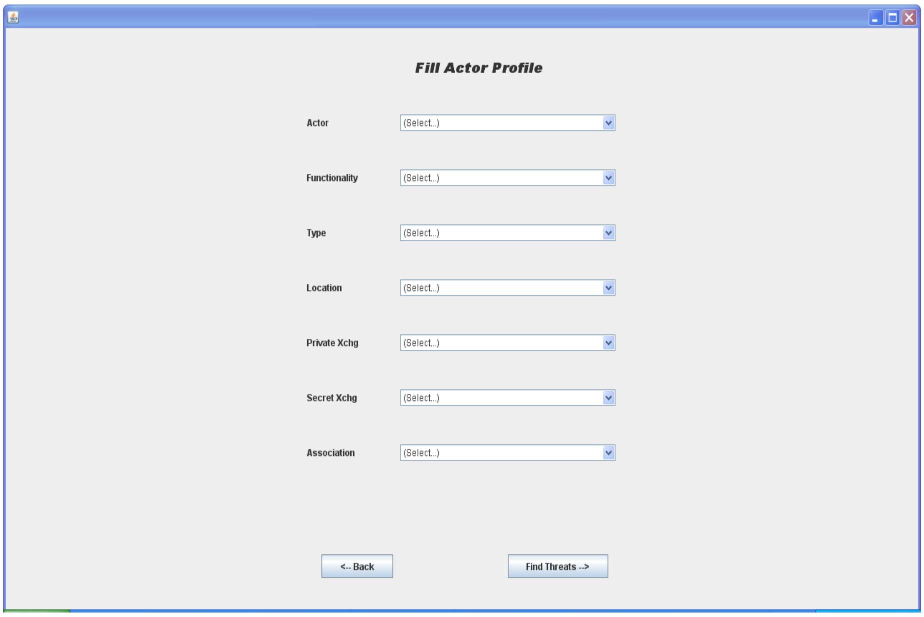
We may run the code using Eclipse IDE. In both the cases i.e. Eclipse IDE or Windows Environment, the same window will open containing a standard menu for creating Actor profile.

6.3 Snapshots



Eliciting Security Requirements for Cloud System

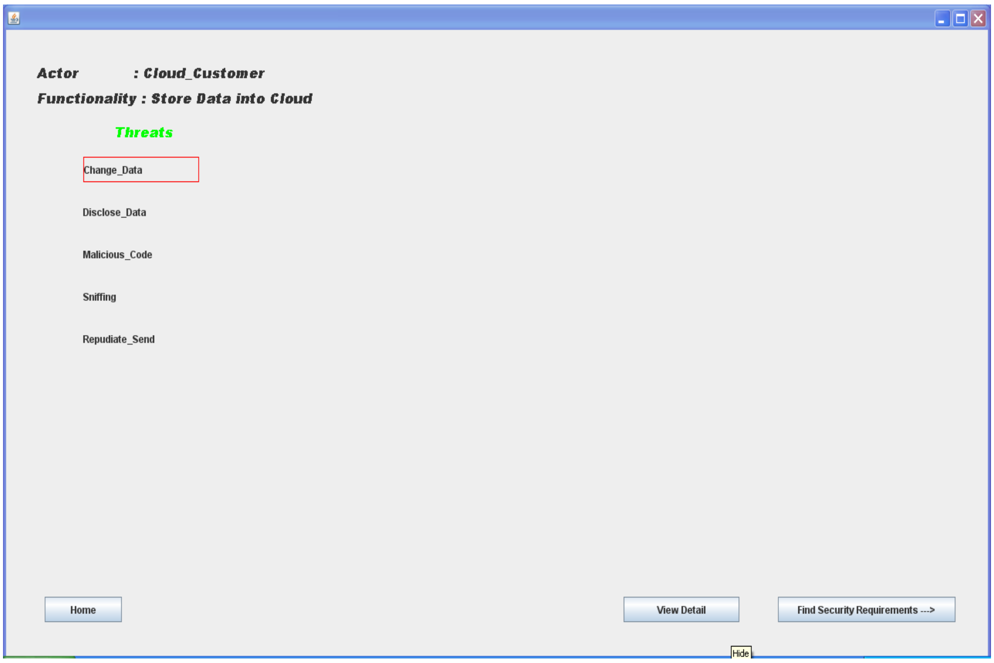
Figure 24: First main window of our Tool



Fill Actor Profile

Actor	(Select..)
Functionality	(Select..)
Type	(Select..)
Location	(Select..)
Private Xchg	(Select..)
Secret Xchg	(Select..)
Association	(Select..)

Figure 25: Actor profile window



Actor : Cloud_Customer

Functionality : Store Data into Cloud

Threats

Change_Data

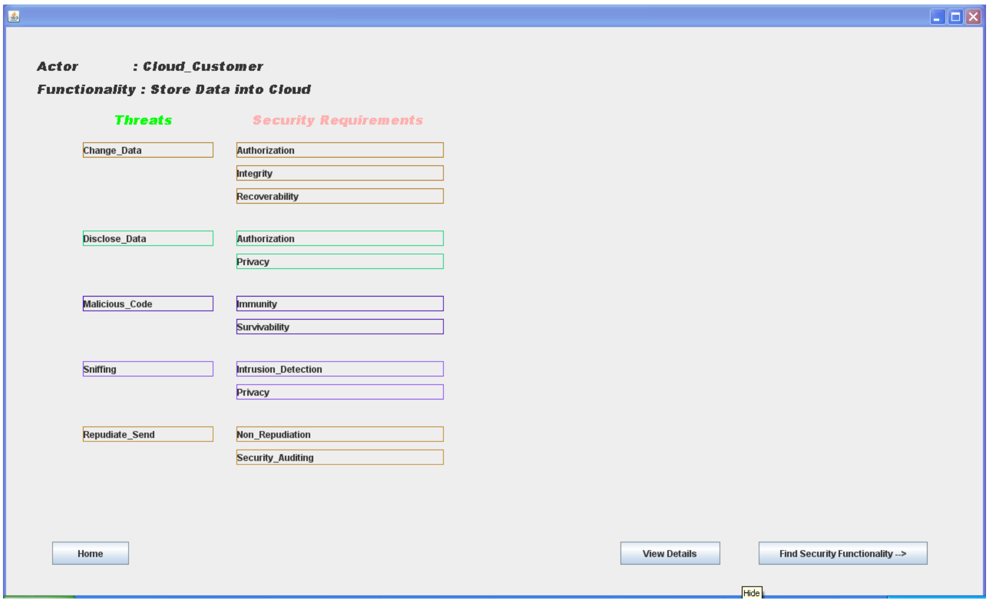
Disclose_Data

Malicious_Code

Sniffing

Repudiate_Send

Figure 26: Threats identification window

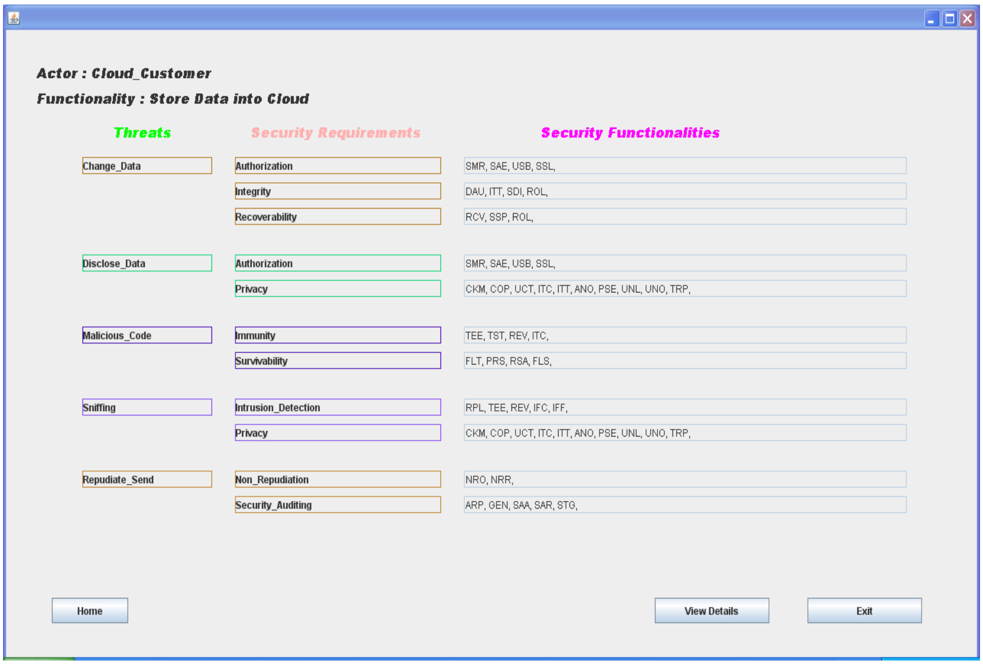


Actor : Cloud_Customer

Functionality : Store Data into Cloud

Threats	Security Requirements
Change_Data	Authorization Integrity Recoverability
Disclose_Data	Authorization Privacy
Malicious_Code	Immunity Survivability
Sniffing	Intrusion_Detection Privacy
Repudiate_Send	Non_Repudiation Security_Auditing

Figure 27: Security Requirements identification window

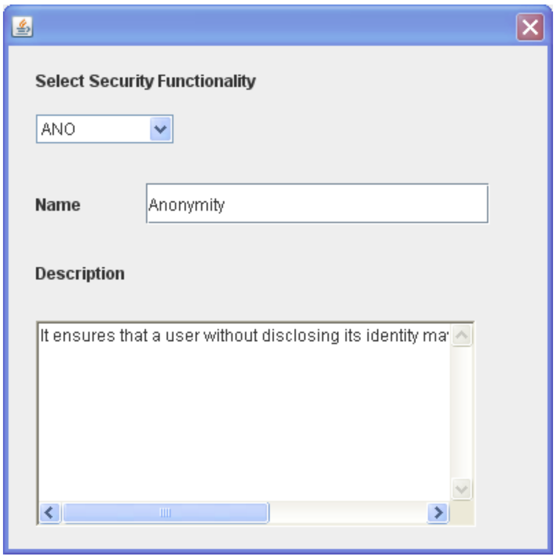


Actor : Cloud_Customer

Functionality : Store Data into Cloud

Threats	Security Requirements	Security Functionalities
Change_Data	Authorization	SMR, SAE, USB, SSL,
	Integrity	DAU, ITT, SDI, ROL,
	Recoverability	RCV, SSP, ROL,
Disclose_Data	Authorization	SMR, SAE, USB, SSL,
	Privacy	CKM, COP, UCT, ITC, ITT, ANO, PSE, UNL, UNO, TRP,
Malicious_Code	Immunity	TEE, TST, REV, ITC,
	Survivability	FLT, PRS, RSA, FLS,
Sniffing	Intrusion_Detection	RPL, TEE, REV, IFC, IFF,
	Privacy	CKM, COP, UCT, ITC, ITT, ANO, PSE, UNL, UNO, TRP,
Repudiate_Send	Non_Repudiation	NRO, NRR,
	Security_Auditing	ARP, GEN, SAA, SAR, STG,

Figure 28: Identifying Security Functionalities window



Select Security Functionality

ANO

Name Anonymity

Description

It ensures that a user without disclosing its identity ma

Figure 29: Functionality detail popup window

Chapter 7

CONCLUSION

Usage of Cloud storage services basically means uploading critical data on third party storage servers where no prior relationship has been established based on trust. Cloud customers who upload their personal data on Cloud want to be sure that only authorized persons can access their data, which may exclude CSP also. Nowadays large number of users and businesses are adopting Cloud computing due to its enormous benefit, but this adoption also brings many security concerns as discussed earlier in Cloud system.

Software engineering community says that to implement Security, first Security Requirements should be gathered, analyzed and then implemented.

Hence, in our thesis we have discussed that how Security Requirements can be elicited in proper and structured way for Storage-as-a-service model in Cloud computing system. In our proposed framework for Security Requirements elicitation we have extended view point oriented Security Requirement elicitation approach [7] to Cloud System as it consider both functional and non functional requirements to trace out true Security Requirements early in the software development process.

Then we have done an analysis on some popular Cloud storage service, which uncovers the vulnerabilities resulting in threats due to improper handling of Security Requirements during their development process. And compare it with our proposed approach to prove the feasibility.

APPENDIX

ABBREVIATIONS USED

ACC	Access Control Policy
ACF	Access Control Function
AFL	Authentication Failures
ANO	Anonymity
ARP	Security Audit Automatic Response
CKM	Cryptographic Key Management
COP	Cryptographic Operations
CSI	Cloud Service Integrator
CSP	Cloud Service Provider
CSU	Cloud Service Users
DAU	Data Authentication
FLS	Fail Secure
FLT	Fault Tolerance
GEN	Security Audit Data Generation
HTTPS	Hypertext Transfer Protocol Secure
IDC	International Data Corporation
IFC	Information flow control policy
IFF	Information flow control functions
ITC	Import from Outside
ITT	Internal System Transfer
LSA	Limitation on scope of selectable attributes
MCS	Limitation on multiple concurrent sessions
MOF	Management of function in SSF
MSA	Management of Security Attributes
MTD	Management of Security Functionality Data
NIST	National Institute of Standards and Technology
NRO	Non-repudiation of Origin
NRR	Non-repudiation of Recipient
PHP	SSF physical protection
PRS	Priority of service
PSE	Pseudonymity

RCV	Trusted recovery
REV	Revocation
ROL	Rollback
RPL	Replay detection
RSA	Resource allocation
SAA	Security Audit Analysis
SAE	Security Attribute Expiration
SAR	Security Audit Review
SDI	Stored Data Integrity
SDLC	Software Development Life Cycle
SDT	Security Design Template
SEP	Security Engineering Process
SMR	Security management roles
SOS	Specification of Secrets
SSL	Session locking and termination
SSP	State synchrony protocol
STG	Security Audit Event Storage
STM	Time stamps
TAB	System access banners
TAH	System access history
TEE	Testing of external entities
TLS	Transport Layer Security
TRC	Internal System SSF data replication consistency
TRP	Trusted Path
TSE	System session establishment
TST	SSF self test
UAU	User Authentication
UCT	User data Confidentiality Transfer Protection
UID	User Identification
UIT	User Data Integrity Transfer Protection
UNL	Unlinkability
UNO	Unobservability
USB	User-Subject Binding
VOSREP	Viewpoint Oriented Security Requirements Elicitation process

REFERENCES

[1] Data Breaches, http://www.databreaches.net/, May 2013.

[2] International Data Center, http://www.idc.com.

[3] Firesmith D. G., "Engineering Security Requirements" Journal of Object Technology, pp. 53-68, 2003.

[4] Johnson J., "Chaos: The Dollar Drain of IT Project Failures" Application Development Trends, pp. 41-47, 1995.

[5] Cloud computing Wikipedia,https://en.wikipedia.org/wiki/Cloud computing, 2013

[6] Firesmith D. G., "Security Use Cases" Journal of Object Technology, pp. 53-64, 2003.

[7] Agarwal A., Gupta D., “Security Requirement Elicitation Using View Points for online System”, IEEE ICETET '08, pp. 1238-1243, 2008.

[8] Jaiswal S., Gupta D., “Security Requirement Prioritization”, in the proceeding of SERP’09, pp. 673- 679, 2009.

[9] Chatterjee K., Gupta D., De A., “A Framework for Security Design Engineering Process”, in ICIP, CCIS 157, pp. 287- 293, 2011.

[10] Gupta D., Chatterjee K., De A., “A Framework for Development of Secure Software”, CSI Transaction on ICT, 2013

[11] Gupta D., Chatterjee K., Jaiswal S., "A Framework for Security Testing", in ICCSA 2013, Part III, LNCS 7973, pp. 187-198, Springer- Verlag Berlin Heidelberg, 2013.

[12] Hanna S., http://www.ists.dartmouth.edu/docs/HannaCloudComputingv2.pdf, May 2013.

[13] Iankoulova I., Daneva M., "Cloud Computing Security Requirements: A Systematic Review." IEEE RCIS '12, pp 1-7, 2012.

[14] Mell P., Grance T., "The NIST Definition of Cloud Computing", NIST. http://www.nist.gov/itl/cloud/upload/cloud-def-v15.pdf, May 2013.

[15] Jansen W. A., “Cloud Hooks: Security and Privacy issues in cloud computing”, NIST 44th Hawaii International conference on System Sciences, pp 1-10, 2011.

[16] Cloud Security Alliance, “Top threats to cloud computing, version1.0”, Tech. Rep., March 2010.

[17] Ren K., Wang C., Wang Q., “Security challenges for the public cloud”, Internet Computing, IEEE, pp 69-73, 2012.

[18] McKendrick J., "Cloud Computing's Vendor Lock-In Problem: Why the Industry is Taking a Step Backward", Forbes, 2011.

[19] Sommerville I., “Software Engineering”, Pearson Education, 9th Edition, 2003.

[20] McDermott J., Fox C., "Using abuse case models for security requirements analysis" 15th Annual Computer security applications conference. IEEE, 1999.

[21] Ware M S., Bowles J.B., Eastman C.M., "Using the Common Criteria to Elicit Security Requirements with Use Cases." SoutheastCon. IEEE, pp 273-278, 2006.

[22] "Common criteria for information technology security evaluation", Technical report CCIMB 99–031, Common Criteria Implementation Board, 1999.

[23] Alexander I.F., “Modelling the interplay of conflicting goals with use and misuse cases”, 8th international workshop on requirements engineering: foundation for software quality (REFSQ’02), 2002.

[24] Sindre G., Opdahl A.L., “Eliciting security requirements with misuse cases”, Requirements Engineering Journal, Springer-Verlag, pp. 34-44, 2005.

[25] Alexander I.F., “Misuse cases, use cases with hostile intent”, IEEE Software, 2003.

[26] Robert J., Ellison, "Attack Trees” Software Engineering Institute, Carnegie Mellon University, 2005.

[27] Mouratidis H., Giorgini P., Manson G., Philp I., "A Natural Extension of Tropos Methodology for Modelling Security", Agent Oriented Methodologies Workshop, 2002.

[28] Lamsweerde A., "Elaborating security requirements by construction of intentional anti-models", 26th International Conference on software engineering IEEE, pp 148-157, 2004.

[29] CCRA Common Criteria Portal, “Security Functional Components”, http://www.commoncriteriaportal.org/files/ccfiles/CCPART2V3.1R4.pdf, 2013.

[30] Cloud Me storage service, www.cloudme.com/.

[31] DropBox storage service, https://www.dropbox.com/.