Design Methodology for Secure Cloud Systems

A dissertation submitted in the partial fulfillment for the award of Degree of

Master of Technology

In

Software Technology

By

Jyotsna Sharma (Roll no. 2K12/SWT/10)

Under the guidance of

Prof (Dr) Daya Gupta

DEPARTMENT OF SOFTWARE ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

BAWANA ROAD, DELHI

2015

DECLARATION

I hereby want to declare that the thesis entitled "**Design Methodology for Secure Cloud Systems"** which is being submitted to the **Delhi Technological University**, in partial fulfillment of the requirements for the award of degree in **Master of Technology in Software Technology** is an authentic work carried out by me. The material contained in this thesis has not been submitted to any institution or university for the award of any degree.

_____

Jyotsna Sharma

Department of Software Engineering

Delhi Technological University,

Delhi.

CERTIFICATE

DELHI TECHNOLOGICAL UNIVERSITY

BAWANA ROAD, DELHI-110042

Date: _____

This is to certify that the thesis entitled "**Design Methodology for Secure Cloud Systems"** submitted by **Jyotsna Sharma (Roll Number: 2K12/SWT/10),** in partial fulfillment of the requirements for the award of degree of Master of Technology in Software Technology, is an authentic work carried out by him under my guidance. The content embodied in this thesis has not been submitted by him earlier to any institution or organization for any degree or diploma to the best of my knowledge and belief.

Prof (Dr) Daya Gupta,

Department of Software Engineering,

Delhi Technological University, Delhi-110042

ACKNOWLEDGEMENT

I would like to take this opportunity to express my appreciation and gratitude to all those who have helped me directly or indirectly towards the successful completion of this work.

Firstly, I would like to express my sincere gratitude to my guide **Prof (Dr) Daya Gupta**, **Department of Software Engineering**, **Delhi Technological University, Delhi** whose benevolent guidance, encouragement, constant support and valuable inputs were always there for me throughout the course of my work. Without her continuous support and interest, this thesis would not have been the same as presented here.

Also I would like to extend my thanks Mrs Shruti Jaiswal (Research Scholar, Delhi Technological University) for her critical review and support, and the entire staff in the Department of Software Engineering, DTU for their help during my course of work.

JYOTSNA SHARMA

2K12/SWT/10

Table of Contents

ABSTRACT 6

1 Introduction 8

1.1 Introduction 8

1.2 **Motivation** 9

1.3 Related Work 10

1.4 Problem Statement 11

1.5 Scope Of The Work 12

1.6 Approach 12

1.7 Organization Of Thesis 14

2 Background Study 15

2.1. Cloud Computing 15

2.2. Storage as a Service (SaaS) 19

3 Security Engineering 22

3.1 Proposed framework for security engineering 23

3.2 Security Requirement Engineering 24

3.3 Security Design Engineering 25

3.4 Security Implementation and Testing 26

4 Clouds System Security Requirements 27

5 Design Methodology for Secure Cloud Systems 45

6 Case Study 61

CONCLUSIONS 66

REFERENCES 67

TABLE OF FIGURES

FIGURE 2-1 FEATURES OF PUBLIC, PARTNER AND PRIVATE CLOUD 16
FIGURE 2-2 AREAS OF SECURITY CONCERNS IN A CLOUD SYSTEM 18
FIGURE 2-3 EVOLUTION OF CLOUD STORAGE....20
FIGURE 2-4 EVOLUTION OF CLOUD STORAGE....21
FIGURE 3-1 FRAMEWORK FOR SECURITY ENGINEERING PROCESS....23
FIGURE 6-1 AMAZON WEB SERVICES CLOUD PLATFORM 62

TABLE OF TABLES

TABLE 2.1 VARIOUS FEATURES OF CLOUD DEPLOYMENT MODELS SUMMARY 16
TABLE 4.1 IDENTIFY REQUIREMENTS 28
TABLE 4.4 IDENTIFY VULNERABILITY AND ATTACKS 31
TABLE 4.5 IDENTIFY SECURITY REQUIREMENTS....35
TABLE 4.6 IDENTIFY ASSOCIATED ASSETS WITH REQUIREMENTS....38
TABLE 4.7 IDENTIFY RISKS ASSOCIATED ASSETS....42
TABLE 4.8 IDENTIFY RISKS ASSOCIATED WITH THREATS 43
TABLE 5.1 MAPPING OF SECURITY REQUIREMENTS WITH SECURITY SERVICES 45
TABLE 5.2 SYMMETRIC CIPHERS....49
TABLE 5.3 ASYMMETRIC CIPHERS....49
TABLE 5.4 HASH FUNCTIONS....49
TABLE 5.5 SLA EXAMPLE 54
TABLE 5.6 MAPPING OF SECURITY MITIGATION TECHNIQUES WITH SECURITY SERVICES....54
TABLE 5.7 IMPACT ANALYSIS OF CRYPTOGRAPHIC MITIGATION TECHNIQUES 56
TABLE 5.8 SECURITY DESIGN CONSTRAINTS....57
TABLE 5.9 SECURITY DESIGN TEMPLATE....59

ABSTRACT

Cloud computing popularity over traditional software and hardware has increased considerably over last decade. The on-demand, scalable, multi-tenant and measured service provisioning is making Cloud Computing and Services a hot favorite among not only industry big players but small to medium scale SMEs. As companies, customers and end users are moving their business and data to cloud, more concerns over Cloud Security is getting raised. As more and more software vulnerabilities are getting exploited by attackers and malicious users, huge cost is been spent in security attacks mitigation. This requires integration of security into application development lifecycle and adaptation of security related activities in software engineering practices and methodologies.

In this thesis, we propose a Design Methodology using Security Engineering Framework that involves converting security requirement and threats into design decisions to mitigate identified vulnerabilities. A security design template is prepared considering various constraints and mitigation techniques that will help in later stages of design process.

CHAPTER 1

1 Introduction

1.1 Introduction

The progress of IT Industry and its applications have resulted in reduced time for software development life cycle. The horde to bring customer centric product in market with increased quality has opened avenues to re-think existing software processes. With the advent of big data and cloud computing, major industry shift is seen toward these technologies. Most of the industry majors are considering Cloud for application development and services, as it provides ubiquitous, on-demand scaling of infrastructure, reduced expenditure and effort on infrastructure management. This software environment change has caused security of software and customer data to become one of the areas to be researched and enhance the existing software engineering practices and methodologies.

Cloud Security opens a new paradigm, worth to study, research and propose enhancement in existing processes. Until now, security measures used to be considered at the time of design phase, which increased cost of design changes and usually such approach resulted in design solutions which were not optimal or may have software vulnerabilities left at open at large [2]. This enforces elicitation and requirement analysis with Security Risks and Threats assessment in mind and design software specifically mapping these security requirements.

Gathering security requirement and development of guidelines for software or system has been focused by many government institutions, industry players and researchers now. National Institute of Standards and Technology (NIST), under Information Technology Laboratory (ITL), USA, uses various publications (IEEE, ISO, IEC, ITU-T, ETSI) to promulgate computer security standards and guidelines. Security Standards are published under Federal Information Processing Standards (FIPS) and Security Guidelines and Recommendations are published as SP-800 series publications [3][4][5]. Non-profit organizations, task force, open forums like Cloud Security Alliance (CSA), Distributed

Management Task Force (DMTF), Storage Networking Industry Association (SNIA), Open Grid Forum (OGF), Open Cloud Consortium (OCC), TM Forum, Object Management Group (OMG), Association of Retail Technology Standards (ARTS) are working diligently to provide security guidelines, methods and standards for Cloud Security.

Various independent works from academics on Cloud Security, like, framework for development of secure software [1], proposes gathering and analyzing security requirements and threats, mapping these to Firesmith high level requirements [6], converting these requirements to design decisions to mitigate threats, its implementation and testing.

The work in this thesis, targets to propose Design Methodology for Secure Cloud Systems assuming security requirement elicitation is done following well defined process identified in [1].

1.2 **Motivation**

As per the Elastica Shadow Data Report, 2015[7] for SaaS applications, millions of files were found to be exposed to either direct compliance violations, possible intellectual property leaks, or generic risk exposures. Elastica ingenious metric to measure cloud data exposures for companies using SaaS storage providers (Box, Google Drive) was $13.85M per business. Report also stated 25% of user files are broadly shared, in which 12.5% contains sensitive data. Of this sensitive data 54% have Personal Information (name, phone numbers and social security number), 31% healthcare related records, and 15% payment related information.

Total number of docs shared in cloud is increasing exponentially every year (60% [7]). Security Violations is also seen increasing in same ratio (1.8% (Q4 2014) -> 3.2% (Q2 2105)). 1.34% anomalous user behavior were observed, of this 43% accounted from too many logins from infeasible locations and 40% from anomalous frequent session, pointing out possible cyber-attacks by bots or malicious users.

Office 365, Dropbox, Google Drive are popular and common SaaS apps by usage, while Mindomo, Airbrake.io and formmule are fast catching up. Skype, box, You Tube accounts for most network traffic, while, Office 365, Slideshare, centraldesktop accounts for longest session times [7]. Trend of application usage provides increased avenues of exploitation areas to attackers.

All above data points to identifying and dealing with security risks in situ, while developing Cloud Platform or using Cloud as SaaS for application development. Hence, use of Security Engineering for SDLC plays a major role. It includes identification, prioritization and management of Security Requirements, Design and implementation and testing.

Major security goals, protect Confidentiality, maintain Integrity and ensure Availability (CIA Triad), must be met while designing a secure platform or application.

1.3 Related Work

Security Model achieving multi-level security goals was developed by Bell-LaPadula (BLP model) [12] in 1973. It was also adopted as the foundation of Trusted Computer System Evaluation criteria by US Department of Defense. This model proposed set of rules to control information flow in order to protect disclosure of sensitive data. The combination of hierarchical and non-hierarchical categories were used to enforce mandatory access policies. This model provides the only protective mechanisms to overcome security breach and since there security requirements and design are not considered during system design it is not capable to cater todays security threats and attacks.

Using Security Patterns and Design Models, PICO and UMLsec [13] proposed services to handle different services (subscription services, presence services, instant message services) for securely forwarding instant message service to different users. Again this covers just a part of development processes.

Security Quality Requirement Engineering (SQUARE), provides systematic process for Security Requirements elicitation, categorization, and prioritization for an

organization. As per business goals, safety and security goals are laid down, various assets and threats are identified and then evaluated, using Use Case, Attack Trees and Architectural diagrams. Security requirements are elicited, categorized and prioritized and verified. Here, environment and device constraints are not considered.

Security Requirement Engineering Process (SREP), **Daniel M et.al** embeds concept of Common Criteria that helps threat modelling and security object mapping. Here again environment and device constraints are not considered.

Framework by Haley et al. proposes iterative process of refining security requirement but lacks requirement prioritization and environment and device constraints consideration.

As trusted computing technologies are emerging and developing at exponential rate, there is a need to address multilevel security issues and design principles. Application of security engineering framework targeting Cloud and its application is need of hour. Though enough literature can be found now which take care of Cloud Security special use cases, but no framework is there to address this concern as a unified process. One such work is providing effective mechanism for data deletion in Cloud System, which becomes challenging because of multi-tenancy environment of Cloud system.

A Secure Erasure Code-Based Cloud Storage System with Secure Data Forwarding, Hsiao-Ying Lin and Wen-Guey Tzeng, June 2012 [9], addresses problem faced on data storage in a third party's cloud system. Data confidentiality can be protected using general encryption schemes, but this limits functionality of storage systems. The paper proposes re-encryption scheme with threshold proxy and integration with decentralized erasure code with this scheme, such that a secure distributed storage system is created. This allows users to forward his data to other storage servers and retrieving it back.

1.4 Problem Statement

Designing and deployment of secure Cloud systems requires integration of security requirements during development lifecycle and adapting current software engineering process and methodologies to incorporate specific security related activities. Developers

enforce security related measures usually during design phase or later in the processes, which causes inclusion of various unnecessary security related constraints in the system. Over deployed security mechanism may lead to over constrained system and increase the cost.

Study of various work over cloud security, provides ample availability of guidelines but almost no work on proposing a generic and common Design methodology for Secure Cloud Systems. This must be evidently done after ensuring that security requirement are properly elicited using either Use Case Analysis or various stakeholders viewpoints, Threats are identified using known techniques (Wide band Delphi methods) which may require discussion and evaluation with SME (Subject Matter Experts) or some verified database is used.

Therefore, the problem of this thesis is as follows: Design a Methodology for Secure Cloud system which can effectively mitigates the vulnerabilities, threats and attacks expected on system.

1.5 Scope Of The Work

The scope of this work includes understanding the need of security engineering for present Cloud Systems. Identify Cloud Systems safety and security goals and classify them into functional and non-functional requirements. Identify various actors, assets, threats corresponding to the functional requirements. Through threat modeling prioritize threats and attacks on the system and then prioritize the security requirements.

It then proposes the Design Methodology that will effectively map each security requirement and attack with the mitigation techniques. This will take into account various design constraints. Using security design template, a security design decision will be reached.

1.6 Approach

Any proposal for any system requires deep understanding of the system. We first understand the Cloud Systems and applications, its functioning and various usages. We take understand functioning of one the Cloud Services – Storage as a Service.

As cloud is interfacing with its users over Internet, all the security risks applicable to any web application is applicable to Cloud System and its applications as well. Keeping this in mind we first do security requirement elicitation of a system. We first identify all the actors using use case analysis and various stakeholders' viewpoints. Categorize various functional and non – functional requirements with respect to identified actors.

We then list down all possible security vulnerabilities on the system. Prepare exhaustive lists of attacks possible on the system because of these vulnerabilities. We also identify all the assets in the system which is affected because of these security vulnerabilities.

These attacks are then mapped to functional and non-functional requirements. Various security requirements as mentioned by Firesmith in his Journal for Object Technology [6] are then mapped to these functional and non-functional requirements.

Using CRAMM, threat modeling and risk analysis is done which further help us to prioritize security requirements.

Security Design kicks in after this step. Mapping of attacks to security requirements is done. We list down all security design constraints and identify and prioritize design attributes. We make a security design template which helps in making optimal security decision.

With the help of case study of a IaaS application, we discuss the applicability and benefit of this methodology.

1.7 Organization Of Thesis

This thesis follows below mentioned organization of chapters:

Chapter 2 discusses Cloud Architecture, SaaS architectural details and security and privacy issues in Cloud System.

Chapter 3 discusses about Security Requirement Engineering method and framework proposal for Cloud Security.

Chapter 4 is dedicated to Cloud's Security requirement elicitation, vulnerability and associated threats and attacks identification and security requirement prioritization.

Chapter 5 proposes Design Methodology on Security Requirements elicited in previous chapter.

Chapter 6 presents a Case Study analysis and discusses the applicability of proposed design methodology.

Chapter 7 discusses the conclusion and paves way for Future Work.

CHAPTER 2

2 Background Study

2.1. Cloud Computing

Today the term “cloud” is widely adopted by the internet community with many different meanings, from that of “well managed” data- center, to that of the hardware - software structure supporting the concepts of utility computing and elastic computing introduced by Amazon [12].

Based on underlying infrastructure Cloud systems are classified into various models:

- **Public Cloud**: Public Cloud offers the computing resources for the general public over the internet via Web applications or Web browsers either for free or on pay- per- use license policy.
- **Private Cloud**: It offers the infrastructure to be used by only one organization which can be located on the premises of the Cloud Service Provider (CSP). These are used in private networks and hence restrict the unwanted public access to the data that is used by the organization.
- **Hybrid Cloud**: Hybrid Cloud is a combination of Public Cloud and Private Cloud. So through its implementation an organization can benefit from the advantages of both Clouds.
- **Community Cloud**: Community Cloud offers to share the resources and hardware between organizations that have similar needs. It is a Private Cloud for a community.

Each model has its own merit and demerits such as in case of Public Cloud one need not to buy any hardware but the data is stored off- premise so no control over it, in case of Private Cloud everything is available in the premise but it is very costly as one has to buy

all hardware and resources, in Hybrid Cloud one can put its critical information within the premise but it is less efficient than Public Cloud and in Community Cloud efficient use of hardware is done and is cost effective but it is also less efficient than Public Cloud.

TABLE 2.1 VARIOUS FEATURES OF CLOUD DEPLOYMENT MODELS SUMMARY

Deployment Model	Managed By Infrastructure	Owned By	Infrastructure located At	Accessible and Consumed By
Public	Third party provider	Third party provider	Off-premise	Untrusted
Private	Organization	Organization	On-premise	Trusted
			Off-premise	
	Third party provider	Third party provider	On-premise	
			Off-premise	
Managed	Third party provider	Third party provider	On-premise	Trusted/Untrusted
Hybrid	Both organization and third party provider	Both organization and third party provider	Both on-premise and off-premise	Trusted/Untrusted

The advantages of using Cloud Computing include:

i) Reduced hardware cost

ii) Reduced maintenance cost

iii) Global accessibility

iv) Flexible and highly automated processes, with low customer concerns on software up-gradation, for eg... [13, 14].

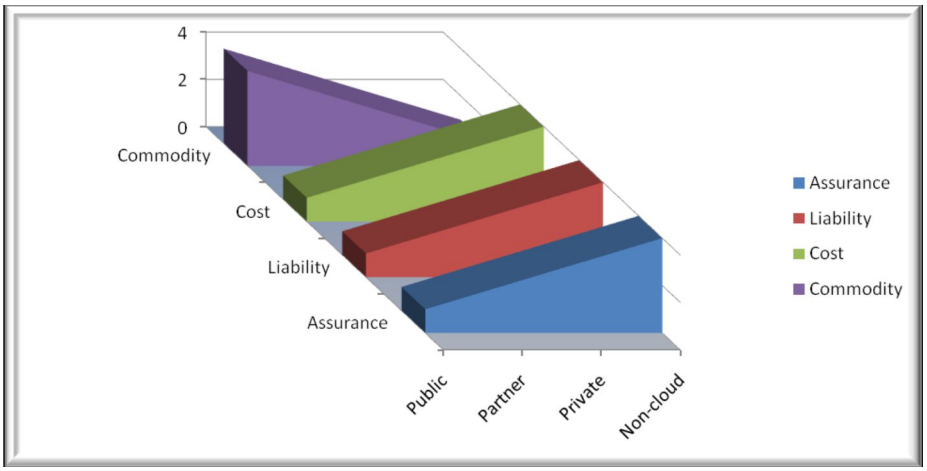

FIGURE 2-1 FEATURES OF PUBLIC, PARTNER AND PRIVATE CLOUD

Depending upon the service level provided, Clouds can be classified as:

- **Infrastructure as a Service** (IaaS): The computing infrastructure like servers, network equipment's and software are provided as on- demand where the customer can install operating system images with applications to create their own customized environment. The CSP owns the hardware and it is responsible for housing and their maintainenace. Various Clouds exist in this category, such as Rackspace, Amazon EC2, Google Compute Engine, GoGrid.
- **Platform as a Service** (PaaS): The computing platform is provided on demand by CSP 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 and testers. Its purpose is to reduce the cost, complexity of buying, managing underlying hardware and software components. Clouds in this category are GAE, Force.com, and Windows Azure Compute.
- **Software as a Service** (SaaS): The Cloud Service User (CSU) access the application software installed and maintained by CSP at providers end. In this implementation, deployment is abstracted from the user and only a 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. Clouds in this category are MS Office 365, Quickbooks online, Salesforce.com

Cloud comprises of many technologies – networks, operating systems, databases, virtualization, transaction management, resource scheduling and load balancing, concurrency control and memory management. All security concerns in these are applicable to cloud system. Network which connects the systems in cloud must be secure. Secure mapping of virtual machines to the physical machines is required. Data security via encryption and appropriate policy enforcement for data sharing must be in place.

Also, secure resource allocation and memory management must be there. Intrusion detecting systems using data mining techniques must be deployed.

Six specific areas which needs to be considered in cloud computing are

1. Security of data at rest
2. Security of data in transit
3. Users/applications/processes authentication
4. Isolation of resources of different customers
5. Legal and regulatory issues and mitigation
6. Incident response

This is shown in Figure 2-2.

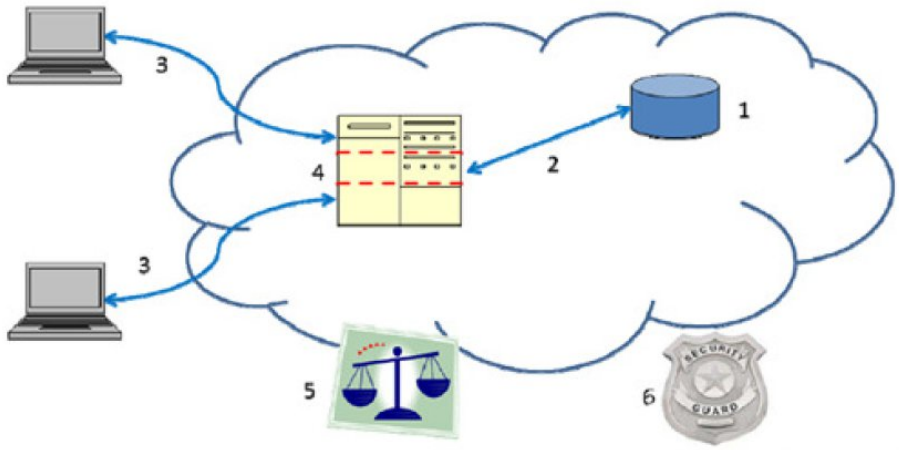

FIGURE 2-3 AREAS OF SECURITY CONCERNS IN A CLOUD SYSTEM

Encryption mechanisms are best for securing data at rest as well as in transit. A customer can be ensured of its data source and destination if strong authentication and integrity protection mechanisms are in place. Cloud Service provider and customer real time communication in ensured by trusted computing group's (TCG's) IF-MAP standard.

The customer's identity management system can notify cloud provider in real time when access privilege of user is revoked/modified. Intentional/inadvertent access to sensitive

information of other cloud customer can be controlled by use of trusted platform module hardware based verification of hypervisor/VM integrity.

Security implications of legal and regulatory issues in a cloud system in undeniable. Care must be taken for compliance, data retention and destruction, data security and export, auditing and legal discovery. For data retention and deletion trusted storage and platform module access techniques must be used.

In today's scenario, customer needs to plan for cloud provider security breaches and user misbehavior. The integration of different security systems can be enabled by IF-MAP (Metadata Access protocol) by trusted computing group (TCG) Specification, which provides real time notifications user misbehavior and such incidents.

2.2. Storage as a Service (SaaS)

Large companies provide Storage Infrastructure on rent to smaller companies or individual. In terms of cost (personnel, hardware and physical Storage) and to manage backups, SaaS is very convenient. This is targeted by SaaS vendors in the enterprise for developing secondary storage applications.

A network administrator can relieve himself of tasks like maintaining large tapes library and/or arranging offsite storage supports. He can specify data and frequency of backup. Under SLA, storage space is rented by the smaller companies on a cost-per gigabyte-stored/cost-per-data-transfer basis. The company's data is then automatically transferred at the specified time to service provider storage using WAN/Internet. One of the advantages offered by SaaS provider is during corruption/loss of user data, a copy can always be requested.

There are many traditionally hosted or managed service providers (MSP) who offer block or file storage. Other emerging solutions are like Amazon S3 service resembles flat databases designed to store large objects. Cloud storage can be accessed by FTP, NFS/CIFS, WebDAV, or block protocols remotely.

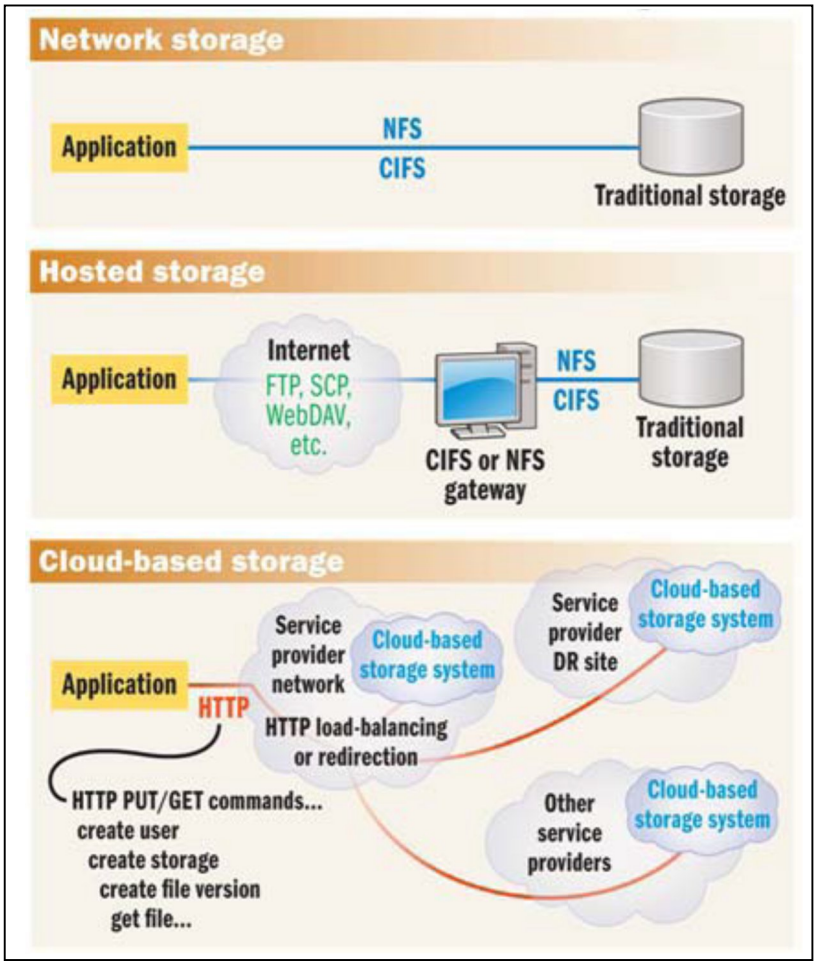

FIGURE 2-4 EVOLUTION OF CLOUD STORAGE

Types of Cloud Storage Systems

Choosing the right kind of storage is of utmost importance among available cloud storages. Each has their own advantages and limitations.

1. Object Storage Systems
2. Distributed File Storage System
3. Relational Database Storage Systems (RDS)

Storage Access Methods

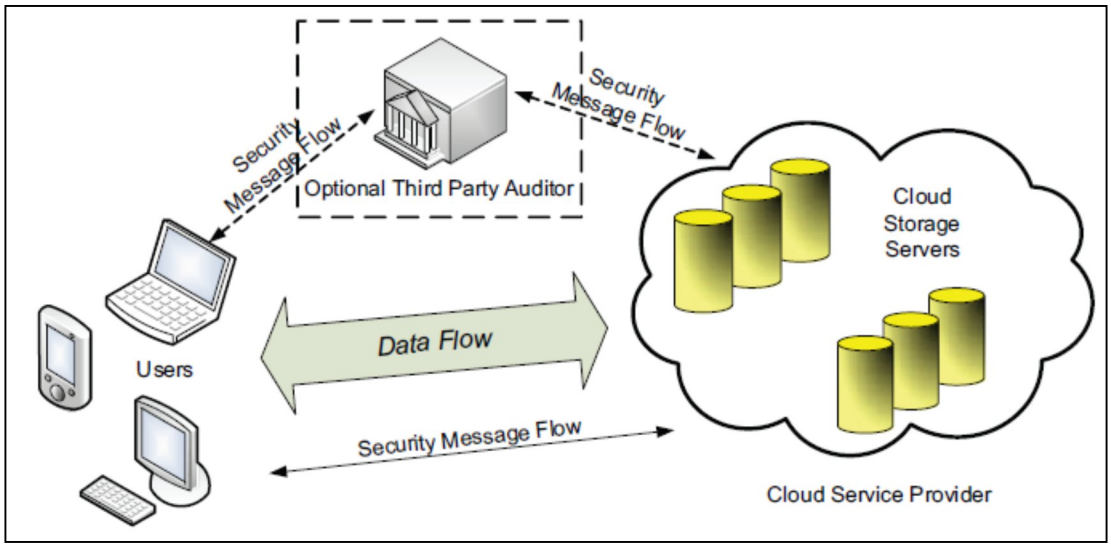

FIGURE 2-5 EVOLUTION OF CLOUD STORAGE

As shown by the Fig2.4., there are three ways to access computing storage space –

- through Web services.
- block-based (SAN or iSCSI).
- file-based (CIFS/NFS).

Enterprise application designs use Block and file-based access to enable greater control of availability, performance, security. To access data Web services interfaces like representational state transfer (REST) and SOAP and are used. Even though most flexible, it has performance implications.

An enterprise cloud may provide all three access methods for storage to support different application architecture.

CHAPTER 3

3 Security Engineering

Security engineering is a complex process. It contains different security-related activities - security requirements identification, prioritizing and management of security requirements, security design, implementation of security mechanisms and comprehensive security testing. Appropriate design decisions are facilitated by appropriate security requirements elicitation. The design phase defines how the requirements can be implemented in a given environment. Overall structure of the software from a security viewpoint can be seen. This phase targets to get a systematic and well organized structure of security functionalities and design decisions.

3.1 Proposed framework for security engineering

The overall security engineering process (SEP) for a Cloud System is shown in below figure:

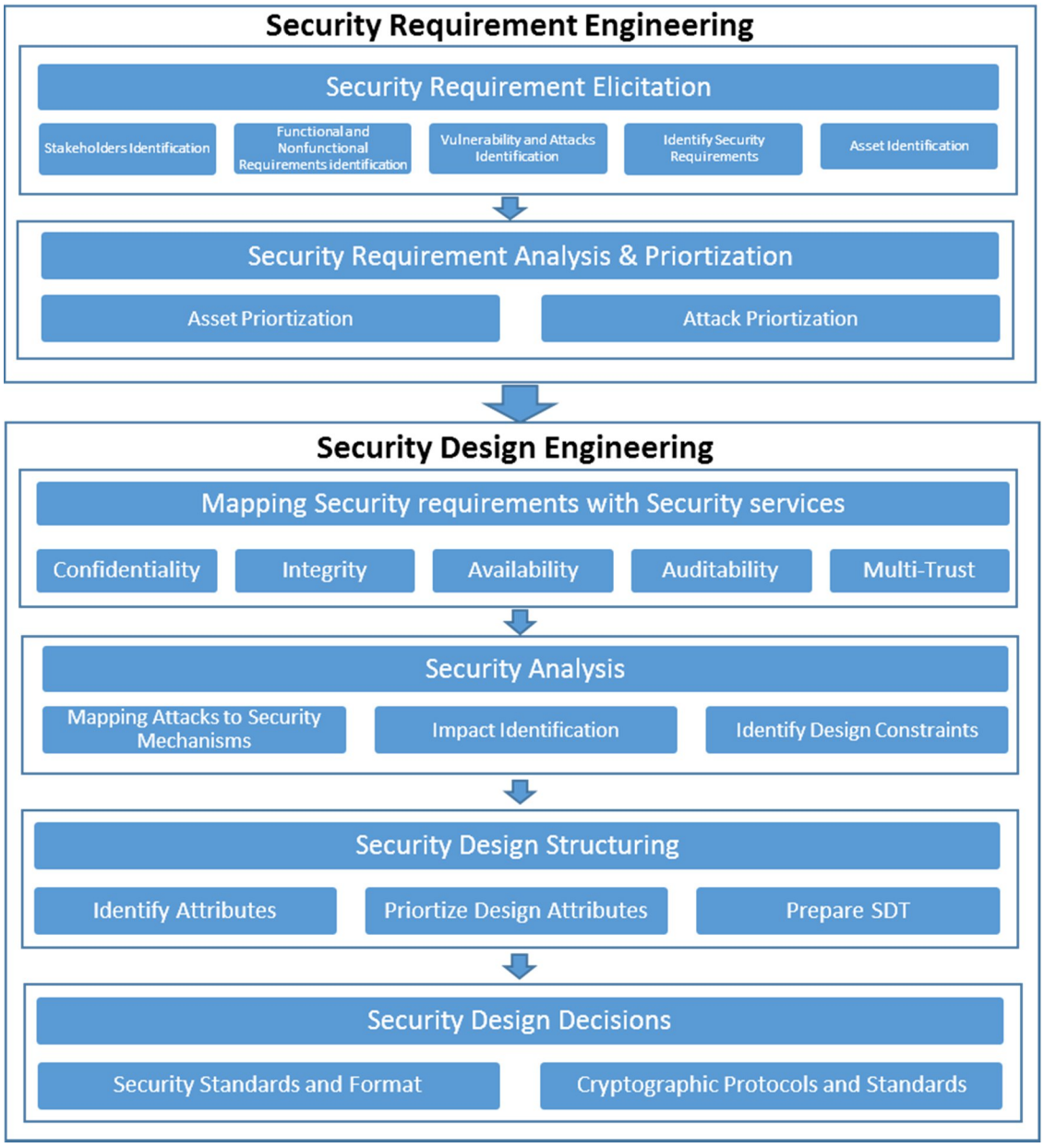

FIGURE 3-1 FRAMEWORK FOR SECURITY ENGINEERING PROCESS

In this framework proposal is made to design a framework taking view of stakeholders and design constraints.

3.2 Security Requirement Engineering

Security requirements are discovered, analyzed and managed in this phase. It consists of four different stages:

Step 1: Security Requirement Elicitation

This includes

1. Identification of various stakeholders using view-point analysis. Direct actors, the one directly interacting with the system such as human, software system or hardware devices are identified. Indirect actors are for example, software developer, people who regulate application domain.
2. Identify functionalities of actors and determine associated non-functional requirements.
3. Identify the vulnerabilities and attacks.
4. Identify Security Requirements associated with each attack
5. Identify assets affected by vulnerabilities and attacks.

Step 2: Security Requirement Analysis & Prioritization

Check for completeness of the elicited security requirements. They should mitigate all possible threats on the functionality of the system.

1. Prioritize assets getting most affected by the security attacks.
2. Prioritized security requirements on measure of threat, vulnerability and risk.
   Assign value to corresponding vulnerability (0-100) and Impact (1-10) using CRAMM.
   a. Calculate threat. *Threat = Vulnerability X Impact*.
   b. Estimate value of Risk.

3.3 Security Design Engineering

In this phase software structure is designed to realize the specifications of the system. In this phase various measure for example cryptographic techniques, coding standard identification, all available techniques and required standards to be followed etc. are mapped for mitigating identified security requirement. In this phase, any bad decision will lead to design flaws making system vulnerable to attacks.

Step 1: Security requirements mapping with security services**.**

There are 12 Firesmith Requirements – Identification, Authentication, Authorization, Immunity. Integrity, Intrusion, Nonrepudiation, Privacy, Security, Survivability, Physical Protection and System Maintenance [6]. Prioritized Firesmith security requirements are mapped to security services like confidentiality, integrity, availability (CIA Triad), auditability and multi-trust. Auditability and Multi-trust are two new services added. This later helps in specifying security mechanisms for specific security requirements.

Step 2: Security design analysis

Prioritize threats and affected assets are defined in this step. It contains two sub steps:

1. Attacks are mapped to Security Mechanisms
2. Security threats are mapped to available cryptography techniques. Applicability of attacks if it resists is mentioned by ‘N’ else ‘Y’. Impact of attack is accordingly evaluated.
3. Identifying security design constraints

   All the design constraints for a system must be identified in this phase. Constraints like channel capacity, bandwidth, power consumption, throughput or computational constraints.

Step 3: Security design structuring

In this step, different design attributes are identified. It consist of two sub steps:

1. Identify design attributes and prioritizing them

   Design attributes like cost, choice of implementation platform, applicability of mitigating techniques are identified in this step. Symmetric key algorithms - AES, 3DES are suitable for confidentiality service requirements, as they are 1000 times faster than asymmetric key algorithms like RSA.

2. Preparation of security design template(SDT)
   Security design template to take care of each requirement is prepared. This will store each specification of the design constraints and mitigation techniques.

Step 4: Security design decision

In this final step, knowledge based approach to find best suitable protocol depending upon the design attributes applied in Security Design Template.

3.4 Security Implementation and Testing

Implementation of all functionalities is carried out based on design decision made in security designing phase. In testing of implemented security functionality, vulnerability scan and assessment, security assessment and audit, and review are performed.

CHAPTER 4

4 Clouds System Security Requirements

In this section, we will elicit security requirements for a Cloud System. The process followed is similar as discussed in previous chapter. We first identify the stakeholders in a cloud system. Usually this identification is done using view point analysis of various stakeholders. We then define actors for all possible Use Cases. Corresponding functional and non-functional requirements are then elicited and then mapped against each actors.

After this important step of vulnerability and corresponding possible attacks identification is done. In this study, [15] is taken as reference. The attacks are then mapped against each actor's functional and non-functional requirements. This is further mapped with Firesmith security requirements [6].

Using CRAMM methodology, impact and vulnerabilities are classified. This is used in calculating Threat Rating as Vulnerability X Impact. This is then used for Risk Assessment.

Security vulnerabilities/threats affecting various Assets are identified and prioritized.

Step1. Identify the Stakeholders

Direct Stakeholders are:

Cloud Customer: Cloud customer is the one who stores data over the Cloud and pay for the services provided by Cloud Service Provider.

Cloud User: Cloud user is the one who has access to view the data shared with Cloud customer. It can be a subscriber or a non-subscriber of Cloud services.

Cloud Service Provider (CSP): who owns, manage and operate the Cloud system to deliver services. It also receives the payment from Cloud customers for services provided.

Cloud Service 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 Stakeholders are:

Security Administrator: who maintains security related functions.

Auditor: who manages the audit and log details.

Government: who ensures Service Level agreement are followed. There is no loss of Personal Identifiable Information (PII) and service provider losing same will be punishable

Step2. Identify Requirements

Functional requirements are one which define the behaviour of a system or its functions while non-functional requirements are one which define constraints on design or implementations like reliability, quality or security. In this step we identify functional and non-functional requirements

I. Identify the Functional Requirements.

II. Identify the Non- Functional Requirements.

TABLE 4.1 IDENTIFY REQUIREMENTS

Actors	*Functional Requirements*	*Non Functional Requirements*
Cloud Customer	*SaaS* 1. Registration & Login 2. Update Login Details 3. Store data into Cloud	1. Reliability 2. Less Response Time 3. Scalable 4. Correctness 5. Consistency

	4. Manage automatic backup 5. Manage Sharing with cloud user 6. Download data stored in the cloud 7. Select storage location 8. Make payment for services used 9. Maintenance of identity management system 10. Identity management system 11. Authentication platform management (including enforcing password policy) 12. Data and Traffic monitoring for security risk avoidance PaaS 1. Maintenance of identity management system 2. Identity management system 3. Authentication platform management (including enforcing password policy) IaaS 1. Maintenance of identity management system 2. Identity management system 3. Authentication platform management (including enforcing password policy) 4. Guest OS patch and hardening procedures management 5. Guest security platform Configuration management 6. Guest systems monitoring 7. Security platform management and configuration (Firewall rules, IDS/IPS tuning, etc) 8. Log Collection & security monitoring	6. Recovery 7. Lawfulness of content 8. Compliance with data protection law 9. Personnel Security 10. Supply Chain Assurance
Cloud Users	1. Registration & Login 2.View shared data based on permission 3. Submit request to join the group 4.Unjoin a group	1. Reliability 2. Less Response Time 3. Scalable 4. Correctness 5. Consistency 6. Recovery
Cloud Service Provider	SaaS 1. Manage Cloud customer's Account 2. Manage Customer Data	1. Reliability 2. Integrity 3. Recovery

	3. Manage cloud hardware and software 4. Receive cloud usage payment 5. Maintain SLA 6. Data Processing 7. Physical support infrastructure, security and availability 8. OS patch management and hardening procedures 9. Security platform management and configuration (Firewall rules, IDS/IPS tuning, etc) 10. Systems monitoring 11. Log Collection & security monitoring 12. Define Backup Strategy PaaS 1. Physical support infrastructure, security and availability 2. OS patch management and hardening procedures 3. Security platform management and configuration (Firewall rules, IDS/IPS tuning, etc) 4. Systems monitoring 5. Log Collection & security monitoring 6.Multi-Tenanted Application separation 7. Sandboxing of application IaaS 1. Physical support infrastructure, security and availability 2. Host Systems - Hypervisor, virtual firewall)	4. Performance 5. Data and Traffic monitoring for security risk avoidance 6. Personnel Security 7. Supply Chain Assurance 8. Scalability 9. Response Time 10. Restricted access to concerned cloud customer enterprise
Cloud Service Integrator	1. Registration & Login 2. Combine cloud based & in-house services	1. Reliability 2. Integrity 3. Recovery 4. Performance

Step3. Identify the Threats

All possible vulnerabilities possible on a cloud system are identified. ENSIA, reports published by various SMEs on Cloud are used[15]. A repository of threats is created and then threats are mapped against vulnerability.

TABLE 4.2 IDENTIFY VULNERABILITY AND ATTACKS

Sl.No.	Vulnerabilities	Vulnerability Description	Possible Attacks/Security Risks
1	AAA	Unauthorized access to resources, privileges escalation, and impossibility of resources misuse tracking - can be facilitated if there is poor Authentication, Authorization and Accounting (AAA) System: 1. Cloud Customer storing cloud access credentials insecurely 2. Insufficient available roles 3. Credentials storage on a Transitory Machine	1. Password based Authentication Attacks (Trojan to steal corporate passwords) - Password_Cracking 2. Impersonate 3. Sniffing 4. Tampering 5. Social_Engg 6. Disclose_Data 7. Malicious_Code 8. Repudiate 9. Data_Theft 10. Human_Error 11. Password_Reuse 12. Multilocation_Dataplacement 13. Insider 14. MITM
2	User Provisioning	No customer control on provisioning process Inadequate verification of customer identity at registration Synchronization delay between cloud system components Multiple, unsynchronized creation of identity data	1. Replay attacks 2. Impersonate 3. Data_Theft 4. Multilocation_Dataplacement 5. DoS_DDoS 6. MITM
3	De-Provisioning Vulnerabilities	Time delays in roll-out of revocation, de-provisioned credentials are still valid.	1. Replay attacks 2. Impersonate 3. Data_Theft 4. Multilocation_Dataplacement 5. DoS_DDoS 6. MITM
4	Remote Access to Management Interface	Weak authentication of responses and requests allow vulnerabilities in end-point machines(Cloud User) to compromise cloud infrastructure(single cloud customer or cloud provider)	1. MITM 2. DoS_DDoS 3. Sniffing 4. VM_Threat 5. Impersonate
5	Hypervisor Vulnerabilities	The physical resources and VMs can be fully accessed by Hypervisor.	1. VM_Threat 2. Data_theft 3. Password_Reuse 4. Multilocation_Dataplacement 5. DoS_DDoS 6. Malicious_Code
6	Lack of Resource Isolation	Unauthorized access to shared resources. Mostly seen in case of IaaS, where there is a possibility of mapping of virtual drive to its own and use the data. It's one of the impact of Hypervisor Vulnerability. Assets inside cloud facility can also be manipulated. Can cause direct financial damage.	1. DoD_DDoS 2. Data_Theft
7	Lack of Reputational Isolation	One customer maliciously using another customer data and impacting its reputation	1. Sabotage 2. Tampering

8	Communication encryption vulnerabilities	Searching becomes very expensive operation when done on encrypted data. There is threat of reading data while in transit, because of poor authentication, acceptance of self-signed certificates.	1. MITM 2. DoS_DDoS 3. Sniffing 4. Impersonate 6. Data_Theft
9	Lack or poor encryption of archives and data in transit	Data in Transit or in rest (databases/archives), un-mounted VM images, forensic logs and data, sensitive logs are at risk	1. Data_Theft 2. Sabotage 3. Tampering
10	Poor Key Management procedures	Different kinds of keys - file encryption keys, Session keys for data in transit (SSL keys), Cloud Provider identification key pairs, customer identification key pairs, authorization tokens, and revocation certificates - are required to be managed and stored. Hardware Security Module is not possible because of distributed, ever scaling infrastructure of cloud system. Key Management interfaces get exposed to Public internet.	1. Data_Theft 2. Sniffing 3. Impersonate 4. Sabotage 5. Tampering
11	Key Generation: Low entropy of Random Number Generation	An attacker can guess encryption keys generated on other virtual machine as source of entropy used for random number generation can be similar. This needs to be taken care during system design.	1. Data_Theft 2. Impersonate 3. Sabotage 4. Tampering
12	Lack standard technologies and solutions	Customer faces risk of their data to be 'locked-in' to a cloud provider. It becomes big risk if Provider ceases its operations. Also managed security services and external security technologies become difficult to be used.	1. Lock-In
13	Inaccurate Modeling of Resource Usage	Resources are provisioned statistically using standard algorithms - Token Bucket, Fair Queuing and Class Based Queuing. Overbooking or over-provisioning can lead to resource wastage by cloud provider. Poor classification of resources based on job or packets can also cause resource exhaustion	1. Resource_Exhaustion
14	No Control On Vulnerability Assessment Process	As per ToU customers are made responsible to secure infrastructure security elements. This may cause serious security problem. Restrictions of port scanning and Vulnerability Testing must be done for Vulnerability Assessment	1. Governace_Loss
15	Possibility of Internal (Cloud) Network Probing	Port Scans or other tests performed by cloud customer within internal network on other cloud customers.	1. Malicious_Probes_Scans 2. Data_Leakage 3. Data_Theft 4. Sabotage 5. Tampering
16	Possibility of co-resident checks	Attackers can determine shared resources among cloud customers because of lack of resource isolation	1. Side-Channel Attacks 2. Malicious_Code
17	Lack of Forensics Readiness	Providers not providing services and terms to enable/improve forensic readiness.	1. Operational_Logs_Compromise 2. Security_Logs_Compromise
18	Sensitive Media Sanitization	Because of shared tenancy, data cannot be entirely wiped or physical resource be destroyed, as other customer may still be using it.	1. Data_Deletion
19	Responsibilities Synchronization or Contractual Obligations External to Cloud	Customers are not aware of the responsibilities assigned to them. Archive Encryption is often left on Cloud Providers even though in terms of the contract mentions no such responsibility has been undertaken.	1. Governace_Loss

20	Cross-Cloud applications creating hidden dependencies	Service Supply chain hidden dependencies when third parties or subcontractors or the customer company have been separated from the service provider or vice versa.	1. Governace_Loss 2. Supply_Chain_Failure
21	SLA Clauses with Conflicting Promises to different Stakeholders	Service Providers puts conflicting promises by other clauses or other provider clauses.	1. Governace_Loss 2. Customer_Provider_Hardening_Process
22	Audit or Certification not Available to Customers	Any open source sw eg hypervisor Xen, used by CP doesn't mention whether it has met Common Criteria Certification or any such certification	1. Governace_Loss 2. Compliance_Challenges
23	Inappropriate Certification Schemes in cloud Infrastructures	Security Vulnerabilities are likely to be missed since Certification scheme are not under Cloud Specific control	1. Governace_Loss 2. Compliance_Challenges
24	Inadequate Resource Provisioning and Investments in Infrastructures	Any failure will cause long term impact	1. Resource_Exhaustion
25	No proper Policies for Resource Capping	When resource usage is unpredictable, flexible and configurable settings of limits on resources is required at customer and CP.	1. Resource_Exhaustion 2. DoS_DDoS
26	Multiple Jurisdiction data storage and Lack of Transparency about this	Data Mirroring for delivery over edge nw and not location information to customer about data storage can introduce certain levels of vulnerability.	1. Legal_Issues
27	Lack of Jurisdictions information	Data stored/processed in high risk jurisdiction can be vulnerable by forced entry confiscation. Customers cannot take any measures to avoid such scenarios in absence of information.	1. Legal_Issues
28	SLA usage, transparency and completeness	SLA not containing enough information on terms of Usage	1. Lock_In 2. Governace_Loss 3. Compliance_Challenges
29	Lack of Security Awareness	A Cloud customer may not be aware pf security risks he may face on migrating to Cloud - loss of control or vendor lock-in. exhausted resources. CP may not be aware of measures to take to mitigate these risks.	1. Social_Engg
30	Unclear Roles and Responsibilities	Roles and responsibilities are not clearly defined	1. Governace_Loss 2. Customer_Provider_Hardening_Process 3. Priviledge_Abuse
31	Poor Enforcement of Role Definitions	Lack of segregation of roles may lead to access of entire Cloud system to a person within CP, making system highly vulnerable.	1. Governace_Loss 2. Priviledge_Abuse
32	Need-To-Know Principle Not Applied	Unnecessary data access must be avoided to anyone.	1. Priviledge_Abuse
33	Inadequate Physical Security Procedures	System facilities designed without enough security procedures and enough vetting of personnel.	1. Priviledge_Abuse 2. Social_Engg 3. Backup_Lost_Stolen 4. Unathorized_Physical_Access

34	Mis-configuration	Human Error/Untrained Administration/inadequate security baseline and hardening procedure application	1. MITM 2. DoS_DDoS 3. Sniffing 4. VM_Threat 5. Impersonate 6. Network_Issues 7. Priviledge_Abuse
35	System/OS Vulnerabilities	Inherent design/implementation issues in system causing exploitable security flaws.	1. MITM 2. DoS_DDoS 3. Sniffing 4. VM_Threat 5. Impersonate 6. Network_Issues 7. Priviledge_Abuse 8. Operational_Logs_Compromise 9. Security_Logs_Compromise
36	Lack of/Poor Untested Business Continuity/ Disaster Recovery Plan	Not enough planning done for disaster recovery in case of extreme scenarios.	1. Natural_Disaster 2. Network_Issues
37	Unclear Asset Ownership	Asset ownership is not clear	1. Governace_Loss
38	Poor Provider Selection	Customer doesn't investigate thoroughly before Provider Selection	1. Lock_in 2. Cloud_Service_Termination 3. Supply_Chain_Failure
39	Lack of Supplier Redundancy	Not enough evaluation of 3rd party suppliers	1. Lock_in 2. Resource_Exhaustion 3. Cloud_Service_Termination 4. Supply_Chain_Failure
40	Application Vulnerabilities or Poor Patch Management	Application SW defects, confliction patching procedures between provider and customer, untested patch applicability, browser vulnerability…	1. Priviledge_Abuse 2. MITM 3. DoS_DDoS 4. Sniffing 5. VM_Threat 6. Impersonate 7. Data_Leakage
41	Lack of Policy/Poor Procedures for Log Collection and Retention	No well-defined policies for taking operational and security logs.	1. Operational_Logs_Compromise 2. Security_Logs_Compromise
42	Inadequate/Misconfigure d Filtering Resources	No proper resource filtering in place.	1. DoS_DDoS

It is noticed here that, Cloud has additional sets of vulnerabilities which results in risks like customer Lock-in, loss of governance, resource exhaustion, isolation failure, risks from changes of jurisdiction, etc. This implies there is requirement of more security parameters in case of cloud, which is different from generic security requirement identification used for web applications.

Step4. Identify the Security Requirements.

In view of above new Security requirement Multi-Trust has been added [16]. This requirement is defined to take into account any breach in SLA between Cloud Provider and Customer.

TABLE 4.3 IDENTIFY SECURITY REQUIREMENTS

Actors	*Functional Requirements*	*Non Functional Requirements*	*Possible Attacks/Security Risks*	*Security Requirements*
Cloud Customer	*SaaS* 1. Registration & Login 2. Update Login Details 3. Store data into Cloud 4. Manage automatic backup 5. Manage Sharing with cloud user 6. Download data stored in the cloud 7. Make payment for services used 8. Maintenance and Management of identity management system 9. Management of authentiation platform(including enforcing password policy) 10. Data and Traffic monitoring for security risk avoidance *PaaS* 1. Maintenance of identity management system 2. Identity management system 3. Authentication platform management (including enforcing password policy) *IaaS* 1. Maintenance of identity management system 2. Identity management system 3. Authentication platform management (including enforcing password policy)	1. Reliability 2. Less Response Time 3. Scalable 4. Correctness 5. Consistency 6. Recovery 7. Lawfulness of content 8. Compliance with data protection law 9. Personnel Security 10. Supply Chain Assurance 11. Use of standard Technologies and Solutions to avoid vendor Lock-in	1. Password_Cracking 2. Impersonate 3. Sniffing 4. Tampering 5. Social_Engg 6. Disclose_Data 7. Malicious_Code 8. Repudiate 9. Data_Theft 10. Human_Error 11. Password_Reuse 12. Multilocation_Dataplacement 13. Insider 14. MITM 15. Replay attacks 19. DoS_DDoS 20. VM_Threat 21. Sabotage 22. Tampering 23. Lock-In	1. Authentication 2. Authorization 3. Auditing 4. Immunity 5. Integrity 6. Intrusion Detection 7. Nonrepudiation 8. Privacy 9. Physical Protection 10. System Maintenance 11. Multi-Trust

	4. Guest OS patch and hardening procedures management 5. Guest security platform Configuration management 6. Guest systems monitoring 7. Security platform management and configuration (Firewall rules, IDS/IPS tuning, etc) 8. Log Collection & security monitoring		24. Governace_Loss 25. Operational_Logs_Compromise 26. Security_Logs_Compromise 27. Data_Deletion 28. Supply_Chain_Failure 29. Compliance_Challenges 30. Legal_Issues 31. Priviledge_Abuse 32. Backup_Lost_Stolen 33. Unathorized_Physical_Access 34. Natural_Disaster	
Cloud Users	1. Registration & Login 2.View shared data based on permission 3. Submit request to join the group 4.Unjoin a group	1. Reliability 2. Less Response Time 3. Scalable 4. Correctness 5. Consistency 6. Recovery	1. Password_Cracking 2. Impersonate 3. Sniffing 4. Change_Data 5. Social_Engg 6. Disclose_Data 7. Data_Theft 8. DoS_DDOS 9. Tampering	1. Authentication 2. Authorization 3. Immunity 4. Integrity 5. Nonrepudiation 6. Privacy
Cloud Service Provider	*SaaS* 1. Manage Cloud customer's Account 2. Manage Customer Data 3. Manage cloud hardware and software 4. Receive cloud usage payment 5. Maintain SLA 6. Data Processing	1. Reliability 2. Integrity 3. Recovery 4. Performance 5. Data and Traffic monitoring for security risk avoidance 6. Personnel Security 7. Supply Chain Assurance	1. Password_Cracking 2. Impersonate 3. Sniffing 4. Tampering 5. Social_Engg 6. Disclose_Data 7. Malicious_Code	1. Authentication 2. Authorization 3. Auditing 4. Immunity 5. Integrity 6. Intrusion Detection 7. Nonrepudiation

	7. Physical support infrastructure, security and availability	8. Scalability	8. Repudiate	8. Privacy
	8. OS patch management and hardening procedures	9. Response Time	9. Data_Theft	9. Physical Protection
	9. Security platform management and configuration (Firewall rules, IDS/IPS tuning, etc)	10. Restricted access to concerned cloud customer enterprise	10. Human_Error	10. System Maintenance
	10. Systems monitoring		11. Password_Reuse	11. Multi-Trust
	11. Log Collection & security monitoring		12. Multilocation_Dataplacement	
	12. Define Backup Strategy		13. Insider	
			14. MITM	
	PaaS		15. Replay attacks	
	1. Physical support infrastructure, security and availability		19. DoS_DDoS	
	2. OS patch management and hardening procedures		20. VM_Threat	
	3. Security platform management and configuration (Firewall rules, IDS/IPS tuning, etc)		21. Sabotage	
	4. Systems monitoring		22. Tampering	
	5. Log Collection & security monitoring		23. Lock-In	
	6.Multi-Tenanted Application separation		24. Governace_Loss	
	7. Sandboxing of application		25. Resource_Exhaustion	
			26. Operational_Logs_Compromise	
	IaaS		27. Security_Logs_Compromise	
	1. Physical support infrastructure, security and avalibility		28. Data_Deletion	
	2. Host Systems - Hypervisor, virtual firewall)		29. Supply_Chain_Failure	
			30. Compliance_Challenges	
			31. Legal_Issues	
			32. Priviledge_Abuse	
			33. Backup_Lost_Stolen	
			34. Unathuorized_Physical_Access	
			35. Natural_Disaster	
			36. Network_Issues	

			37. Customer_Provider_Hardening_Process	

Step 5: Identify Assets associated with Security requirements

TABLE 4.4 IDENTIFY ASSOCIATED ASSETS WITH REQUIREMENTS

Actors	*Functional Requirements*	*Non Functional Requirements*	*Vulnerabilities*	*Possible Attacks/Security Risks*	*Assets*
Cloud Customer	*SaaS* 1. Registration & Login 2. Update Login Details 3. Store data into Cloud 4. Manage automatic backup 5. Manage Sharing with cloud user 6. Download data stored in the cloud 7. Make payment for services used 8. Maintenance and Management of identity management system 9. Authentication platform management (including enforcing password policy) 10. Data and Traffic monitoring for security risk avoidance *PaaS* 1. Maintenance of identity management system 2. Identity	1. Reliability 2. Less Response Time 3. Scalable 4. Correctness 5. Consistency 6. Recovery 7. Lawfulness of content 8. Compliance with data protection law 9. Personnel Security 10. Supply Chain Assurance 11. Use of standard Technologies and Solutions to avoid vendor Lock-n	AAA Lack or poor encryption of archives and data in transit Remote Access to Management Interface Communication encryption vulnerabilities Multiple Jurisdiction data storage and Lack of Transparency about this Lack of Jurisdictions information Possibility of co-resident checks Possibility of Internal (Cloud) Network Probing Lack of standard technologies and solutions Poor Provider Selection Lack of supplier redundancy Lack of completeness or transparency in terms of use No Control On Vulnerability Assessment Process Application Vulnerabilities or Poor Patch Management Audit or Certification not	1. Password_Cracking 2. Impersonate 3. Sniffing 4. Tampering 5. Social_Engg 6. Disclose_Data 7. Malicious_Code 8. Repudiate 9. Data_Theft 10. Human_Error 11. Password_Reuse 12. Multilocation_Dataplacement 13. Insider 14. MITM 15. Replay	1. Company Reputation 2. Personal Sensitive Data 3. Personal Data 4. Personal Data - Critical 5. Service delivery- real time services 6. Service delivery 7. Customer Trust 8. Employee Loyalty and experience 9. Certifications 11. Intellectual Property 12. HR Data 13. Cloud service Management Interface 14. Backup or Archive Data 15. Credentials 16. User Directory (data)

	management system 3. Authentication platform management (including enforcing password policy) *IaaS* 1. Maintenance of identity management system 2. Identity management system 3. Authentication platform management (including enforcing password policy) 4. Guest OS patch and hardening procedures management 5. Guest security platform Configuration management 6. Guest systems monitoring 7. Security platform management and configuration (Firewall rules, IDS/IPS tuning, etc.) 8. Log Collection & security monitoring		available to customers Cross-cloud applications creating hidden dependency No proper Policies for Resource Capping Unclear Roles and Responsibilities Poor Enforcement of Role Definitions Need-To-Know Principle Not Applied System OS Vulnerabilities Application Vulnerabilities or Poor Patch Management Misconfiguration Sensitive Media Sanitization Inadequate/Misconfigured Filtering Resources User Provisioning Vulnerabilities De-Provisioning Vulnerabilities Poor Key Management procedures Key Generation: Low entropy of Random Number Generation Lack of Security Awareness	attacks 19. DoS_DDoS 20. VM_Threat 21. Sabotage 22. Tampering 23. Lock-In 24. Governace_Loss 25. Operational_Logs_Compromise 26. Security_Logs_Compromise 27. Data_Deletion 28. Supply_Chain_Failure 29. Compliance_Challenges 30. Legal_Issues 31. Priviledge_Abuse 32. Backup_Lost_Stolen 33. Unathorized_Physical_Access 34. Natural_Disaster	17. Network(connections, etc) 18. Access control/authentication/authorization(root/admin vs others) 19. Operational Logs 20. Security Logs
Cloud Users	1. Registration & Login 2.View shared data	1. Reliability 2. Less	AAA Lack or poor encryption of	1. Password_Cracking 2. Impersonate	Personal Sensitive Data Personal Data

	based on permission	Response Time	archives and data in transit		
	3. Submit request to join the group	3. Scalable	Communication encryption vulnerabilities	3. Sniffing	Personal Data - Critical
	4.Unjoin a group	4. Correctness	Poor Key Management procedures	4. Change_Data	Customer Trust
		5. Consistency	Lack of Security Awareness	5. Social_Engg	Intellectual Property
		6. Recovery	Hypervisor Vulnerabilities	6. Disclose_Data	Backup or Archive Data
			Lack of Jurisdictions information	7. Data_Theft	Credentials
			SLA usage, transparency and completeness	8. DoS_DDOS	Network(connections, etc)
			Lack of Reputational Isolation	9. Tampering	
Cloud Service Provider	*SaaS* 1. Manage Cloud customer's Account 2. Manage Customer Data 3. Manage cloud hardware and software 4. Receive cloud usage payment 5. Maintain SLA 6. Data Processing 7. Physical support infrastructure, security and availability 8. OS patch management and hardening procedures 9. Security platform management and configuration (Firewall rules, IDS/IPS tuning, etc.) 10. Systems monitoring 11. Log Collection & security monitoring 12. Define Backup Strategy	1. Reliability 2. Integrity 3. Recovery 4. Performance 5. Data and Traffic monitoring for security risk avoidance 6. Personnel Security 7. Supply Chain Assurance 8. Scalability 9. Response Time 10. Restricted access to concerned cloud customer enterprise	AAA Lack or poor encryption of archives and data in transit Remote Access to Management Interface Communication encryption vulnerabilities Multiple Jurisdiction data storage and Lack of Transparency about this Lack of Jurisdictions information Possibility of co-resident checks Possibility of Internal (Cloud) Network Probing Lack of standard technologies and solutions Responsibilities Synchronization or Contractual Obligations External to Cloud Lack of supplier redundancy SLA usage, transparency and completeness No Control On Vulnerability Assessment Process Application Vulnerabilities or Poor Patch Management	1. Password_Cracking 2. Impersonate 3. Sniffing 4. Tampering 5. Social_Engg 6. Disclose_Data 7. Malicious_Code 8. Repudiate 9. Data_Theft 10. Human_Error 11. Password_Reuse 12. Multilocation_Dataplacement 13. Insider 14. MITM	Company Reputation Personal Sensitive Data Personal Data Personal Data - Critical Service delivery- real time services Service delivery Customer Trust Employee Loyalty and experience Certifications Intellectual Property HR Data Cloud service Management Interface Backup or Archive Data Credentials

	PaaS 1. Physical support infrastructure, security and availability 2. OS patch management and hardening procedures 3. Security platform management and configuration (Firewall rules, IDS/IPS tuning, etc.) 4. Systems monitoring 5. Log Collection & security monitoring 6.Multi-Tenanted Application separation 7. Sandboxing of application *IaaS* 1. Physical support infrastructure, security and availability 2. Host Systems - Hypervisor, virtual firewall)		Audit or Certification not available to customers Cross-cloud applications creating hidden dependency Inaccurate modelling of resource usage Inadequate Resource Provisioning and Investments in Infrastructures No proper Policies for Resource Capping Unclear Roles and Responsibilities Poor Enforcement of Role Definitions Need-To-Know Principle Not Applied System OS Vulnerabilities Inadequate Physical Security Procedures Encrypted Data processing issues Application Vulnerabilities or Poor Patch Management Misconfiguration Sensitive Media Sanitization Inadequate/Misconfigured Filtering Resources User Provisioning Vulnerabilities De-Provisioning Vulnerabilities No proper Policies for Resource Capping Poor Key Management procedures Key Generation: Low entropy of Random Number	15. Replay attacks 19. DoS_DDoS 20. VM_Threat 21. Sabotage 22. Tampering 23. Lock-In 24. Governace_Loss 25. Resource_Exhaustion 26. Operational_Logs_Compromise 27. Security_Logs_Compromise 28. Data_Deletion 29. Supply_Chain_Failure 30. Compliance_Challenges 31. Legal_Issues 32. Priviledge_Abuse 33. Backup_Lost_Stolen 34. Unathorized_Physical_Access 35. Natural_Disaster 36. Network_Issues 37. Customer_Prov	User Directory (data) Network(connections, etc) Access control/authentication/authoriztion(root/admin vs others) Operational Logs Security Logs

			Generation	ider_Hardening _Process	

Step 6: Identify Risks associated with each Asset using CRAMM methodology

Asset prioritization is done using SMEs and various other surveys. Below table mentions Asset Prioritization using ENSIA Risk Assessment [15]:

TABLE 4.5 IDENTIFY RISKS ASSOCIATED ASSETS

Assets	Owner [actors/ involved organizations]	Observed Value [Scale: VERY LOW, LOW, MEDIUM , HIGH, VERY HIGH]	Score
1. Company Reputation	Cloud customer	Very High	10
2. Personal Sensitive Data	CP / Cloud customer	Very High	10
3. Personal Data	CP / Cloud customer	MEDIUM (operational value) / HIGH (value if lost)	7
4. Personal Data - Critical	CP / Cloud customer	HIGH (operational value) / HIGH (value if lost)	8
5. Service delivery- real time services	CP / Cloud customer	Very High	10
6. Service delivery	CP / Cloud customer	Medium	5
7. Customer Trust	Cloud customer	Very High	10
8. Employee Loyalty and experience	Cloud customer	High	8
9. Certifications	CP / Cloud customer	High	8
11. Intellectual Property	Cloud customer	High	8
12. HR Data	Cloud customer	High	8
13. Cloud service Management Interface	CP / Cloud customer	Very High	10
14. Backup or Archive Data	CP / Cloud customer	Medium	6
15. Credentials	Cloud customer	Very High	10
16. User Directory (data)	Cloud customer	High	8
17. Network(connections, etc)	CP / Cloud customer	High	8
18. Access control/authentication/authorization(root/admin vs others)	CP / Cloud customer	High	8
19. Operational Logs	CP / Cloud customer	Medium	6
20. Security Logs	CP / Cloud customer	Medium	6

21. Physical Hardware	CP / Cloud customer	LOW (depends on how much you lose) / MEDIUM (could be serious if stolen and not protected)	5

Various Risk Rating methodologies are studied so that Impact, Vulnerabilities and threat can be evaluated [14] [15]. Below table shows Risk Assessment for any asset in Cloud System as example. This can be later extended to all the assets identified to evaluate high asset risks and corresponding high rated security functionality.

TABLE 4.6 IDENTIFY RISKS ASSOCIATED WITH THREATS

Asset: 1.x Asset Owner: y																		
Security Requirement/Objectives	Authentication & Authorization Probability							Auditing		Immunity						Integrity		
Impact Requirement (1-10)	10 / secure							6/Auditing Supported		10/secure						10/secure		
Threats (list all that apply)	Password_Cracking	Impersonate	Social_Engg	Human_Error	Password_Reuse	Privileged_Abuse	Insider	Operational_Logs_Compromise	Security_Logs_Compromise	Malicious_Code	VM_Threat	Sabotage	MITM	DoS_DoS	Replay attacks	Tampering	Disclose_Data	Data_Deletion
Vulnerability [1-10] none [0], low [1-4], moderate [5-7], high [8-9], very high [10]	8.5	9	7	7	7.7	6.3	8.5	5.8	5.8	7.3	8.3	4.9	6	6.1	6.8	5.5	8.5	5
Threat (1 to 100) Impact X Vulnerability	85	90	70	70	77	63	85	34.8	34.8	73	83	49	60	61	68	55	85	50
Risk Level Low (1-35), Medium (36-70), High (71-100)	High	High	Medium	Medium	High	Medium	High	Low	Low	High	High	Medium	Medium	Medium	Medium	Medium	High	Medium

Asset: 1. x
Asset Owner: y

Security Requirement/Objectives	Intrusion Detection		Non repudiation	Privacy		Physical Protection				System Maintenance		Multi-Trust				
Impact Requirement (1-10)	9/secure		10/secure	10/secure		7/secure				7/secure		10/secure				
Threats (list all that apply)	Sniffing	Insider	Repudiate	Data_Theft	Tampering	Multilocation_Dataplacement	Backup_Lost_Stolen	Unathorized_Physical_Access	Natural_Disaster	Network_Issues	Customer_Provider_Hardening_Process	Lock-In	Governace_Loss	Supply_Chain_Failure	Legal_Issues	Compliance_Challenges
Vulnerability [1-10], none [0], low [1-4], moderate [5-7], high [8-9], very high [10]	5.3	8.5	8.5	8.3	8.3	7.3	6.5	6.5	5	7	7	7.8	7.8	6.7	10	8.2
Threat (1 to 100) Impact X Vulnerability	47.7	76.5	85	83	83	51.1	45.5	45.5	35	49	49	78	78	67	100	82
Risk Level Low (1-35), Medium (36-70), High (71-100)	Medium	High	High	High	High	Medium	Medium	Medium	low	Medium	Medium	High	High	High	High	High

CHAPTER 5

5 Design Methodology for Secure Cloud Systems

In this chapter Design Methodology is proposed based on Security Requirements elicited in previous chapter using below mentioned steps.

5.1 Mapping of security requirements with security services

Five key security criteria are identified for cloud platforms, and it is shown in previous Chapter in Step 4, that most of the typical attacks map under one of these six categories:

TABLE 5.1 MAPPING OF SECURITY REQUIREMENTS WITH SECURITY SERVICES

Security services	Security requirement	Security Service Example
Confidentiality	Privacy requirements	Disclosure of illicit information
	Immunity requirements	
Integrity	Integrity requirement	Unauthorized information alteration
Availability	Authentication requirements	Access disruption to systems
	Authorization requirement	
	Intrusion Detection	
	Non-repudiation requirement	
	Physical Protection	
	System Maintenance	
Audit ability	Auditing	Lack or compromised auditable record of usage.
Multi-Trust	Multi-Trust	SLA breach of one tenant due to another's actions

5.2 Security design analysis

In this step we map attacks to various security mechanisms in place. Brief description of these security mechanisms are mentioned below. These mitigation

mechanisms can be used alone or in combination to mitigate the risks/attacks on a Cloud System.

5.2.1 Ensure Data Portability

Designing Cloud Services and its applications keeping in mind data portability helps to avoid cloud customer and cloud user Lock-In threat. Lock-In has become one of the primary threats, which is affecting current Cloud Vendors reputation and customer trust. Taking care of this threat reduces migration issues and associated cost of cloud customers. It has become one of the major selection criteria for cloud customer to differentiate among various cloud vendors. This can be taken care by using Standard APIs and Protocols. This is taken care by Free Libre Open Source Software (FLOSS) licensing. The Open Group following technologies for standardization:

Cloud Interface Libraries	Hypervisors	Cloud Platforms
Apache Libcloud	Virtual Box	Open Eucalyptus
Redhat libvirt	XEN	Open Nebula
DeltaCloud API	KVM	OpenStack
Jclouds	QEMU	openQRM

Cloud Interfaces

These software interfaces allow developers to integrate in their solution different platforms. Even though implemented using different programming language they provide same access to underlying platform.

Hypervisors

Hypervisors are used for implementing hardware virtualization. This allows host computer in running many other virtualized operating system.

Open Virtualization Format (OVF) – Distributed Management Task Force (DMTF) submitted proposal for this format in September 2007. These standards were designed to be independent of any hypervisor or processor

architecture and provide "open, secure, portable, efficient and extensible format for the packaging and distribution of software to be run in virtual machines" [16].

Cloud Platforms

Scalability, elasticity and tolerance to failure is provided by cloud platform, which integrates cloud interfaces, hypervisors and management technologies [17].

5.2.2 Ensure compliance to other necessary standards

5.2.2.1 PCI-DSS

The Payment Card Industry – Data Security Standard (PCI-DSS) is organization's proprietary information security standards who are dealing with credit cards like Visa, American Express, MasterCard, etc. To reduce credit card frauds via sensitive information exposure led to creation of this standard [18].

5.2.2.2 SSAE16

Various auditing standard are followed by service organizations. Statement on Standards for Attestation Engagement (SSAE) 16 is one of them. It supersedes SAS70 (Statement of Auditing Standard No.70). It provides two reports – SOC1 (Service Organization Control Report 1), SOC2. Effective internal controls including financial reporting (Sarbanes Oaxley) requirement can be complied by using SSAE16.

5.2.2.3 ISAE3402

International Standards for Assurance Engagement is assurance standard and see applicability in Information security as well. It is similar to SSAE16 and generates SOC reports. Such reports give assurance to organizations customers and service users.

5.2.2.4 ISO 270001:2013

This information security standard was published by International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) on 25th September 2013. On completion of formal audit process, organizations meeting the standards may get official certification issued from independent and accredited certification body. This Standard adds new controls like Information security in project management, Restrictions on software installation, Secure development policy, system engineering principles, development environment, security testing, Information security policy for supplier relationships, Information and communication technology on supply chain, Assessment and decision on information security events, Response to information security incidents, Availability of information processing facilities.

5.2.3 Cryptographic techniques

Cryptography provides a systematic way for securing data from various threats and attacks. A sensitive data can be encrypted and via authentication and access control it can be protected against unauthorized modification.

Secret key or Symmetric cryptography uses same key to encrypt and to decrypt text. Sharing of secret keys becomes major vulnerability of Symmetric key cryptography. This is overcome by public key cryptography where two keys - private and public- mathematically related are used. The security compromise is avoided as sharing of private key is not there.

Hash functions are collision-resistant one-way function. Any size of message, computes a smaller is converted fixed-size message called a digest or hash.

TABLE 5.2.3.1 SYMMETRIC CIPHERS

Name of algorithm	Block size (bits)	Key size (bits)	Encryption speed (on 33 MHZ 486SX) (Kb/s)
DES	64	56	35
Blowfish	64	128	182
3DES (Triple DES)	64	168	12
IDEA	64	128	70
AES	128	128	60
CAST	64	128	53
RC5	64	128	86
RC4 (Stream cipher)	One byte at a time	256	164
SEAL (Stream cipher)	One byte at a time	160	381
PIKE (Stream cipher)	One byte at a time	160	62

TABLE 5.2.3.2 ASYMMETRIC CIPHERS

Name of algorithm	Encryption (ms)	Decryption (ms)	Sign (ms)	Verify (ms)
RSA (512 bits)	30	160	160	20
RSA (768 bits)	50	480	520	70
RSA (1024 bits)	80	930	970	80
ECDSA (160 bits)	797	281	150	230
ECDSA (233 bits)	882	385	250	521
ECDSA (283 bits)	928	400	25	580
HECDSA (81 bits)	668	191	60	31
HECDSA (83 bits)	893	224	56	32
ElGamal (512 bits)	330	240	250	1370

TABLE 5.2.3.3 HASH FUNCTIONS

Algorithm	Hash length (bits)	Encryption speed (on 33 MHZ 486SX) (Kb/s)
MD4	128	23
MD5	128	236
HAVAL	128	174
N-HASH	128	29
SHA1	160	75
SHA2	160	70

5.2.4 Two_Factor_Authentication

Two Factor authentications are based on assumption that an unauthorized actor may not be able to provide both factors required for access. This includes:

a. Any physical object in possession with the user – a key, smart card, USB Stick with secret token, a bank card, etc...
b. Any secret known only to the user – password, username, pin etc.
c. Any physical characteristics (biometrics) – fingerprint, retina, voice, etc.

5.2.5 Multi-factor_Authentication

Multi-factor is combination of more than one of the factors used in authentication:

a. Knowledge Factor – User is required to provide a secret in order to authenticate, for eg. Password, PIN, secret questions.
b. Possession Factor – Something's which only user can possess, like disconnected token (these token have no connection with client computer, user is shown authenticated data), connected tokens (tokens/devices which can physically connect to the computer, eg. key), inherence factors (biometric method) [19].

Identity Theft and other online fraud are prevented by Multi-factor authentication, but it is still vulnerable to phishing and man-in-the-middle attack. The obvious disadvantages of Multi-factor authentication is use of Hardware token which has become additional burden for user to maintain. Also, there are interoperability issues as each vendor develops its own mechanism for multi-factor authentication.

5.2.6 Kerberos

Kerberos was developed by MIT to protect Project Athena network services. It is a computer network authentication protocol and uses symmetric key cryptography, optionally public-key cryptography and requires a trusted third party. It works on the basis of tickets, allowing nodes communicating over a network which is unsecure to prove identity to each other in secure manner.

After client authenticates itself to Authentication Server (AS), then the AS forwards the username to key distribution center (KDC). Then KDC issues ticket-granting ticket (TGT). This is then time-stamped. The TGT is encrypted using user's password. This encrypted result is returned to the user's workstation. This is done very few times, and on expiry of TGT at some point of time, it is renewed transparently by user's session manager.

When client communicated to another node, it sends TGT to ticket-granting services (TGS), which shares the same host as KDC. After TGT verification, the user is permitted for requested service access [20].

Kerberos provides protection against eavesdropping and replay attacks.

5.2.7 OCSP Stapling

Online Certificate Status Protocol (OCSP) provides checking for revocation status of X.509 digital certificates. OCSP messages are encoded in ASN.1 [21] and communicated over HTTP usually. OCSP servers are termed as OCSP responders, because of request/response nature of OCSP messages. OCSP responder sends a signed response indicating the certificate specified in request is good, revoked or unknown or may return an error code. OCSP can be vulnerable to replay attacks if good response gets captured by malicious intermediary and replayed at a later date to the client. This is overcome by OCSP by including "nonce" (a random or pseudo number used once). As it is required to contact a third party for confirming certificate validity, security concerns have risen. Also OCSP introduces significant cost to certificate

authorities (CA) as it requires them to respond every client in real time. This slows down browsing. This is taken care by OCSP Stapling – the certificate holder queries at regular interval the OCSP server by themselves and obtain a signed time-stamped OCSP response.

When any user at the client site attempts to connect to the site, the OCSP time stamped response is stapled along with the TLS/SSL Handshake via Certificate Status Request extension response. It may appear that allowing the site operator to control verification responses, may make any fraudulent site issue false verification for revoked certificate, the stapled responses cannot be forged, as they are issued directly from certificate authority. As client continues to have assurance that can be verified from certificate authority on certificate validity, it no longer needs to contact OCSP server again and again [21].

5.2.8 Vulnerability_Assessment_Tools

As number of attacks is increasing, many tools are becoming available to a person/organization to detect and stop malware and cracking attempts. Some common vulnerability assessment tool that can be used by system admin are Wireshark (www.wireshark.org), Nmap (nmap.org), Metaspoilt (metasploit.com), OpenVAS (www.openvas.com), Aircrack (www.aircrack-ng.org).

5.2.9 Need-to-know_Principle_Enforcement

The organization must enforce this principle to avoid any unnecessary risk due to unnecessary access given to person/parties. This helps to avoid data theft/tampering by malicious insider by abuse of high privilege roles.

5.2.10 RnR_Clarity

Inadequate attribution of roles and responsibilities can lead to Loss of Governance risk for a Cloud Provider. This means that not only organizations' strategy is impacted but it would mean complying with security requirements difficult, lack of confidentiality, integrity and availability of data, deterioration of performance and quality of service.

5.2.11 SLA_Strengthening

Service Level Agreement (SLA) is very important legal document and must be carefully designed by cloud provider and must be carefully studied and understood by the cloud customer. SLA can contain clauses which can be detrimental in the area of Intellectual Property, like CP having rights to any content getting stored in its provided cloud in the cloud infrastructure. SLA can impact supply chain assurances if SLA provisions guaranteed by 3rd party suppliers are lower that SLAs a CP offers to CS. A CS must evaluate any SLA on terms of Data Security and Protection, Data Transfer, Law enforcement access, Intellectual Property, confidentiality and non-disclosure, Risk Allocation and limitation of liability and Change of control.

SLAs must be able to quantify various possible risk scenarios and the possible impacts of security breaches on reputation. Example of this is mentioned below [15]:

TABLE 5.2.11.1 SLA EXAMPLE

Goal	KPI	Value	Penalties
Service availability	% Uptime per month	99.99	20% reduction in bill for every factor of 10
Latency (NB this is the time from when the stock market publishes data)	Average response time over 100 requests over 1 day	1 sec	5% reduction in bill for every violation
Administration	Time to respond to request in minutes	60 min	5% reduction in bill for every violation
Alerting	Minutes to alert customer of a violation of service (not including this one...)	5 min	5% reduction in bill for every violation
Time to recover from fault	Hours	2 hours	5% reduction in bill for every violation

All of the above cloud security risk mitigation techniques have been mapped against attacks which have most severe impact (Chapter 4, Step 6).

TABLE 5.2.11.2 MAPPING OF SECURITY MITIGATION TECHNIQUES WITH SECURITY SERVICES

Security services	Security requirement	Attacks	Security Mitigation Techniques
Confidentiality	Privacy requirements	Malicious_Code	Crpytographic Techniques, Two_Factor_Authentication, Multi_Factor_Authentication Kerberos, Vulnerability_Assessment_Tools
	Immunity requirements	VM_Threat	OVF, VM_Traffic_Monitoring, Cryptographic Techniques, Two_Factor_Authentication, Multi_Factor_Authentication, Vulnerability_Assessment_Tools, Kerberos,Resource_Allocation_Separation_Mechanisms, Hooksafe
		Sabotage	Crpytographic Techniques, Two_Factor_Authentication, Multi_Factor_Authentication ,Kerberos, Vulnerability_Assessment_Tools
		MITM	Crpytographic Techniques, Two_Factor_Authentication, Multi_Factor_Authentication, Kerberos, Vulnerability_Assessment_Tools

		DoS_DDoS	DoS_DDoS_Mitigation
		Replay attacks	Cryptographic Techniques, Kerberos
Integrity	Integrity requirement	Tampering	Crpytographic Techniques, Two_Factor_Authentication, Multi_Factor_Authentication, Kerberos, Vulnerability_Assessment_Tools
		Data_Deletion	Cryptographic Techniques
Availability	Authentication requirements	Password_Cracking	Crpytographic Techniques, Two_Factor_Authentication, Multi_Factor_Authentication ,OCSP, OCSP Stapling, Multiple Certificate Status Request Extension, Vulnerability_Assessment_Tools
	Authorization requirement	Impersonate	Crpytographic Techniques, Two_Factor_Authentication, Multi_Factor_Authentication, Vulnerability_Assessment_Tools, Kerberos
	Intrusion Detection	Social_Engg	Crpytographic Techniques, Two_Factor_Authentication, Multi_Factor_Authentication, RnR_Clarity
	Physical Protection	Password_Reuse	Crpytographic Techniques, Two_Factor_Authentication, Multi_Factor_Authentication, Kerberos
	System Maintenance	Priviledge_Abuse	Need-to-know_Principle_Enforcement, RnR_Clarity
		Insider	Crpytographic Techniques, Two_Factor_Authentication, Multi_Factor_Authentication, RnR_Clarity
		Sniffing	Kerberos, Vulnerability_Assessment_Tools
Audit ability	Auditing	Operational_Logs_Compromise	SLA_Stregthening
		Security_Logs_Compromise	SLA_Stregthening
Multi-Trust	Multi-Trust	Lock-In	Data_Portability,Standard APIs, OVF, CMPI, SMI-S, PCI-DSS, SAS70, SSAE 16, SLA_Strengthing
		Governace_Loss	SLA_Stregthening
		Legal_Issues	SLA_Stregthening

There are many cryptographic techniques. Comprehensive evaluation of each is required with respect to the attack they mitigate. It can be seen from below table, that not all attacks mentioned above are mitigated by a single cryptographic technique alone. It needs to be used with other mitigation techniques.

TABLE 5.2.11.3 IMPACT ANALYSIS OF CRYPTOGRAPHIC MITIGATION TECHNIQUES

Security services	Security requirement	Attacks	Suitable Cryptography Algorithm										
			Asymmetric Algorithm			Symmetric Alogorithm			Hashing Algorithm		Signature Algorithm		
			RSA	ECC	HECC	AES	DES	Triple DES	MD5	SHA1	RSA+DSA	ECDSA	HECDSA
Confidentiality	Privacy requirements	Malicious_Code	N	N	N	Y	Y	Y	N	N	Y	Y	Y
	Immunity requirements	MITM	Y	N	N	Y	Y	Y	N	N	N	N	N
		DoS_DDoS	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y
		Replay attacks	Y	N	N	Y	N	N	N	N	N	N	N
Integrity	Integrity requirement	Tampering	N	N	N	Y	Y	Y	N	N	Y	Y	Y
		Data_Deletion	N	N	N	Y	Y	Y	N	N	Y	Y	Y
Availibility	Authentication requirements	Password_Cracking	Y	N	N	Y	Y	Y	N	N	N	N	N
	Authorization requirement	Impersonate	Y	N	N	Y	Y	Y	N	N	N	Y	Y
	Intrusion Detection	Password_Reuse	Y	Y	Y	N	N	N	Y	Y	Y	Y	Y
	Physical Protection	Sniffing	Y	N	N	Y	N	N	N	N	N	N	N
Total Impact			6	1	1	9	7	7	2	2	5	6	6

Impact analysis provides applicability of the cryptographic algorithm for a particular attack. “Y” shows that the algorithm is impacted by the particular attack and vice-a-versa. Lowest value of Total impact shows better resilience of the algorithm. As per the table above, it is found that ECC, HECC provides best protection against most of the attacks. This when combined with design constraints provide better solution to a developer.

5.3 Identifying Security Design Constraints

In this step we indentify Security Design constraints as mentioned in Table 5.8

TABLE 5.2.11.4 SECURITY DESIGN CONSTRAINTS

Quality Attributes	Attribute Details	IaaS		PaaS		SaaS			
						Cloud Customer		Client	
		Complexity	Priority	Complexity	Priority	Complexity	Priority	Complexity	Priority
Performance	Runtime Performance	High	High	High	High	High	High	High	High
	Low Memory Footprint	Medium	Medium	Medium	Medium	Medium	Medium	High	High
	Power Consumption	Medium	Low	Medium	Low	Medium	Medium	High	High
	Network Availability	High	High	High	High	High	High	High	Medium
Security	CIAAM Security Objectives	High	High	High	High	High	High	High	High
Scalability	Scalable without affecting current functioning	High	High	High	High	High	High	High	Medium
Portability	OS Independence	Medium	High	High	High	High	Medium	Medium	High
Usability	Compatibility	High	High	High	High	High	Medium	High	Medium
	Programmability	High	Low	High	High	Medium	Medium	Medium	Low
Cost	Cost of Chosen Solution	Medium	High	Medium	High	Medium	High	Medium	High

5.4 Security Design Structuring

a. Identifying design attributes and prioritizing them

- Based on design constraints, tradeoff is required.
- For Web Application Hosting Cloud Storage, if application is required to be made for Smart Phones, PDA, Tablets, following Quality Attributes must be taken care:
    i. Performance – Runtime, Low Memory, Less Power Consumption
    ii. Security – CIA Triad
    iii. Usability – Compatibility and Programmability
- For Authentication and Privacy, most effective Algorithm resisting all kinds of attack is found to be
    - ECDSA
    - HECDSA

  Based on Quality Attributes requirements, HECDSA should be chosen over ECDSA for Authentication.

b. Preparation of security design template

Security design template is prepared to take care of each requirement. This will store each specification of the design constraints and design attributes of specific environment. All the mitigation techniques against Quality attributes are listed. Based on design constraints the required mitigation technique is selected.

TABLE 5.2.11.5 SECURITY DESIGN TEMPLATE

SDT for Cloud System										
Design Constraint	Design Attribute	IaaS		PaaS		SaaS				Security Mechanisms
						Cloud Customer		Client		
		Complexity	Priority	Complexity	Priority	Complexity	Priority	Complexity	Priority	Techniques
Performance	Runtime Performance	High	High	High	High	High	High	High	High	Cryptographic Techniques RSA ECC HECC AES DES Triple DES MD5 SHA1 RSA+DSA ECDSA HECDSA Data Portability Cloud Interface Libraries Apache Libcloud Redhat libvirt DeltaCloud API Jclouds Hypervisors OVF Virtual Box XEN KVM QEMU Cloud Platforms Open Eucalyptus Open Nebula OpenStack openQRM Other Certifications and Standards PCI-DSS SSAE 16 ISAE3402 ISO 270001:2013 Availibility Techniques Two_Factor_Authentication Multi-factor_Authentication
	Low Memory Footprint	Medium	Medium	Medium	Medium	Medium	Medium	High	High	
	Power Consumption	Medium	Low	Medium	Low	Medium	Medium	High	High	
	Network Availability	High	High	High	High	High	High	High	Medium	
Security	CIA Triad	High	High	High	High	High	High	High	High	
Scalability	Scalable without affecting current functioning	High	High	High	High	High	High	High	Medium	

Portability	OS Independence	Medium	High	High	High	High	Medium	Medium	High	Kerberos OCSP Stapling Others VM_Traffic_Monitoring Vulnerability_Assessment_Tools Resource_Allocation_Separation_Mechanisms Hooksafe DoS_DDoS_Mitigation RnR_Clarity Need-to-know_Principle_Enforcement SLA_Stregthening
Usability	Compatibility	High	High	High	High	High	Medium	High	Medium	
	Programmability	High	Low	High	High	Medium	Medium	Medium	Low	
Cost	Cost of Chosen Solution	Medium	High	Medium	High	Medium	High	Medium	High	

CHAPTER 6

6 Case Study

The proposed requirement elicitation and design methodology considers all possible threats and design constraints. It targets early consideration of possible attacks, right from requirement elicitation and design. This helps in bringing down the cost spent in adding corrective measures at later stages. Also any such addition can make system architecture constraint.

In order to validate the proposed Design methodology we will consider the popular Cloud System and service - **Amazon EC2**.

Amazon is one of the major industry players in Cloud Computing. It provides all major services – IaaS, PaaS and SaaS. Amazon had decade of experience in managing large-scale, efficient and reliable IT Infrastructure as world's largest online platforms. In 2006, Amazon Web Services (AWS) was launched opening its large-scale distributed and transactional IT infrastructure for all. It boasts being Flexible, Cost-effective, Scalable and elastic, Secure and Experienced and provides obvious advantages of cloud platform. Many organizations are using AWS today for HR solutions, Inventory Management solutions, Payroll applications, e-commerce website hosting, Storage as a Service, media database, large-scale simulations for a pharmaceutical company.

Amazon has SAS70, SSAE16, ISA3402 SOC report completion. It has ISO27001 certification and has been successfully validated level 1PCI-DSS payment card industry standard [24].

It can be seen in our proposed design methodology as well, similar and more Standards and Certification requirements has been specified.

Below mentioned diagram summarizes Amazon Web Services Cloud Platform [24]:

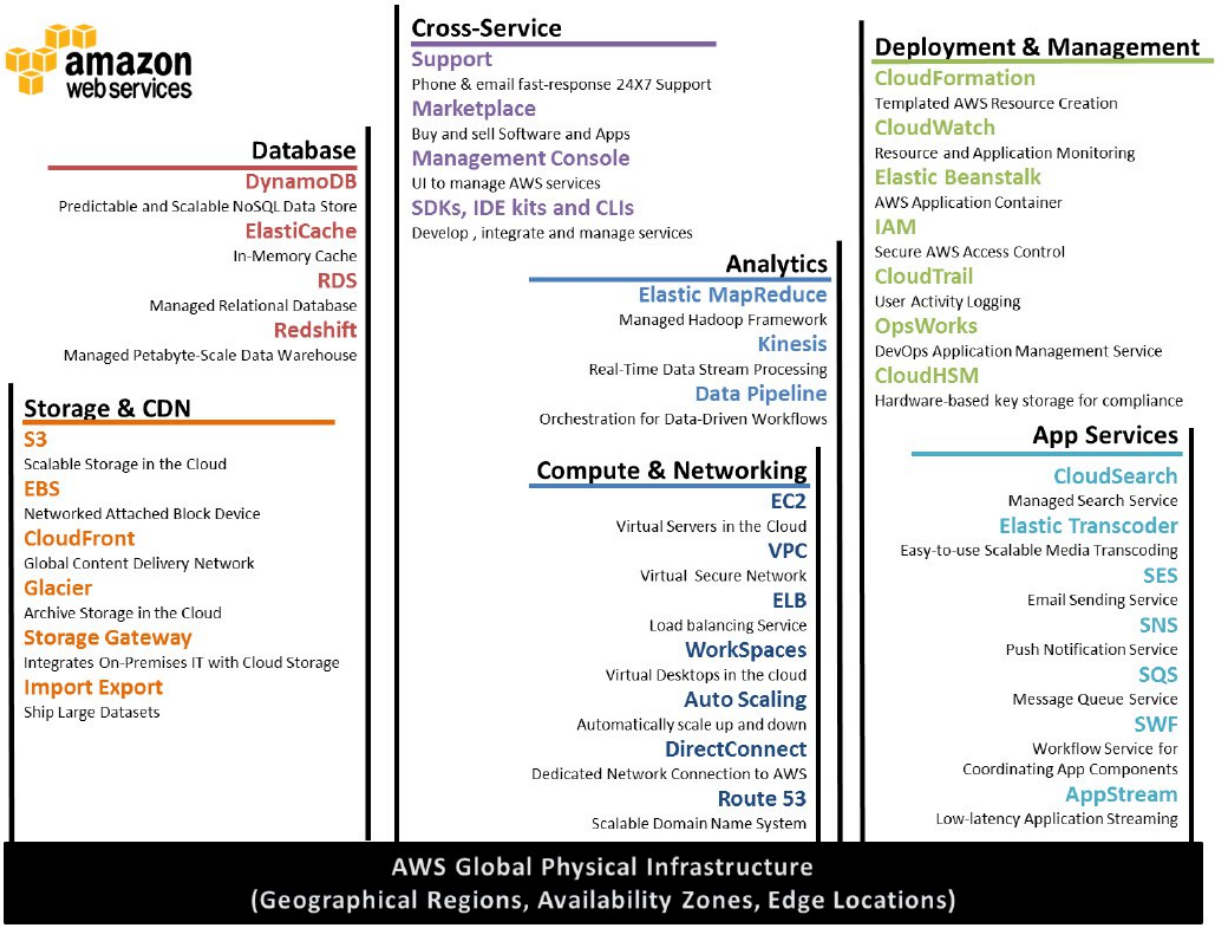

FIGURE 6-1 AMAZON WEB SERVICES CLOUD PLATFORM

Amazon IaaS has become very popular recently. Elastic Cloud Compute (EC2) uses Xen virtualization as hypervisor and is having one if the biggest deployed installations. It uses two different kinds of storage – Instance Storage (for non-persistent data) and Elastic Block Store (EBS) (network based persistent storage). User pays on per hour basis for create, launch and terminate server instances as needed using web service application provided by EC2.

Amazon Web Services (AWS) below mentioned security threat mitigation technique to counter various vulnerabilities and threats. It can be seen that almost direct mapping exists between the Security requirement identified in this thesis and what is provided by Amazon:

- **Network and Security Monitoring Systems** – DDoS and password brute-force detection on AWS accounts.
- **Secure Access** – Secure communication session with AWS server via SSL/TLS over HTTP/HTTPS, securing customer access point (API endpoints)
- **Built-in firewalls** – Instances accessibility can be configured from public to completely private.
- **Unique Users** – Identity and Access Management(IAM) level of access to cloud users to AWS services with unique security credentials (no sharing of password/keys required)
- **Multi-factor authentication** - Multi-factor authentication provided for root AWS account as well as IAM user account.
- **Encrypted data storage** – AES 256 is used for storing data in amazon servers.
- **Security Logs** – All user activity within AWS account are logged.
- **Physical Security** – All AWS data-center are physically secured, utilizes multi-factor access control to prevent unauthorized access [25].

Beside above, AWS also provides dedicated connection options using 802.1q VLANs, perfect forward secrecy, private subnets, Hardware Security Module (HSM) for cryptographic key storage, Trusted Advisor to alert security configuration gaps.

Two things must be noted here. Firstly these security mechanisms were developed over a period of time. This can be seen over the various attacks (DDoS attack on Amazon in 2008) chronology on AWS. Secondly, extensive cost and development time have been associated to have such mechanism in place [24].

This all, points to need of security vulnerabilities consideration at design phase itself. But, no system can be completely secure unless need for security is understood by each of the party involved implying – security is not only responsibility of Cloud service Provider but also Cloud Customer and Cloud User.

1. In one of the reports published by independent news agency [23] on Aug 2014, points out exploit on Amazon Hypervisor vulnerability because of mis-configuration at Cloud Customer end. A Web Application Security Scanning application security

expert, pointed out in 2014 Black Hat USA briefing, that a simple configuration error enables a determined attacker a route to control virtual instances and access sensitive and critical resources stored at Amazon Web Service (AWS). He was able to access web application repository, its public and private keys, was able to download the source code, identify different functions for configuring the source code.

2. In another such incidence, CloudForce.com had to suspend all its operations and close business within 12 hours of targeted DDoS attack either by insider or attacker who gained administrative privilege via social-engineering and deleted all data stored in Amazon EC2.

These are among worst attack possible on a Cloud system. If proper security vulnerability identification was done during security requirement and design this attack could have been mitigated. The proposed design methodology presented in this thesis, if followed by Cloud Customer and User would have mitigated this attack.

We have already identified VM_Threat as one of the high risk attack on Confidentiality Services in our Security Requirement for Cloud - prioritization of various possible attacks in Cloud System (Chapter 4, Step 6).. Its mitigation techniques include:

- OVF
- VM_Traffic_Monitoring
- Cryptographic Techniques
- Two_Factor_Authentication
- Multi_Factor_Authentication
- Vulnerability_Assessment_Tools
- Kerberos
- Resource_Allocation_Separation_Mechanisms
- Hooksafe

VM_Traffic_Monitoring and Vulnerability_Assesment_Tools would have identified and disabled the scanning of metadata server which caused access to web application repository and its public and private keys.

Beside above Need-to-know_Principle_Enforcement and RnR_clarity would have saved CloudForce.com from social engineering attack which made administrative user id/password accessible to the attacker.

Proper SLA understanding by CloudForce.com would have let known security is not only Cloud Provider responsibility, and a Cloud Customer as well needs to enforce standard security mechanisms to take care of all possible vulnerabilities in its system.

Our security requirement elicitation for this reason identifies and considers various actors including Cloud Service Provider, Cloud Customer and Cloud User. It then identifies various security requirement associated with them based on their respective functional and non-functional requirements.

The validity of all the security requirements and services – Confidentiality, Integrity. Availability, Auditability and Multi-trust to mitigate all possible security threats and vulnerabilities is established. With this use-case example, we also establish the validity of some of the various security mechanisms proposed in this thesis.

CONCLUSIONS

In this project design methodology for secure cloud system is presented. We proposed a Security Engineering Framework for the same as well. Based on the well-defined steps in this framework we have completed security requirement elicitation by identifying stakeholders, their functional and non-functional requirements, identifying vulnerabilities, assets and all possible attacks on them and then mapping all with security requirements. We also prioritized the attacks and threats by calculating their impact using CRAMM.

In security design engineering, we first identified 6 Security Objectives – Confidentiality, Integrity, Availability, Auditability and Multi-trust to map security requirements. We then mapped prioritized attacks and their respective mitigation techniques with security services. For cloud system we emphasized that not only cryptography but additional techniques using same needs to be in place. Comparison of cryptography algorithm against various attacks was done to identify best algorithm. We identified all design constraints which can impact on selection of security mitigation techniques. We then proposed a security design template that will ease designer load in choosing appropriate Security Mechanism for his system. Based on design constraints we identified effective algorithm

Using Amazon EC2 Cloud Service provider example, we verified that our design methodology and identified security requirements based on vulnerabilities present in cloud system is correct. Here we also found that security is joint responsibility of all the stakeholders in the system.

As security vulnerability landscape is changing at rapid space there is need to identify various security methodologies, their affective comparison and evaluate the impact on various threats keeping design constraints like system performance, cost in mind. We must keep on updating our SDT for effective decision making. The future work of this project includes former and validation of this design methodology on various Cloud Services.

REFERENCES

1. A framework for development of secure software, 2013, CSI Transactions on ICT, ISSN 2277-9078, CSIT, Kakali Chaterjee, Daya Gupta & Ashok Dey

2. OpenSSL_HeartBleed_Bug,2014,http://heartbleed.com, https://blog.cloudsecurityalliance.org/2014/06/16/openssl-ccs-injection-vulnerability-countdown

3. Draft_NIST_SystemSecurityEngineering_SP800_160, available at http://csrc.nist.gov/publications

4. NIST_Guidelines_ForCloudSecurity_SP800-144, available at http://csrc.nist.gov/publications

5. Draft_NIST_Cloud_Computing_Synopsis_and_Reccomendation_SP800-146, available at http://csrc.nist.gov/publications
6. Donald G. Firesmith, Engineering Security Requirements, Journal of object technology, 2003, vol 2
7. Shadow_Data_Report_2015, available at http://cdn2.hubspot.net
8. CSCC Security_for_Cloud_Computing10ways-Final_080912, 2012
9. A Secure Erasure Code-Based Cloud Storage System with Secure Data Forwarding, by Hsiao-Ying Lin and Wen-Guey Tzeng, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 23, NO. 6, JUNE 2012
10. Amazon Inc. elastic Computing Cloud, November 2008. http://aws.amazon.com/ec2

11. CloudStorageArichtecture, 2012, 7th International Conference on Telecommunication Systems, Services, and Applications (TSSA), Gurudatt Kulkarni, Rani Waghmare, Rajnikant Palwe, Vidya Waykule, Hemant Bankar, Kundlik Koli.
12. Bell DE, La Padula LJ (1973) Secure computer systems: Vol. I— mathematical foundations, Vol. II—a mathematical model, Vol. III—a refinement of the mathematical model. Technical Report MTR-2547 (three volumes), Mitre Corporation, Bedford, MA, March–December
13. Apvrille A, Pourzadi M (2005) Secure software development by example. IEEE Secur Priv 3(4):10–17
14. https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology
15. http://www.enisa.europa.eu/ - Cloud Computing Security Risk Assessment.pdf
16. https://en.wikipedia.org/wiki/Open_Virtualization_Format
17. Open Source Cloud Computing Systems: Practices and Paradigms by Luis M. Vaquero, Juan cancerous& Juan J. Hierro.
18. https://en.wikipedia.org/wiki/Payment_Card_Industry_Data_Security_Standard
19. https://en.wikipedia.org/wiki/Multi-factor_authentication
20. https://en.wikipedia.org/wiki/Kerberos_(protocol)
21. https://en.wikipedia.org/wiki/Abstract_Syntax_Notation_One
22. https://en.wikipedia.org/wiki/OCSP_stapling
23. http://www.crn.com/news/security/300073621/security-researcher-warns-amazon-web-services-security-prone-to-dangerous-lapses.htm
24. AWS_Overview.pdf - https://d36cz9buwru1tt.cloudfront.net/AWS_Overview.pdf
http://aws.amazon.com/security/