File Security on Cloud using Hybrid Encryption

Thesis submitted in partial fulfillment of the requirements for the award of degree

of

Masters of Technology

in

Computer Science and Applications

Submitted By
Reema Gupta
(Roll No. 601103015)

Under the supervision of
Dr. Rajesh Kumar
Associate Professor, SMCA

School of Mathematics and Computer Applications
Thapar University
Patiala –147004

July 2013

Certificate

I hereby certify that the work which is presented in the thesis entitled, "**File Security on Cloud using Hybrid Encryption**", in partial fulfillment of the requirements for the award of degree of the Master of Technology in **Computer Science and Applications**, submitted in School of mathematics and Computer Applications of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of **Dr. Rajesh Kumar** and refers others researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other university.

Reema

Reema Gupta
601103015

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

21/06/

Dr. Rajesh Kumar
Associate Professor
School of Mathematics and Computer Applications

Countersigned by

Dr. Rajesh Kumar
Head
School of Mathematics and Computer Applications
Thapar University
Patiala

Dr. S.K. Mohapatra
Dean Academic Affairs
Thapar University
Patiala

Acknowledgment

I would like to acknowledge everyone who supported me during my experience at Thapar University and my work on this thesis. First of all I would like to thank my family who stayed with me and supported me all the time while I made this dream come true. I extend my thanks to my thesis supervisor, Dr. Rajesh Kumar, who helped me to materialize my ideas on this thesis. His enthusiasm and optimism made this experience both rewarding and enjoyable. Most of the novel ideas and solutions found in this thesis are the result of our numerous stimulating discussions. His feedback and editorial comments were also valuable for writing this thesis.

My thesis work would be incomplete without thanking my friends who were always there in the hour of need.

Abstract

Cloud computing is next generation architecture of IT Enterprise. In contrast to traditional solutions, Cloud computing moves the application software and files to the large data centers, where the management of the files and services may not be fully trustworthy. This unique feature, however, raises many new security challenges which have not been well understood.

The resources of Cloud are shared among all of the servers, users and individuals. As such, it is difficult for the cloud provider to ensure file security. As a result, it is very easy for an intruder to access, misuse and destroy the original form of data. In case of compromise at any cost; entrusting cloud is of no use. There is need to have "practically infeasible to attack" solution for file information preserving system on cloud.

The proposed approach uses hybrid cryptosystem. If the data stored in the cloud is in encrypted form, it would effectively solve various security issues. The proposed hybrid cryptosystem uses a combination of Blowfish Algorithm coupled with file splitting and merging mechanism and SRNN Algorithm. The performance of symmetric algorithm is integrated with security of asymmetric algorithm.

List of Figures

List of Tables

Table of Contents

Chapter 1

Introduction

1.1 Cloud Computing

Cloud computing originated from large-scale distributed computing technology, provides users with computation power, storage, and software services. It does not require knowledge of physical location or specific hardware configuration of where the services are running. It runs on the same logic that applies to a power grid providing energy to different houses, where a consumer does not need to know how the power is produced or how it is delivered to the house.

As per NIST definition [1] of Cloud Computing - "A model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction". The primary idea of cloud computing is that organizations no longer manages and own their IT infrastructure, but have it delivered as a service by a Cloud Service Provider (CSP). Over the last few years, there is a vogue to outsource more and more of IT services to external parties.

Cloud computing has been coined as an umbrella term to describe a category of sophisticated on-demand computing services initially offered by commercial providers, such as Amazon, Google, and Microsoft. It denotes a model on which a computing infrastructure is viewed as a "Cloud", from which businesses and individuals access applications from anywhere in the world on demand.

1.2 Foundation of Cloud Computing

The emergence of Cloud computing [2] is closely linked to technologies such as grid computing, utility computing etc.

1.2.1 Grid Computing

IBM defines grid computing as "the ability, using a set of open standards and protocols, to gain access to applications, data, processing power, storage capacity and a vast array of other computing resources over the Internet."

Grid computing enables aggregation of distributed resources and transparently. A key aspect of the grid vision realization has been building standard Web services-based protocols that allow distributed resources to be "discovered, accessed, allocated, monitored, accounted for, and billed for, etc., and in general managed as a single virtual system."

The development of standardized protocols [2] for several grid computing activities has contributed—theoretically—to allow delivery of on-demand computing services over the Internet. Lack of performance isolation has prevented grids adoption in a variety of scenarios, especially on environments where resources are oversubscribed or users are uncooperative. Activities associated with one user or virtual organization (VO) can influence, in an uncontrollable way, the performance perceived by other users using the same platform.

1.2.2 Utility Computing

"Utility computing [2] is a service provisioning model in which a service provider makes computing resources and infrastructure management available to the customer as needed, and charges them for specific usage rather than a flat rate."

In utility computing environments, users assign a "utility" value to their jobs, where utility is a fixed or time-varying valuation that captures various QoS constraints (deadline, importance, satisfaction). The valuation is the amount they are willing to pay a service provider to satisfy their demands. This is different from the conventional computing model as in utility computing, the customers do not have to invest in owning (peak need) resources anymore, but only are billed for the actual use of resources. Providers can choose to prioritize high yield (i.e., profit per unit of resource) user jobs, leading to a scenario where shared systems are viewed as a marketplace, where users compete for resources based on the perceived utility or value of their jobs.

1.2.3 Cluster Computing

A computer cluster consists of a set of loosely connected or tightly connected computers that work together so that in many respects they can be viewed as a single system. The components of a cluster are usually connected to each other through fast local area networks ("LAN"), with each node (computer used as a server) running its own instance of an operating system. Computer clusters emerged as a result of convergence of a number of computing trends including the availability of low cost microprocessors, high speed networks, and software for high performance distributed computing.

Being much more cost-effective than single computers of comparable speed and availability, clusters are usually deployed to improve performance and availability over that of a single computer,

1.3 Cloud Computing Deployment Models

Clouds can also be classified based upon the underlying infrastructure deployment model as Public, Private, Community, or Hybrid clouds. These deployment models describe who owns, manages and who is responsible for the services

1.3.1 Private Cloud

Private cloud is defined as “Internal data center of a business or other organization, not made available to the general public.”

In a Private cloud, the services are completely dedicated to the particular customer; resources are not shared with other customers. The cloud infrastructure is operated solely for an organization. It may be managed by the organization or a third party and may exist on premise or off premise [4].

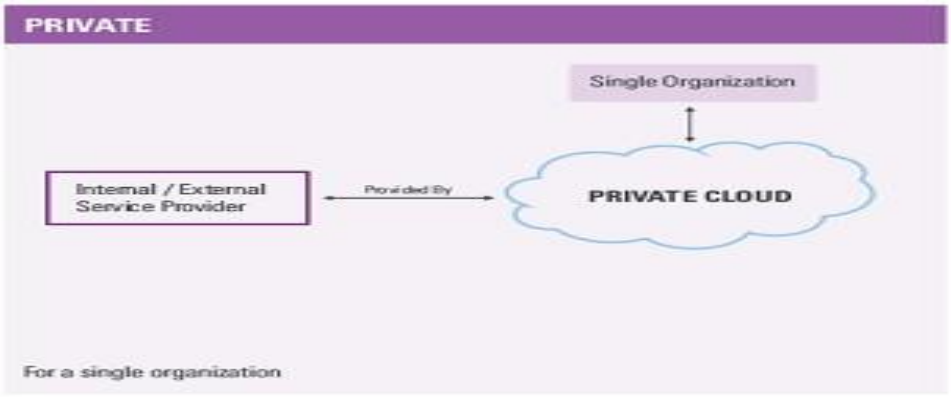

Figure 1.1: Private Cloud [3]

1.3.2 Public Cloud

Public cloud is defined as a "Cloud made available to the general public in a pay as you go manner". In a Public cloud, the delivered services are shared with other customers. The Cloud infrastructure is made available to the general public and is owned by a provider selling cloud services [4].

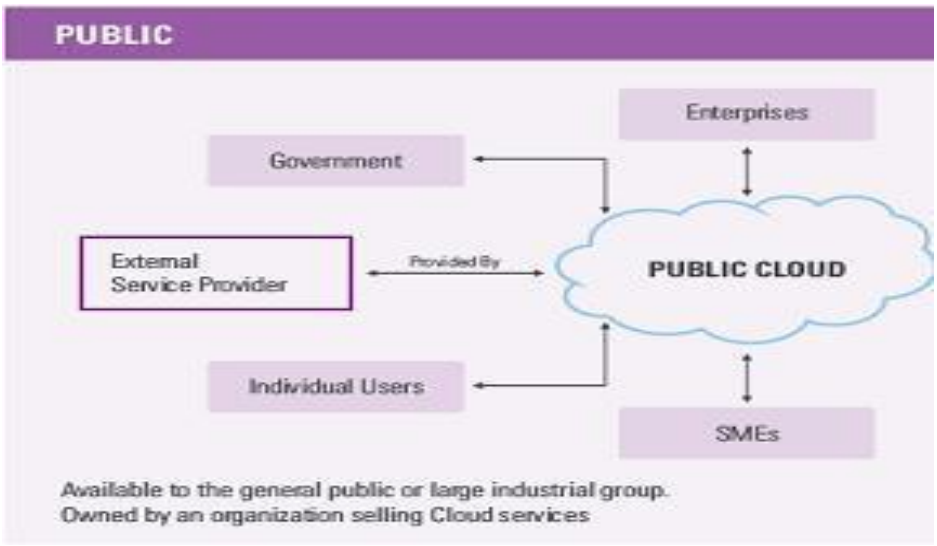

Figure 1.2 Public Cloud [3]

1.3.3 Community Cloud

A community cloud is defined as a "cloud infrastructure shared by the several organizations and supports a specific community that has shared concerns e.g., mission, security requirements, policy, and compliance considerations".

The community cloud combines aspects of the private cloud and public cloud, resources are shared, but only with other customers that have the same requirements. The cloud infrastructure is shared by several organizations and supports a specific community that has shared concerns. It may be managed by the organizations or a third party and may exist on premise or off premise [4].

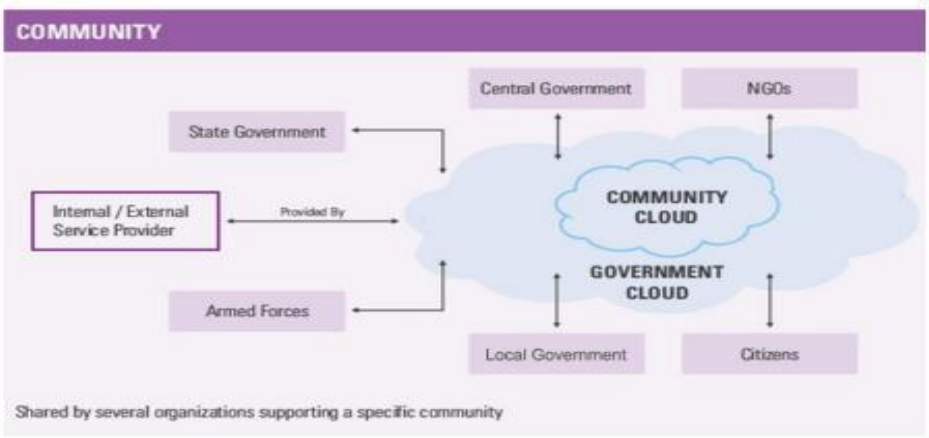

Figure 1.3: Community Cloud [3]

1.3.4 Hybrid Cloud

A hybrid cloud combines multiple deployment models. Hybrid cloud is a composition of two or more clouds (private, community, or public) that remain unique entities but are bound together, offering the benefits of multiple deployment models. It can also be defined as multiple cloud systems that are connected in a way that allows programs and data to be moved easily from one deployment system to another [4].

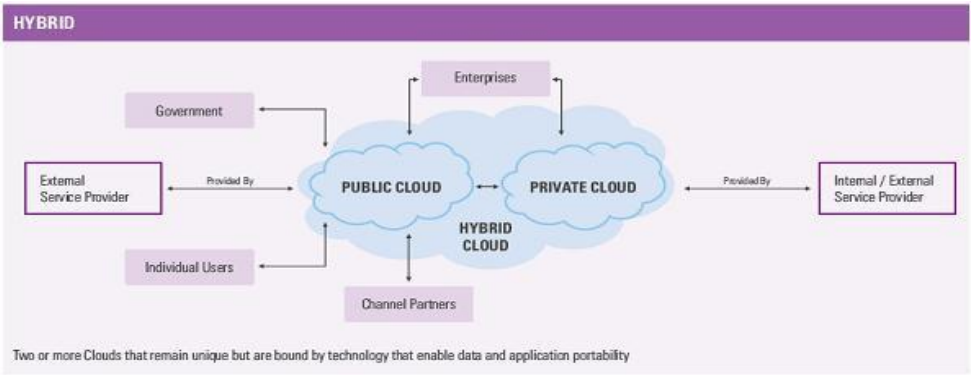

Figure 1.4: Hybrid Cloud [3]

1.4 Cloud Computing Service Models

To be able to talk about more specific about the services, cloud computing can be split into three service models, Software, Platform and Infrastructure as a Service [1]. These service models depict the degree of service or control of the Cloud Service Provider (CSP) on services and the degree of freedom possessed by a customer. Figure 1.5 gives a graphical representation of different service models and their components. The blue blocks are managed by the customer and grey block are managed by the CSP.

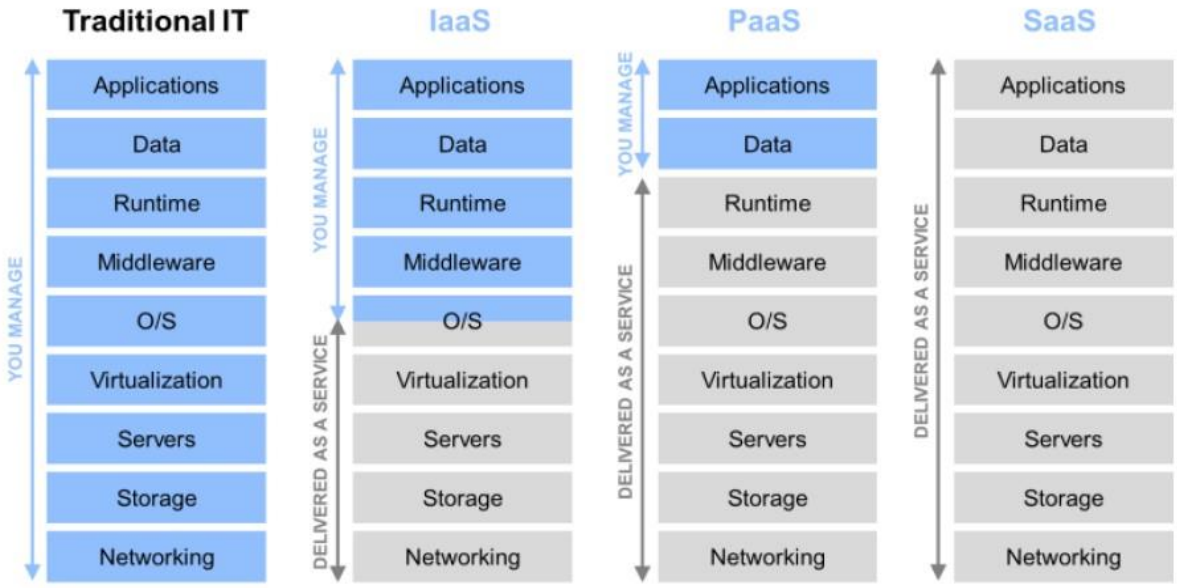

Figure 1.5: Cloud Computing Service Models [1]

1.4.1 Infrastructure as a Service (IaaS)

Infrastructure as a Service (IaaS) is a service provisioning model in which an organization outsources the equipment used to support operations, including storage, hardware, servers and networking components. The service provider owns the equipment and is responsible for housing, running and maintaining it. The client typically pays on a per-use basis.

As per NIST,

"IaaS is a capability provided to the consumer to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The cloud customer does not manage or change parameters of the underlying infrastructure but has control over operating systems, storage, deployed applications, and possibly limited control of select networking components (e.g., host firewalls)."[1]

1.4.2 Platform as a Service (PaaS)

Platform as a Service is a delivery model that allows the customers to rent virtualized servers and associated services for running existing applications or developing and testing new ones. In case of PaaS, the cloud provider not only provides the hardware, but they also provide a toolkit and a number of supported programming languages to build higher level services.

As per NIST,

"PaaS is a capability provided to customer to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly application hosting environment configurations"[1].

1.4.3 Software as a Service (SaaS)

Software as a Service (SaaS) is software distribution model in which applications are hosted by a vendor or service provider and made available to customers over a network

As per NIST,

"SaaS is a capability to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through a thin client interface such as a web browser. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings"[1] .

1.5 Characteristics of Cloud

The cloud model mainly contains following five characteristics [5]:

- Resource pooling- The resources are assigned and reassigned to the users as per the consumer demand. The users have no knowledge over the exact location of the resources that are provided.
- On-demand self-service- Without requiring human interaction with every service's provider, a consumer can unilaterally pro-vision computing capabilities, such as server time and net-work storage, as needed automatically.
- Rapid elasticity- To the users the capabilities available for provisioning often appear to be infinite and can be purchased in any extent at any time.
- Broad network access: All the services are available over the network and they can be accessed by any devices such as mobile phones, laptops etc.
- Measured Service- Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

1.6 Commercial Cloud Providers

The cloud providers in market are differentiated on the basis of the technique being employed. Some of the cloud providers are as follows:

1.6.1 Amazon

Amazon was one of the first companies to offer cloud services to the public, and they are very sophisticated. Amazon offers a number of cloud services [6], which includes:

- **Elastic Compute Cloud (EC2)**: It offers virtual machines and extra CPU cycles for any organization.
- **Simple Storage Service (S3)**: It allows to store items up to 5GB in size in Amazon's virtual storage service.
- **Simple Queue Service (SQS)**: It allows machines to talk to each other using this message-passing API.
- **Simple DBA**: Simple DBA web service is used for running queries on structured data in real time. This service works in close conjunction with Amazon Simple Storage Service (AmazonS3) and Amazon Elastic Compute Cloud (Amazon EC2), collectively providing the ability to store, process, and query data sets in the cloud.

1.6.2 Google App Engine

In stark contrast to Amazon's offerings is Google's App Engine. On Amazon we get root privileges, but on App Engine, we can't write a file in our own directory. Google removed the file write feature out of Python as a security measure, and to store data we must use Google's database. Google offers online documents and spreadsheets, and encourages developers to build features for those and other online software, using its Google App Engine. Google reduced the web applications to a core set of features, and built a good framework for delivering them.

1.6.3 Window Azure

Microsoft's cloud computing solution is called Windows Azure, an operating system that allows organizations to run Windows applications and store files and data using Microsoft's datacenters. It's also offering its Azure Services Platform, which are services that allow developers to establish user identities, manage workflows, synchronize data, and perform other functions as they build software programs on Microsoft's online computing platform.

1.7 Open Source Cloud Computing Frameworks

Eucalyptus, OpenNebula and Nimbus are three major open-source cloud-computing software platforms which provide an important alternative who do not wish to use a commercially provided cloud. The overall function of these systems is to manage the provisioning of virtual machines for a cloud providing infrastructure-as-a-service. Open Cloud platforms offers real alternatives to end-user for improved flexibility and on demand services.

1.7.1 Eucalyptus

Eucalyptus is termed as Elastic Utility Computing Architecture for linking programs to useful systems [7]. It is a Linux-based software architecture that implements scalable, efficiency-enhancing private and hybrid clouds within an organization's IT infrastructure. Eucalyptus uses computational and storage infrastructure for academic research groups and provides a platform that is modular and open to experiment. The system allows users to start, control, access, and terminate entire virtual machines using an emulation of Amazon EC2‟s SOAP and Query interfaces. One striking feature of Eucalyptus, is its choice of the Amazon AWS APIs. The current interface to Eucalyptus is compatible with Amazon's EC2 interface and uses the EC2 tools directly and duplicates Simple Storage Service (S3) service. Eucalyptus implements a distributed storage system called Walrus which is designed to imitate Amazon’s S3 distributed storage. The infrastructure is designed to support multiple client-side interfaces. Eucalyptus is implemented using commonly available Linux tools and basic Web-service technologies making it easy to install and maintain.

1.7.2 Nimbus

Nimbus is an open-source toolkit focused on providing Infrastructure-as-a-Service (IaaS) cloud to its client via WSRF-based or Amazon EC2 WSDL web service APIs [7]. Nimbus project explicitly advertises itself as a “science” cloud solution. However Nimbus has supported many nonscientific research domain applications. Nimbus v2.9 is incredibly customizable. Nimbus supports the Xen hypervisor and virtual machine

schedulers PBS and SGE. It allows deployment of self-configured virtual clusters via contextualization. It is configurable with respect to scheduling, networking leases, and usage accounting. Nimbus provides a complementary tool Cumulus implementation of a quota-based storage cloud designed for scalability and allows providers to configure multiple storage cloud implementations. Nimbus offers scaling tools allowing users to automatically scale across multiple distributed providers, these tools "sky computing tools" operate in a multi-cloud environment combining private and public cloud capabilities.

1.7.3 OpenStack

OpenStack launched in July 2010 is an initiative of Rackspace Hosting and NASA. OpenStack is designed to create freely available code, standards, and common ground for the benefit of both cloud providers and cloud customers. The goal of OpenStack 2.0 is to allow organization to create and offer cloud computing capabilities using open source software running on standard hardware. The project boasts of compute, storage and image service component. OpenStack is open source software designed to provision and manage large networks of virtual machines, creating a redundant and scalable cloud computing platform. It has the software, control panels, and APIs required orchestrating a cloud, including running instances, managing networks, and controlling access through users and projects**.**

1.7.4 OpenNebula

OpenNebula is a fully open-source tool [7] kit to build any type (private, public and hybrid) of infrastructure based cloud. OpenNebula is platform agnostic with broad hypervisor support, allowing to leverage the existing IT infrastructure. The cloud provides infrastructure users with an elastic platform for fast delivery and scalability of services to meet dynamic demands of end-users. It allows the user to dynamically host the services in VMs, enables monitoring and control using interfaces like command line interface, XML-RPC API, Libvirt virtualization API. OpenNebula manages the data center of private cloud and infrastructure of cluster running Xen, KVM or VMware and also support hybrid cloud to connect local and public infrastructure which is very useful

to build highly scalable cloud computing environment. OpenNebula supports heterogeneous execution environments with multiple, even conflicting, software requirements **on** the same shared infrastructure with full control of the lifecycle of virtualized services management**.**

1.8 Security Issues in Cloud

Cloud computing has become a popular choice as an alternative to investing new IT systems. When making decisions on adopting cloud computing related solutions, security has always been a major concern.

Some of the security issues are:

- **Fault tolerance and service availability**: When keeping data at remote systems owned by others, data owners may suffer from system failures of the service provider, as system failures will mean that data will become unavailable if the data depends on a single service provider. Similarly, when deploying IT systems over a single cloud, services may be unavailable if the cloud goes out of operation.
- **Data confidentiality and integrity:** Data generated by cloud computing services are normally kept in the clouds as well. Keeping data in the clouds means users may lose control of their data and rely on cloud operators to enforce access control [23, 2], thus they may not be able to prevent unauthorized disclosure or malicious modification of their data.
- **Data Stealing:** Data stealing is one of the serious issues in a cloud computing environment. Data stealing refers to the illegal acquisition of information. Many cloud service providers do not provide their own server instead they lease server from other service providers due to it is cost effectiveness and flexibility. So there is a high probability that data can be stolen from the external server.
- **Infected Application:** In the cloud computing environment**,** there exists a possibility where a malicious user can penetrate the cloud by acting as a legitimate user, thereby infecting the entire cloud and thus affecting many customers who are sharing the infected cloud. It is the responsibility of the service provider to prevent any malicious user from uploading any infected application

onto the cloud. Cloud computing service provider should have the complete access to the server with all rights for the purpose of monitoring and maintenance of server.

- **Loss of Control**: When organizations port their data or services to cloud, they are not aware of the location of their data and services. Since, the provider can host their data or services anywhere within the cloud. This poses a serious concern as from a user perspective; organizations lose control over their vital data and are not aware of any security mechanisms put in place by the provider.

1.9 Data as a Service (DaaS)

Data as a Service (DaaS) [8] in cloud is a strategy used to facilitate the accessibility of business-critical data in a well-timed, protected and affordable manner. It is an information provision and distribution model in which data files (including text, images, sounds, and videos) are made available to customers over a network, typically the Internet. DaaS offers convenient and cost-effective solutions for customer- and client-oriented enterprises. Providing data as a service has not only fostered the access to data from anywhere at any time but also reduced the cost of investment.

1.9.1 File Security on Cloud

Cloud computing becomes the next generation architecture of IT enterprise. In contrast to traditional solutions, cloud computing moves the application software and files to the large data centers, where the management of the files and services may not be fully trustworthy. This unique feature, however, raises many new security challenges which have not been well understood.

In the cloud environment, resources are shared among all of the servers, users and individuals [9]. As a result files or data stored in the cloud become open to all. Therefore, data or files of an individual can be handled by all other users of the cloud. Thus the data or files become more vulnerable to attack. As a result it is very easy for an intruder to access, misuse and destroy the original form of data. Besides, cloud service providers

provide different types of applications which are of very critical nature. Hence, it is extremely essential for the cloud to be secure [10].

Another problem with the cloud system is that an individual may not have control over the place where the data needed to be stored. A cloud user has to use the resource allocation and scheduling, provided by the cloud service provider. Thus, it is also necessary to protect the data or files in the midst of unsecured processing. In order to solve this problem we need to apply security in cloud computing platforms.

The files stored in the cloud storages is similar with the ones stored in other places and needs to consider three aspects of information security: confidentiality, integrity and availability. Confidentiality and integrity of file transmission needs to ensure not only between enterprise storage and cloud storage but also between different cloud storage services.

Every cloud provider solves this problem by encrypting the data by using various encryption algorithms. The files stored at the cloud storage server are in encrypted form which can only be extracted if its corresponding key is known. Hence, cloud manages to satisfy confidentiality, integrity and availability of data.

1.9.2 Key Components of Data Security

There are four key components of data security:

- **Availability**: Data availability ensures continuous access to data even in the event of a natural or man-made disaster or events such as fires or power outages.
- **Integrity**: Data integrity ensures that the data is maintained in its original state and has not been intentionally or accidentally altered.
- **Confidentiality**: Data confidentiality means information is available or disclosed only to authorize individuals, entities, or IT processes.
- **Traceability**: Data traceability means that the data, transactions, communications, or documents are genuine and that both parties involved are who they claim to be.

1.10 Cryptography

"Cryptography is the art and science of designing or generating the secret message i.e. code or ciphers of the original message for the secure communication between sender and the receiver".

A cryptographic algorithm is a set of mathematical functions and unchanging set of steps to perform encryption and decryption of the original data.

Usually, two related functions are used, one for encryption and the other for decryption. With most modern cryptography, the ability to keep encrypted information secret is based not on the cryptographic algorithm, which is widely known, but on a number called a *key* that must be used with the algorithm to produce an encrypted result or to decrypt previously encrypted information.

The main objective of every cryptographic algorithm is to make it as difficult as possible to decrypt the generated cipher text without using the key.

1.10.1 Encryption

It is a process to transform or convert the data into some another form that appears to be random, meaningless and unintelligible. This is usually accomplished using a secret encryption key and a cryptographic cipher. It can also be said that encryption is the process of transforming plaintext into the cipher text where plaintext is the input to the encryption process and cipher text is the output of the encryption process. It is considered as the subset of cryptography. Encryption is the most effective way to achieve data security.

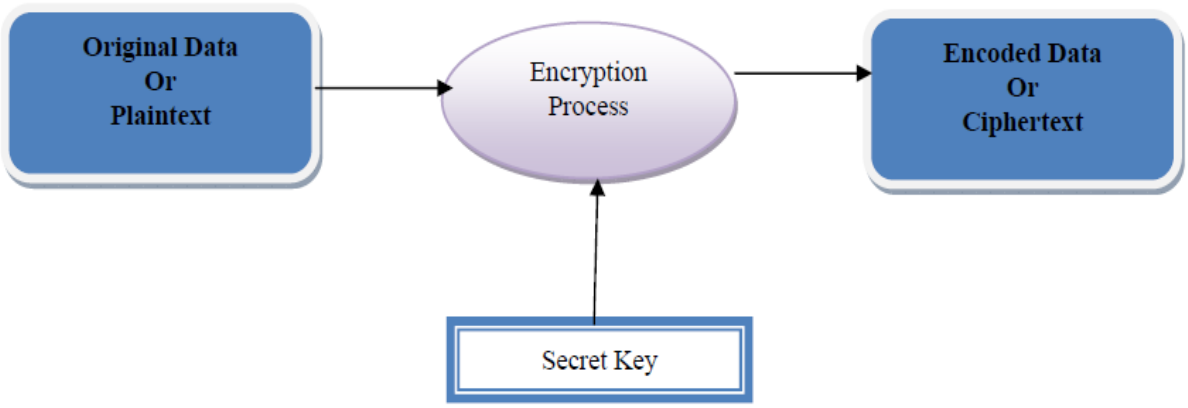

Figure 1.6: Encryption Process [11]

1.10.2 Decryption

It is a process to transform or convert the encoded data into some meaningful form. It can also be said that decryption is the process of transforming cipher text into the plaintext where cipher text is the input to the decryption process and plaintext is the output of the decryption process. Decryption without the correct key is very difficult, if not impossible.

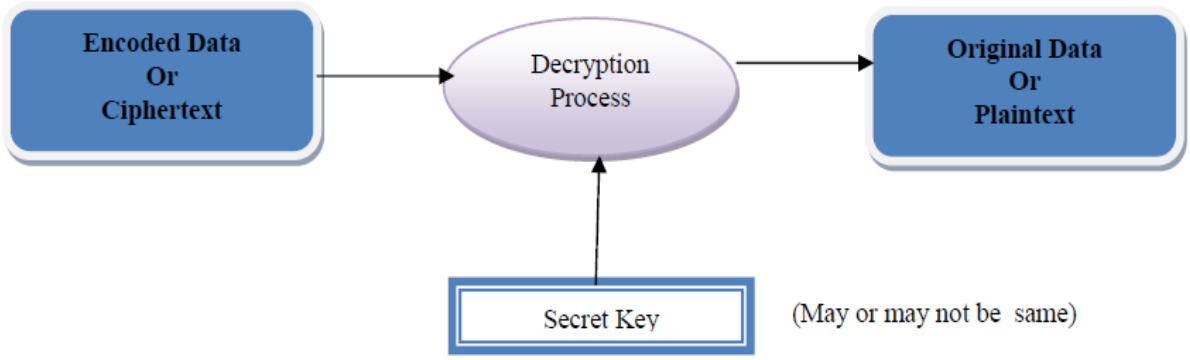

Figure 1.7: Decryption process [11]

1.10.3 Goals of Cryptography

The main goals of cryptography are:

- **Confidentiality**: It is the process of keeping information secret from all, but those who are authorized to see it. Confidentially is the protection of transmitted

data from passive attacks. With respect to the content of data transmission, several levels of protection can be identified. The broadest service protects all user data transmitted between two users over a period of time. The aspect of confidentially is the protection of traffic flow from analysis. This requires that an attacker should not be able to observe source and destination, frequency, length or any other characteristics of the traffic on a communication network.

- **Integrity**: It ensures that the information has not been altered by unauthorized or unknown means. One must have the ability to detect data manipulation by unauthorized parties. Data manipulation includes such things as insertion, deletion, and substitution.
- **Authentication**: It is a service related to identification. Data Authentication implicitly provides data integrity and it is applied to both entities and information itself, because any two parties entering into a communication should identify each other. Information delivered over a channel should be authenticated as to data origin, data content.
- **Non- repudiation**: Non-repudiation prevents either sender or receiver from denying a message. Thus, when a message is sent, the receiver can prove that the message was sent by the alleged sender. Similarly, when a message is received, the sender can prove that the alleged receiver received that message.

1.11 Cryptographic Techniques

There are two types of cryptographic techniques:

i) Symmetric Key Encryption
ii) Asymmetric Key Encryption

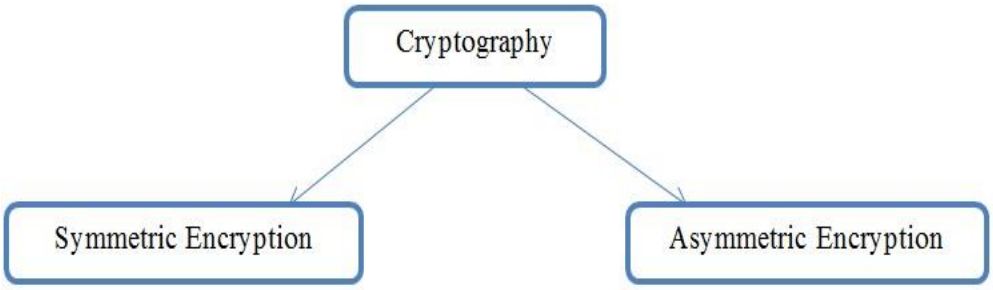

Figure 1.8: Types of Cryptographic Algorithms [11]

1.11.1 Symmetric Cryptography

It uses only one key for both encryption and decryption process. The key is transmitted to both the sender and receiver before the process of encryption and decryption [12]. In the symmetric-key encryption, the encryption key can be calculated from the decryption key and vice versa. The secret key plays an important role and its strength depends on the length of key.

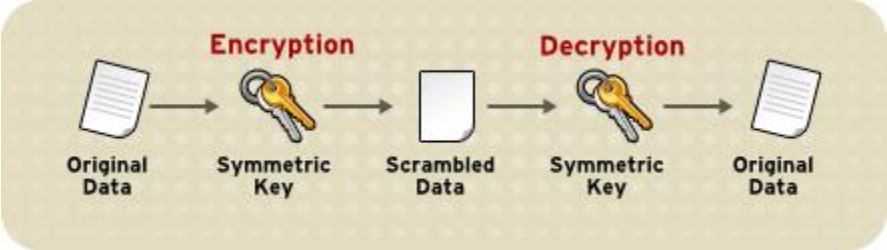

Figure 1.9: Symmetric Key Encryption [12]

1.11.2 Asymmetric Cryptography

Asymmetric-key encryption (also called public-key encryption) involves a pair of keys, a public key and a private key, associated with an entity. Each public key is published, and the corresponding private key is kept secret [12]. Data encrypted with a public key can be decrypted only with the corresponding private key.

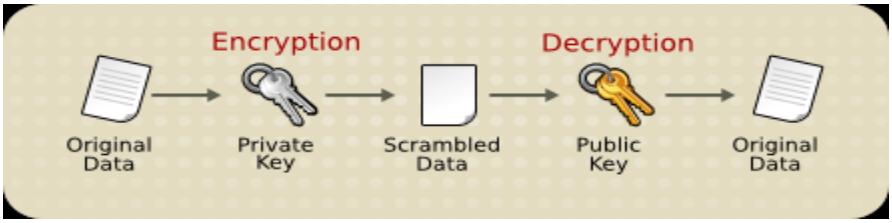

Figure 1.10: Asymmetric Key Encryption [12]

This cryptographic scheme allows public keys to be freely distributed, while only authorized people are able to read encrypted data using their corresponding private keys. In general, to send encrypted data, the data is encrypted with that person's public key, and the person receiving the encrypted data decrypts it with the corresponding private key.

Compared with symmetric-key encryption, public-key encryption requires more processing and may not be feasible for encrypting and decrypting large amounts of data. However, it is possible to use public-key encryption to send a symmetric key, which can then be used to encrypt additional data. This approach is used by the SSL/TLS protocols.

1.12 Hybrid Cryptosystem

Symmetric ciphers are significantly faster than asymmetric ciphers, but require the communicating parties to somehow share a secret key. Asymmetric ciphers have the advantage of increased security and convenience over the symmetric ciphers. Asymmetric cryptosystems are convenient in the sense that they do not require the sender and receiver to share a common secret key in order to communicate securely. However, they often rely on complicated mathematical computations and are thus generally much more inefficient than comparable symmetric-key cryptosystems. The asymmetric algorithms provide a high end security, but at the cost of speed. In many applications, the high cost of encrypting long messages in a public-key cryptosystem can be prohibitive.

A hybrid cryptosystem is a protocol using multiple ciphers of different types together, each to its best advantage i.e. the one that combines the convenience of a public-key cryptosystem with the efficiency of a symmetric-key cryptosystem [16].

To encrypt a message in a hybrid cryptosystem, sender does the following:

i) Obtains receiver's public key.
ii) Generates a fresh symmetric key for the data encapsulation scheme.
iii) Encrypts the message under the data encapsulation scheme, using the symmetric key.
iv) Encrypts the symmetric key, using receiver's public key.
v) Sends both of these encryptions to receiver.

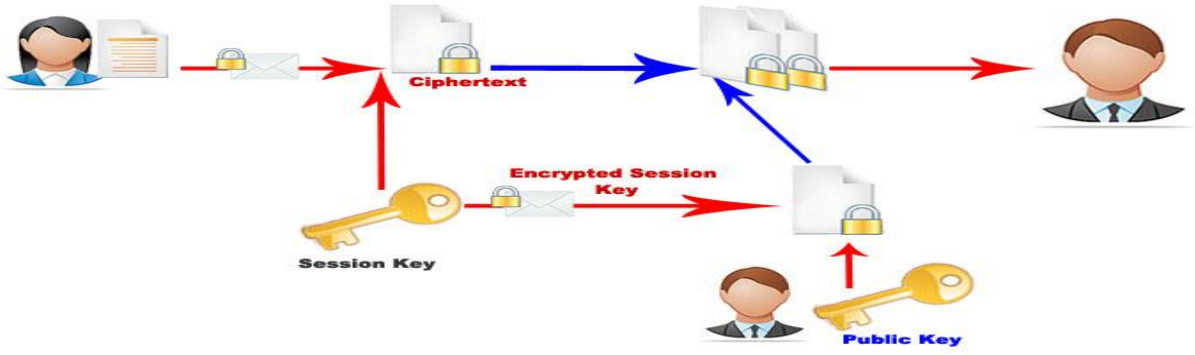

Figure 1.11: Hybrid Encryption Scheme [16]

To decrypt hybrid cipher text, receiver does the following:

i) Uses her private key to decrypt the symmetric key contained in the key encapsulation segment.

ii) Uses this symmetric key to decrypt the message contained in the data encapsulation segment.

Figure 1.12: Hybrid Decryption Scheme [16]

.

Chapter 2

Literature Survey

In this chapter, we have surveyed security issues on file on cloud environment and various encryption techniques and mapped them to our taxonomy to guide future design and development efforts.

2.1 Taxonomy of Cloud Computing Systems

Bhaskar Prasad Rimal *et.al* [4] have surveyed various cloud computing environments and services developed by various projects such as Google, force.com, amazon, open source. The surveyed results are used to identify differences, similarities and differences of the architectural approaches of cloud computing. The author defines the taxonomy and comparative study of cloud computing systems. The criteria for defining the taxonomy is based on the core ideas of distributed systems for massive data processing.

On the basis of proposed taxonomy and technical studies, the authors have evaluated the different cloud computing systems to provide necessary information that can help in future for the new development and improvement on existing systems. The proposed taxonomy provides researcher and developer the ideas on the current cloud systems, hype and challenges.

2.2 Open Source Cloud Environment

Nilesh Mangtani and Sukadha Bhingarkar [5] discussed various open source cloud environment available that provides a substitute for the users that do not hope to use a commercially provided cloud. It focused on three main platforms for the cloud development- Nimbus, OpenNebula and Eucalyptus. They are compared on the aspects of main purpose, platform architecture, networking aspects. Eucalyptus serves as the best solution to offer a public cloud service to a community in a scenario of homogeneous pool of Xen or KVM hypervisors The public, private, hybrid cloud can be created by OpenNebula. It provides IaaS with more diversified and powerful functionalities, as

virtual machine migration and a more useful resource allocation mechanism. Overall, OpenNebula seems to be more suited for research and experimental studies.

Anita S. Pillai and L.S. Swasthimathi [7] has analyzed the characteristics, architecture and applications of some of the most popular open source cloud computing platforms like Eucalyptus, OpenStack, Nimbus and OpenNebula. The main objective is to perform a comparative analysis of various cloud computing platforms to facilitate novice users to choose an open cloud platform depending on their requirements such as cloud types, interfaces, compatibility, implementation, deployment requirement, and development support.

The study also identifies challenges common to the platform and areas for improvement. For the further enhancement in open source platforms, it is suggested to incorporate more features to improve their framework.

Wen *et al*. [17] have described the function of OpenStack and OpenNebula briefly, and then compared them with provenance, architecture, hypervisors, security and other angles in detail. Moreover, the authors have provided some deployment recommendations according to different user demands and platform characteristics.

After the detailed analysis, the authors have concluded that the OpenStack can offer the same services as Amazon and be the alternative Amazon to users who don't want to use a billing cloud. Thus, it is more suitable platform for enterprises as they can not only acquire the resources dynamically but also can encapsulate its services as some applications. Also, OpenNebula is well suitable for dealing with large amounts of data, leveraging existing IT infrastructures and avoiding vendor lock-in. Thus, it is suitable for research institutes, universities and enterprises especially large data centers which want to build an open, flexible and scalable cloud to support their research. It is also ideal for users who want to run a number of cloud machines quickly.

Yang *et al*.[18] have proposed a scheme to build video services on cloud environment, which integrates KVM and OpenNebula open sources to provide a cloud virtual environment for end users. The proposed work integrates both Hadoop distributed file

system (HDFS) and Nutch Search Engine in KVM-based cloud computing environment with video streaming related applications. This system can perform distributed computation in Map-Reduced programming in order to sufficiently shorten the time spent in searching indexes space construction. It uses IaaS and PaaS to construct a feature of cloud computing, which is a profound video cloud service with distributed file storage and a searching engine.

The proposed system is developed by applying open source: OpenNebula is used to build the IaaS environment, Hadoop is used as PaaS, and the application of cloud computing is realized by SaaS. As an IaaS provider, the proposed framework applies the idea of virtualization in the cloud system to economize power, web interfaces and user friendly ways to manage the virtual machines.

2.3 Cloud Security Concerns

Tianfield [19] has presented a comprehensive study on the challenges and issues of security in cloud computing. The author has firstly discussed the impacts of the distinctive characteristics of cloud computing, namely, multi-tenancy, elasticity and third party control, upon the security requirements. Further, he has analyzed the cloud security requirements in terms of the fundamental issues, i.e., confidentiality, integrity, availability, trust, and audit and compliance. Furthermore, the taxonomy for security issues in cloud computing is discussed.

Akhil Bedi [20] addressed emerging security challenges in cloud computing. The research is focused on developing a comprehensive cloud-aware security strategy that can meet the aforesaid research challenges and have the ability to defend the cloud infrastructure and the different layers (including network connections, data at rest, data in transit, applications and VMs) against threats which may arise from within the provider's network or from outside. The security strategy is intended to leverage the existing security (ad-hoc) technologies and employing them in a fluid and dynamic cloud environment.

The author concluded that it is essential to recognize the fact that there is no silver bullet to counter the threats to distributed or cloud computing model. It is however, even more

important to appreciate that with right security strategy, multiple layers of security, and implementing well thought after security controls, it is possible to restrain threats.

Chen *et al.* [10] have provided a concise but all-round analysis on data security and privacy protection issues associated with cloud computing across all stages of data life cycle. The authors have discussed various security issues linked with the data life cycle process from generation to destruction of the data. The main objective is to design a set of unified identity management and privacy protection frameworks across applications or cloud computing services.

On the basis of the analysis of security and privacy issues, the authors have determined that it is necessary to separate the sensitive data and access control in order to achieve the cloud security. It is also suggested to have an integrated and comprehensive security solution to meet the needs of defense in depth. Moreover, authorization and access control mechanisms need to achieve a unified, reusable and scalable access control model and meet the fine grained access authorization.

Eman M.Mohamed [9] discussed the basic problem with cloud computing data security. The data security model of cloud computing based on the study of the cloud architecture is presented. The software is implemented to enhance work in a data security model for cloud computing. Finally the software is tested in the Amazon EC2 Micro instance.

NIST statistical tests results are used to select the highest security algorithm. The test is applied on eight modern encryption algorithms namely RC4, RC6, MARS, AES, DES, 3DES, Two-Fish, and Blowfish. In terms of performance, Blowfish and AES can be used for encrypting decrypting data on Amazon platform.

Wadhawan *et al.* [21] have presented an elaborated study of security holes associated with IaaS implementation and their countermeasures. Infrastructure as a Service (IaaS) serves as the foundation layer for the other delivery models, and a lack of security in this layer also affects the other delivery models .The authors have proposed a Security Model for IaaS (SMI) as a guide for assessing and enhancing security in each layer of IaaS delivery model. The proposed model consists of three sides: IaaS components, security

model, and the restriction level. This model indicates the relation between IaaS components and security requirements, and eases security improvement in individual layers to achieve a total secure IaaS system.

2.4 Cloud Computing Security Architecture

Kawser *et al*. [22] have proposed new security architecture for exchanging information in a cloud computing environment. In order to ensure a secure communication process, the proposed architecture includes AES file encryption system, RSA system for secure communication, onetime password to authenticate users and MD5 hashing for hiding information. The main objective behind the proposed model is to solve main security issues like malicious intruders, hacking, etc. in cloud computing platform. Additionally, distributive server concept is used, thereby ensuring higher security for the whole cloud.
Experimental results have also been shown to determine the efficiency of the proposed model. From the comparative analysis, it is demonstrated that the proposed model works smoothly and ensures higher security than other present running models in a cloud computing environment.
Though the proposed architecture is highly secure, but RSA encryption system becomes fragile in long run process. Thus, there is a need to find out a light and secure encryption system for secure communication and hiding information from others.
In terms of performance, Blowfish is more advantageous than AES, hence the above cloud computing architecture can be enhanced by using Blowfish in place of AES.

Debajyoti *et al*.[23] proposed a methodology to overcome security threats which emerge with shared spaces in the cloud environment. The proposed framework involves securing of files through file encryption mechanism. The file present on the device is encrypted using password based AES algorithm. The user can download any of the uploaded encrypted files and read it on the system after the successful decryption. The suggested scheme aims at preventing every possible attack on the user data existing on the cloud. The integrity and confidentiality of the data uploaded by the user is ensured doubly by not only encrypting it but also providing access to the data only on successful authentication.

The authors have implemented the proposed framework to encrypt only text files which can be further enhanced to encrypt the audio and video files. Additionally, this framework can also be integrated with the social networking sites to exchange data securely in its encrypted form.

Mohta *et al.* [24] has presented a way to implement Third Party Auditor (TPA) who not only checks the reliability of Cloud Service Provider (CSP) but also checks the consistency and accountability of data. The author has also discussed the challenges of open issue of integrity and data dynamics. The main objective is to solve the problem of data privacy, accountability and integrity of data.

The proposed model involves the clients, cloud service provider and TPA. The suggested scheme involves retrieval of file, encryption and decryption of file, integrity checking from CSP and procedure for giving control to TPA. The responsibility of the TPA is to audit the cloud data storage efficiently. CSP provides service to client on successful authentication. Files are encrypted and decrypted using RSA algorithm. Message digest is generated using SHA-512 algorithm. After performing the desired file operation, clients send the data to CSP and TPA. The proposed scheme maintains the consistency at cloud data storage for CSP and client and also checks the integrity of data.

2.5 Cloud Security Deployment Models

Gansen Zhao *et al.* [25] proposed five service deployment models to ease security concerns of cloud computing. The models provide different security related features to address different requirements and scenarios and can serve as reference models for deployment. The proposed deployment models are separation model, cryptographic model, migration model, availability model and Tunnel model.

The proposed models rely on inter-cloud interaction and require multiple clouds to cooperate. The models are user oriented. Design and implementation techniques and methods are development oriented and are opaque for users. The deployment models require the cooperation of multiple clouds and create user awareness on it. By doing this, users trust in deploying IT systems on cloud computing would be increased. The author also discussed the security concerns that users may have when adopting cloud computing,

including fault tolerance and service availability, data migration, and data confidentiality and integrity.

2.6 Comparative Analysis of Blowfish over other Symmetric Algorithms

Diaa Salamass *et.al* [26] evaluated six of the most common encryption algorithms namely: AES (Rijndael), DES, 3DES, RC2, Blowfish, and RC6. A comparison has been conducted for those encryption algorithms at different settings for each algorithm such as different sizes of data blocks, different data types, battery power consumption, different key size and finally encryption/decryption speed. Simulation results are given to demonstrate the effectiveness of each algorithm.

It can be concluded from the simulation results that in the case of changing packet size, Blowfish has better performance than other common encryption algorithms used, followed by RC6. The results show 3DES still has low performance compared to algorithm DES. Finally it is demonstrated that higher key size leads to clear change in the battery and time consumption

Chhaya Nayak [27] discussed the performance of selected symmetric encryption algorithms under different settings; the presented comparison takes into consideration the behavior and the performance of these algorithms when different data loads are used. The selected algorithms are DES, 3DES, RC5, Blowfish and IDEA. The security of algorithms and performance of a given algorithm depends on variety of parameters such as key size, block size, diffusion and confusion properties. The analytical results of the cipher are analyzed quantitatively based on the above mentioned parameters.

Fixing the performance metric is a very important process in measuring the performance of cryptosystems. The two important desirable properties of the cryptosystems are its speed and security. Speed refers to the time taken by the algorithm to convert a given plaintext to cipher text. The security of the algorithm is based on the key size. The increase in the key size reduces the speed of the algorithm but in turn increases the security. Thus, the aim of the designer is to design efficient cryptosystems with acceptable speed and appreciable security strength with large key length. Implementation procedures also play a major role in cryptosystems design.

Srinivasarao D. *et al.* [28] analyzed six most common encryption algorithms namely: DES, 3DES, RC2, Blowfish, AES (Rijndael), and RC6. A comparative study has been carried out for the above six encryption algorithms in terms of encryption key size, block size, number of Rounds, Encryption/decryption time, CPU processing time, CPU clock cycles (in the form of throughput), power consumption.

The results concluded that in the case of using the key, the Blowfish has the best use where the code is unbreakable and it has best practice to avoid data misuse, followed by AES and RC6. In the case of changing encryption/decryption Time again the Blowfish has the best performance which proves it has the best CPU clock cycles in the form of Throughput, so Blowfish is advantageous over other algorithms in terms of time consumption. Also, 3DES still has low performance compared to algorithm DES. Finally, in the case of data files, it is analyzed that Blowfish has a good performance as compared to other algorithms followed by 3DES and Rinjdal.

The cryptographic scheme can be enhanced by using combination of symmetric and asymmetric algorithm. The experimental results concluded that Blowfish becomes the best choice among other symmetric algorithms.

2.7 SRNN

Jitendra Singh Yadav *et al.*[15] proposed a method for implementing a public-key cryptosystem whose security rests in part on the difficulty of factoring large numbers. The algorithm is named as SRNN. The Short Range Natural Number (SRNN) algorithm is similar to RSA algorithm with some modification. This modification increases the security of the cryptosystem. In this algorithm, extremely large number are used that has two prime factors (similar to RSA). In addition of this two natural numbers in pair of keys (public, private) are used to increase the security of the cryptosystem.

It is demonstrated that RSA with modulus length 1024 bits is better in speed but slightly weaker in security. SRNN with modulus length 1024 bits provides a feasible solution in terms speed as well as security. So, SRNN with modulus length 1024 bits may be an optimum solution which provides a balance between speed and security.

The proposed scheme fails in terms of speed for large file sizes. Hence, this cryptographic scheme can be enhanced by using in combination with symmetric algorithms where keys can be encrypted by SRNN and file by symmetric algorithm.

2.8 File Splitting

Manikandan.G *et al.* [29] proposed a software tool which involves cryptographic enciphering and deciphering along with file splitting and merging mechanisms. Modified Blowfish algorithm is used for encryption and decryption of data. Though only one algorithm is used, cryptographic scheme is differentiated by varying the key for varying file slices.

In this approach, a file having sensitive information is sliced into desired number of pieces upon user's specification and then the cryptographic encryption phase is carried out. In order to achieve more security more than one cryptographic scheme can be adopted which definitely ensures nil suspicion and more security. The cryptographic scheme is differentiated by providing different key for each encryption of sliced files, provided the key should be given correctly at the time of decryption to avoid erroneous results. Modified Blowfish algorithm is used for encryption and decryption of data which serves as a better solution both in terms of performance and as well as security.

In the file joining phase, enciphered files thus obtained from the different enciphering techniques are merged and hence transmitted to reception side as a single file which makes the file infeasible to breach and less suspicious to get to know that varying crypto schemes are adopted. Hence, the data security is maximized. At the receiver's end, once again files are sliced and decrypts it using the same algorithm and then sliced files are joined together to retrieve the original message.

This cryptographic scheme uses symmetric algorithm which involves difficulty in transmission of symmetric key over public/insecure network. This can be enhanced by using in combination with asymmetric algorithm where symmetric keys are first encrypted using asymmetric algorithm then transmitted over public network.

2.9 Hybrid Cryptographic System

Singh *et al.* [30] have proposed a scheme that combines the security of a document by hybrid encryption method and authenticity by digital signatures. IDEA-RSA algorithm is used for hybrid cryptosystem and RSA digital signature algorithm is used to obtain digital signature. The proposed joint signature scheme uses "encrypt-then-sign" instead of, "sign-then-encrypt ". The scheme has all the features of symmetric, asymmetric algorithm and digital signature.

The proposed system has been designed and developed using C#. The effectiveness and correctness of the proposed scheme is illustrated through results and its implementation. Through the experimental results, it is determined that the proposed scheme has good throughput in contrast to RSA algorithm, but relatively less than IDEA Algorithm.

Chapter 3

Problem Statement

Previous chapter discussed related works and techniques available for file security on clouds. This chapter focuses on problem statement taken up in the thesis.

3.1 Gap Analysis

In the cloud environment, resources are shared among all of the servers, users and individuals. So, it is difficult for the cloud provider to ensure file security. As a result, it is very easy for an intruder to access, misuse and destroy the original form of data. In case of compromise at any cost; entrusting cloud side is of no use. A need for "practically strong and infeasible to get attacked" technique becomes vital.

In literature survey various related works for file security on cloud. On the basis of literature survey, following gaps are drawn:

- The existing cloud security models are breached at some point of time by effective cryptanalysis, irrespective of its complex algorithmic design.
- There is need to find the more light and secure encryption system for file information preserving system on cloud.

Hence, a file security approach is needed on cloud that should be good balance between speed and security. The approach uses encryption algorithm. If the data stored in the cloud is in encrypted form, it would effectively solve various security issues.

3.2 Objectives of Thesis

The objectives of thesis are as follows:

- To propose hybrid cryptographic approach along with file splitting and merging mechanism.
- To install cloud environment using OpenNebula (open source cloud computing environment).

- To implement the proposed hybrid cryptographic approach on cloud to ensure file security.

At present ensuring security in cloud computing platform has become one of the most significant concerns for the researchers. In this thesis, we have considered these issues to provide some solution correlated with security. In the proposed model, high ranked security algorithms are used for giving secured communication process.

Chapter 4

OpenNebula

OpenNebula act as an effective open source toolkit, which can be easily adapted to work flawlessly even in heterogeneous environment, to build private, public and hybrid IaaS. The overall function of this system is to manage the provisioning of virtual machine for cloud computing infrastructure as a service. It provides an important alternative for those who do not wish to use a commercially provided cloud.

The OpenNebula provides infrastructure to users with an elastic platform for fast delivery and scalability of services to meet dynamic demands of end-users. It allows the user to dynamically host the services in VMs, enables monitoring and control using interfaces like command line interface, XML-RPC API, Libvirt virtualization API. OpenNebula manages the data center of private cloud and infrastructure of cluster running Xen, KVM or VMware and also support hybrid cloud to connect local and public infrastructure, which is very useful to build highly scalable cloud computing environment. OpenNebula supports heterogeneous execution environments with multiple, even conflicting, software requirements on the same shared infrastructure with full control of the lifecycle of virtualized services management.

The OpenNebula is differentiated from other commercial cloud solutions is that its true open source blood guarantees complete interoperability with every existing infrastructure component already available. Thus, it avoids the vendor lock-in using open industrial standards such as EC2 API and Open Cloud Computing Interface (OCCI). Unlike other open source alternatives, OpenNebula does not embrace a particular hypervisor. It also does not have any specific infrastructure requirements, hence appropriate for any pre-existing environment, storage, network, or user-management policies.

The plugin model, on which OpenNebula is implemented, gives system integrators the ability to customize every aspect including virtualization, storage, information, authentication, authorization and remote cloud services. Every action is managed by bash script that can be easily modified or plugged with some other custom script or software written in any language and supported by the operating system.

4.1 Architecture of OpenNebula

OpenNebula has been designed in such a way that it allows integration with many different hypervisors and environments. There is a front-end that executes all the process in OpenNebula while the cluster nodes provide the resources that are needed by VM. There is at least one physical network joining all the cluster nodes with the front-end.

The OpenNebula internal architecture can be divided into three layers. The components which are involved in OpenNebula and also the level at which they operate (lower ones interact directly with the host's resources, higher ones interact with user interfaces) in the three layers are shown in the following figure.

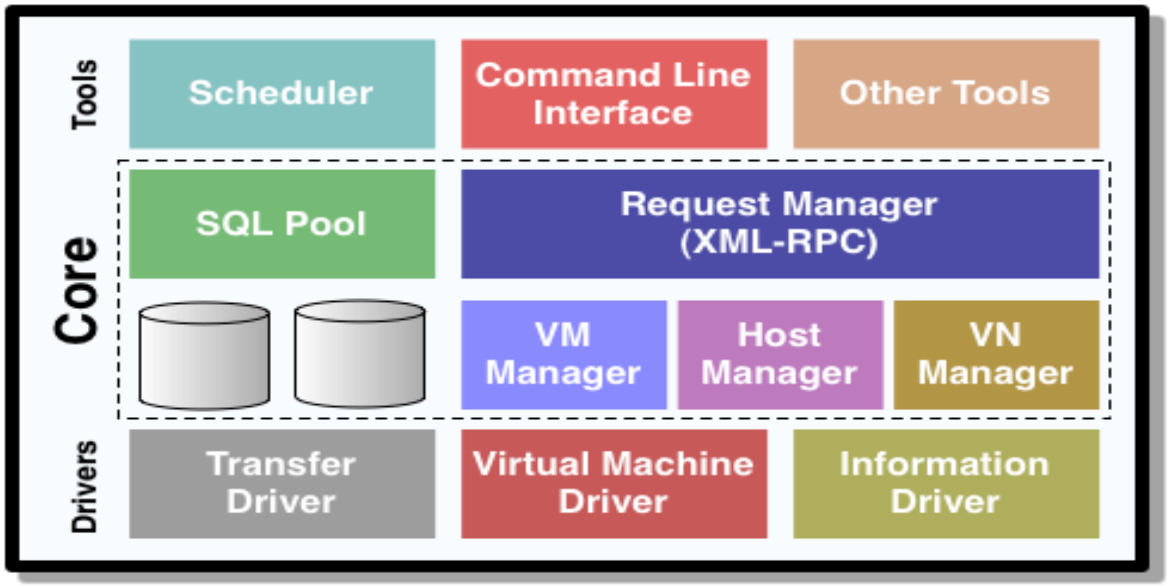

Figure 4.1: Architecture of OpenNebula [31]

4.1.1 Tools

This layer contains tools distributed with OpenNebula, such as the CLI, the scheduler, the libvirt API implementation and also 3rd party tools that can easily be created using the XML-RPC interface or the new OpenNebula Cloud API [31]. They are explained as below:

- **Command Line Interface:** A CLI for infrastructure administrators and users is provided with OpenNebula to manually manipulate the virtual infrastructure

- **Scheduler:** The Scheduler is an independent entity in the OpenNebula architecture, so it can be easily tailored or changed since it is decoupled from the rest of the components. It uses the XML-RPC interface provided by OpenNebula to invoke actions on virtual machines. The scheduler distributed with OpenNebula allows the definition of several resource and load aware policies.

4.1.2 OpenNebula Core

The core consists of a set of components to control and monitor virtual machines, virtual networks, storage and hosts. The core performs its actions (e.g. monitor a host, or cancel a VM) by invoking a suitable driver. The main functional components of OpenNebula core are explained as below:

- **Request Manager:** The Request Manager exposes a XML-RPC Interface, and then depending on the invoked method a given component is called internally. The XML-RPC decouples most of the functionality in the OpenNebula core, from external components i.e. the Scheduler.
- **Virtual Machine Manager:** This Virtual Machine Manager (VMM) is responsible for the management and monitoring of VMs. The operations of the VM Manager are abstracted from the underlying hypervisor with the use of pluggable drivers.
- **Transfer Manager:** The Transfer Manager (TM) is in charge of all the files transfers needed for the correct deployment of virtual machines. This includes the transfer of images to the cluster node selected for running the images of virtual machine, the transfer of the image from the cluster node to the image repository, the transfer of checkpoint files between cluster nodes for cold migrations or to the cluster front-end when the virtual machine is stopped, etc.
- **Virtual Network Manager:** The Virtual Network Manager (VNM) is responsible for the handling of IP and MAC addresses, allowing the creation of virtual networks by keeping track of leases (a set form by one IP and one MAC valid on a particular network) and their association with virtual machines and the physical bridges the VM are using.

- **Host Manager:** The Host Manager (HM) manages and monitors the physical hosts. Monitor and management actions are performed also through a suitable driver. The host monitoring infrastructure is flexible and can be extended to include any host attribute.
- **Database:** A persistent generic pool based on a SQLite3 backend is the core component of the OpenNebula internal data structures. This component provides OpenNebula with the scalability and reliability needed in the management of VMs.

4.1.3 Drivers

OpenNebula has a set of pluggable modules to interact with specific middleware (e.g. virtualization hypervisor, cloud services, file transfer mechanisms or information services), these adaptors are called Drivers [31]. They are explained as follows:

- **Transfer drivers**: These are used to manage the disk images on the current storage system—a shared one, such as Network File System (NFS) or on a non-shared one such as a simple copy over Secure Shell (SSH).
- **Virtual Machine drivers**: These are hypervisor-specific and are used for managing the virtual machine instances on the current hosts.
- **Information drivers**: These are used to retrieve the current status of virtual machine instances and hosts. They are hypervisor-specific, too—they are copied and remotely executed in every physical host through SSH.

4.2 Virtualization Providers for OpenNebula

The Virtualization subsystem is the component in charge of talking with the hypervisor installed in the hosts and taking the actions needing for each step in the VM lifecycle. The various hypervisors compatible with OpenNebula are as follows:

4.2.1 Xen

The first adopted OpenNebula hypervisor is Xen. It has been a unique leading open source virtualization technology for many years. Besides its use as a hypervisor in

OpenNebula, Xen is also used standalone by many internet hosting companies such as Amazon EC2, Linode, and Rackspace Cloud. It was originally distributed as a Linux patchset, but is nowadays included in main GNU/Linux distributions such as SuSe, RedHat, and Debian.

Xen is composed of the following three modules:

- **Hypervisor**: The core component responsible for scheduling and executing all the virtual machine instances currently running.
- **Dom0**: It is a privileged virtual machine running the base system and having direct hardware access. It is used to manage all the other deprivileged instances.
- **DomU**: An unprivileged virtual machine running on the hypervisor and having access only to virtual resources exposed by Dom0.

4.2.2 KVM

KVM(kernel –based virtual machine) is a complete virtualization technique for linx. It offers full virtualization, where each virtual machine interacts with its own virtualized hardware. KVM was being initially developed by a techie start-up, Quramnet, bought in 2008 by RedHat, and is now actively maintained by Linux developers all around the world.

In KVM design ,KVM by itself is only an interface available to user space programs that can be called through the /dev/kvm special system file. For similar reasons, another open source project has been ported to support the KVM interface in gaining a full virtualization environment, QEMU. QEMU make use of KVM when running a target architecture that is the same as the host architecture

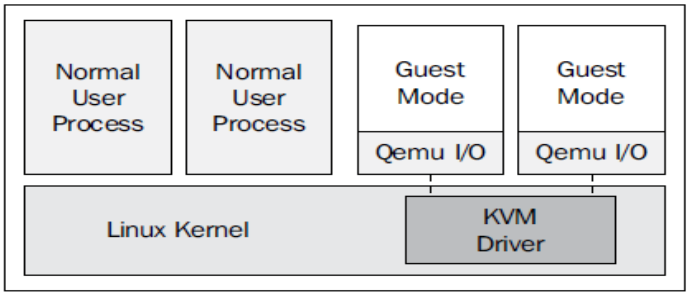

Figure 4.2: Architecture of KVM [31]

4.2.3 VMWare

The VMware Infrastructure API (VI API) provides a complete set of language-neutral interfaces to the VMware virtual infrastructure management framework. By targeting the VI API, the OpenNebula VMware drivers are able to manage various flavors of VMware hypervisors: ESXi (free), ESX and VMware Server.

- **VMware ESXi**: It is a non open source hypervisor, the simplest of the whole family, and natively includes only a command-line interface, and runs on its own kernel (not on Linux, as Xen/KVM do). Hence, the hardware support is pretty limited or highly optimized, depending on how you see it.
- **VMware ESX**: It was the mainline product before ESXi. It includes a Java web interface, and it is available under commercial license only.
- **VMware Server**: It is a free (not open source) hypervisor, available for installation on Linux and Windows; it includes a Java web interface such as ESX, but with fewer features.

The architecture of VMware is given below.

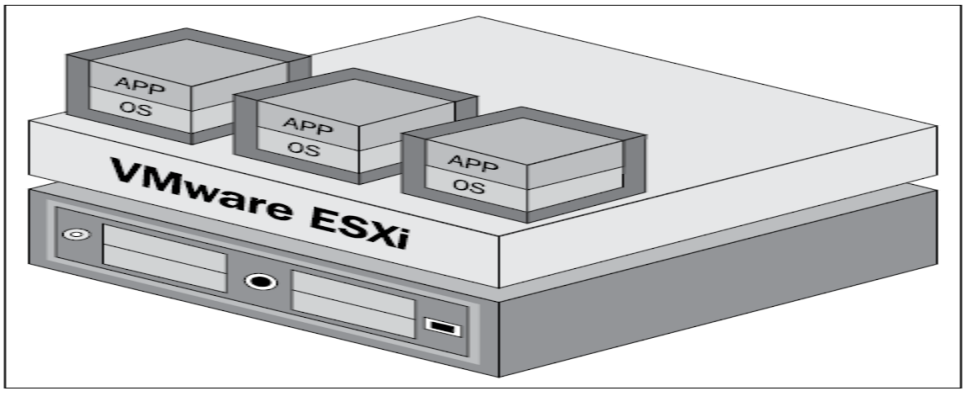

Figure 4.3: Architecture of VMware [31]

4.3 Advantages of OpenNebula

There are numerous reasons to choose OpenNebula over other opensource cloud environment.

Some of them are as follows:

- **Powerful and Innovative**: It bears most advanced and innovative enterprise class functionality for the management of virtualized data centers to build private, public and hybrid clouds.
- **Infrastructure Agnostic**: It is full platform independent with broad support for commodity and enterprise-grade hypervisor, storage and networking resources, allowing leveraging existing IT infrastructure, protecting investments and avoiding vendor lock-in.
- **Adaptable, Extensible and Integral**: It has Open, adaptable and extensible architecture, interfaces and components to build your customized cloud service or product
- **Interoperable**: OpenNebula is interoperable and portable providing cloud consumers with choice across standards and most popular cloud interfaces
- **Fully Open Source**: OpenNebula is not a feature or performance limited edition of an Enterprise version, OpenNebula is truly open-source code, not open core, distributed under Apache license
- **Very Light Solution**: Despite its technical sophistication and advanced functionality, OpenNebula is easy to download, install and update
- **Stable and Proven**: It is rigorously tested through an internal quality assurance process and by a large community with scalability, reliability and performance tested on many massive scalable production deployments
- **Enterprise-class Product**: OpenNebula comprises all key functionalities for enterprise cloud computing, storage and networking in a single install, and ensures it long term stability and performance through a single integrated patching and updating process
- **One-stop Support**: It bears wide variety of community and commercial support from the developers of OpenNebula.

4.4 Compatibility with other Open Source Environment

OpenNebula is highly compatible with OpenStack, Eucalyptus or vCloud.

- The **hybrid cloud capabilities** of OpenNebula enable to implement cloud bursting architectures where an OpenNebula private cloud bursts to a vCloud, OpenStack- or Eucalyptus-based cloud when the demand for computing capacity spikes.
- The **integration capabilities** of OpenNebula allow its integration with OpenStack Swift or OpenStack Quantum for object/block store and networking management respectively in the data center.
- OpenNebula can work as a **data center virtualization manager** within an OpenStack or Eucalyptus cloud.

4.5 Comparative Analysis with other Open Source Cloud Solutions

The comparative analysis of OpenNebula with OpenStack , Eucalyptus, Nimbus is given in following table. ✓ in table indicates positive evaluation.

Table 4.1: Feature Comparison of IaaS Framework [32]

	OpenStack	Eucalyptus 2.0	Nimbus	OpenNebula
Interfaces	EC2 and S3, Rest Interface. Working on OCCI ✓✓	EC2 and S3, Rest Interface. Working on OCCI ✓✓	EC2 and S3, Rest Interface ✓	Native XML/RPC, EC2 and S3, OCCI, Rest Interface ✓✓✓
Hypervisor	KVM, XEN, VMware Vsphere, LXC, UML and MS HyperV ✓✓✓	KVM and XEN. VMWare in the enterprise edition. ✓✓	KVM and XEN ✓	KVM, XEN and VMWare ✓✓
Networking	- Two modes: (a) Flat networking (b) VLAN networking -Creates Bridges automatically -Uses IP forwarding for public IP -VMs only have private IPs ✓✓✓	- Four modes: (a) managed; (b) managed-novLAN; (c) system; and (d) static - In (a) & (b) bridges are created automatically - IP forwarding for public IP -VMs only have private IPs ✓✓✓	- IP assigned using a DHCP server that can be configured in two ways. - Bridges must exists in the compute nodes ✓✓	- Networks can be defined to support Ebtable, Open vSwitch and 802.1Q tagging -Bridges must exists in the compute nodes -IP are setup inside VM ✓✓✓
Software deployment	- Software is composed by component that can be placed in different machines. - Compute nodes need to install OpenStack software ✓	- Software is composed by component that can be placed in different machines. - Compute nodes need to install OpenStack software ✓	Software is installed in frontend and compute nodes ✓✓	Software is installed in frontend ✓✓✓
DevOps deployment	Chef, Crowbar, Puppet ✓✓✓	Chef*, Puppet* ✓ (*according to vendor)	no	Chef, Puppet ✓✓
Storage (Image Transference)	- Swift (http/s) - Unix filesystem (ssh) ✓	Walrus (http/s) ✓	Cumulus (http/https) ✓	Unix Filesystem (ssh, shared filesystem or LVM with CoW) ✓
Authentication	X509 credentials, LDAP ✓✓✓	X509 credentials ✓	X509 credentials, Grids ✓✓	X509 credential, ssh rsa keypair, password, LDAP ✓✓✓

4.6 Components of OpenNebula

OpenNebula assumes that your physical infrastructure adopts a classical cluster-like architecture with a front-end, and a set of cluster nodes where Virtual Machines will be executed. There is at least one physical network joining all the cluster nodes with the front-end.

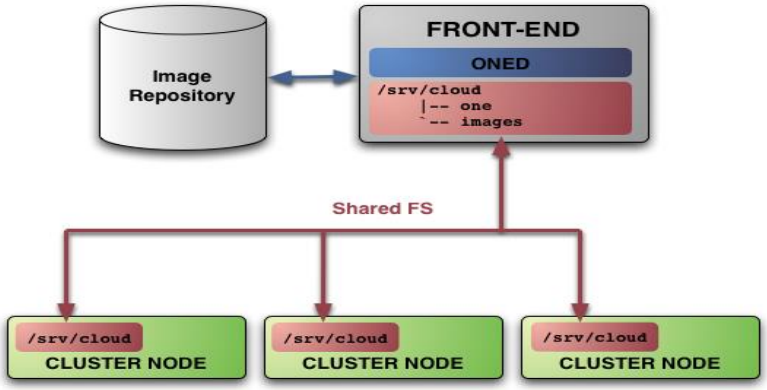

Figure 4.4: Components of OpenNebula [33]

The basic components of an OpenNebula system are:

- **Front-end:** The machine that holds the OpenNebula installation is called the front-end. This machine needs to have access to the storage medium and network connectivity to each host (node). It executes the OpenNebula and cluster services. Additionally, one should be able to ssh passwordlessly to all the hosts (including itself, the front end) from the front end.
- **Nodes:** The nodes are the hypervisor-enabled hosts which run VM's and provide the resources needed by the virtual machines. OpenNebula doesn't need to install any packages in the hosts, and the only requirements for them are ssh server running and properly configured working hypervisor. Additionally, one should be able to ssh passwordlessly to all the hosts (including itself and the front end) from each host.
- **Image repository:** It is the storage medium that holds the base images of the VMs. VM images are registered, or created (empty volumes) in a image repository (datastore). When a VM is deployed the images are transferred from

the datastore to the hosts. Each datastore has to be accessible through the front-end using any suitable technology NAS, SAN or direct attached storage.

- **OpenNebula daemon:** OpenNebula daemon (oned) is the core service of the system. It manages the life-cycle of the VMs and orchestrates the cluster subsystems (network, storage and hypervisors).
- **Drivers:** These are the programs used by the core to interface with a specific cluster subsystem, e.g. a given hypervisor or storage file system.

4.7 Planning and Installation of OpenNebula

Following are the different steps required for planning and installation of OpenNebula:

4.7.1 System Requirements

i) OneHost: [Server/Frontend]

The front end should satisfy the following system specifications:

- VT-x enabled server hardware [Hostname : OneHost]
- Atleast 100 GB HDD free space
- Atleast 2 GB RAM.

ii) VMHost: [Cluster node]

The cluster node should satisfy the following system specifications:

- VT-x enabled server hardware [Hostname : VMHost]
- Atleast 100 GB HDD free space,
- Atleast 2 GB RAM

{Note: You can also have more than one VM Host}

4.7.2 Preparing OneHost

i) Install Ubuntu Server 12.10 64 bit software in OneHost

ii) Have GRUB installed.

iii) Install OpenSSH server alone.

iv) Once, the installation is complete, configure OneHost to act as a router by enabling packet forwarding. Edit /etc/sysctl.conf and uncomment the below line.

net.ipv4.ip_forward = 1

v) Edit /etc/rc.local and add the following lines just above the "exit 0" line and restart OneHost machine.

iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

iptables -t nat -I POSTROUTING -s 172.31.4.149/24 -d 172.31.4.150/8 -j MASQUERADE

vi) Reboot the system using the following command:

sudo reboot

4.7.3 Preparing VMHost

i) Install Ubuntu Desktop 12.04 32 bit.

ii) Install OpenSSH client.

iii) Once installation is over, login to VMHost using server console/SSH. One should be able to ping OneHost and connect to internet.

iv) Install bridge-utils using below command.

sudo apt-get install bridge-utils

v) Edit /etc/network/interfaces to add the below lines to add bridge configuration.

```
auto br0
iface br0 inet dhcp
bridge_ports eth0
bridge_fd 9
bridge_hello 2
bridge_maxage 12
bridge_stp off
```

vi) Restart the networking using the following command.

sudo /etc/init.d/networking restart

4.7.4 Configuring OneHost

i) Create a folder “cloud” and create a group named “cloud”.

sudo mkdir -p /srv/cloud/

sudo groupadd -g 10000 cloud

ii) Create a user “oneadmin”, add user to group “cloud” and have /srv/cloud/one as home folder.

sudo useradd -u 10000 -m oneadmin -d /srv/cloud/one -s /bin/bash -g cloud.

iii) Set up the password for "oneadmin" and make oneadmin owner of "/srv/cloud".

sudo passwd oneadmin

sudo chown -R oneadmin:cloud /srv/cloud/

iv) Test by logging as user "oneadmin" and exit

su -l oneadmin

exit

v) Install Network file Server [NFS]

sudo apt-get install nfs-kernel-server

vi) Edit /etc/exports and add the following line to make folder /srv/cloud/one shareable with VMHost

/srv/cloud/one 172.31.4.149/24 (rw, fsid=0, nohide, sync, root_squash, no_subtree_check)

vii) Restart NFS server

sudo /etc/init.d/nfs-kernel-server start

viii) Add VMHost's Hostname to /etc/hosts of OneHost

sudo nano /etc/hosts [add below line]

172.31.4.149 VMHost

ix) Create a SSH key for oneadmin and disable host key checking else make all host keys known on the OpenNebula node.

su -l oneadmin

ssh-keygen

ssh-copy-id -i ~/.ssh/id_rsa.pub <remote-host>

[add below two lines to SSH config file]

Host *

StrictHostKeyChecking no

{Note: ssh-copy-id appends the keys to the remote-host's .ssh / authorized _ key}

x) Exit from editor and try pinging VMHost, it should ping.

4.7.5 Configuring VMHost

Before configuring VMHost, check if you are able to ping OneHost IP (172.31.4.150) and OneHost's Gateway (172.31.4.1) from VM Host.

i) Add oneadmin to sudo group

usermod -a -G sudo oneadmin

ii) Install "NFS common" to enable access to the folder "/srv/cloud/one" of OneHost.

sudo apt-get update

sudo apt-get install nfs-common

iii) Edit /etc/fstab and add an NFS entry for /srv/cloud/one

172.31.4.150:/srv/cloud/one /srv/cloud/one nfs defaults 0 0

iv) Create folder structure /srv/cloud/one in VMHost and mount it as per "fstab" entry.

sudo mkdir -p /srv/cloud/one

sudo mount /srv/cloud/one

v) Create user "oneadmin" and group "cloud".

sudo groupadd -g 10000 cloud

sudo useradd -u 10000 -g cloud -m oneadmin -s /bin/bash

sudo usermod -d /srv/cloud/one oneadmin

sudo usermod -a -G cloud,root oneadmin

sudo passwd oneadmin

sudo chown oneadmin:cloud /srv/cloud

sudo mount /srv/cloud/one

mount

Note: You should see the below line

172.31.4.150:/srv/cloud/one on /srv/cloud/one type nfs (rw, addr=172.31.4.150)

vi) Install KVM hypervisor using the following command.

sudo apt-get install qemu-kvm libvirt-bin ubuntu-vm-builder bridge-utils ruby

vii) Libvirt needs to be configured to enable users of group "cloud" to manage the VM's and to allow VNC connections. Edit "/etc/libvirt/libvirtd.conf" and make the following changes:

unix_sock_group = "cloud"

(Search for string "unix_sock", if commented, uncomment this line and change the existing value to "cloud").

viii) Edit /etc/libvirt/qemu.conf and uncomment vnc_listen line and restart libvirt

vnc_listen = "0.0.0.0"

sudo service libvirt-bin restart

ix) Configure libvirt to allow access from the members of group "cloud"

sudo chown :cloud /var/run/libvirt/libvirt-sock

4.7.6 Install and Configure OpenNebula in OneHost

i) Before installing OpenNebula, install all pre-requisite packages

sudo apt-get install libsqlite3-dev libxmlrpc-c3-dev g++ ruby libopenssl-ruby libssl- dev ruby-dev

sudo apt-get install libxml2-dev libmysqlclient-dev libmysql++-dev libsqlite3-ruby libexpat1-dev

sudo apt-get install rake rubygems libxml-parser-ruby1.8 libxslt1-dev genisoimage scons

sudo gem install nokogiri rake xmlparser

sudo apt-get install mysql-server

ii) Configure MySql

mysql -uroot –preema {Note: Here 'root' is the user and 'reema' is the password}

Create user 'oneadmin'@'localhost' IDENTIFIED BY 'oneadmin';

Create Database opennebula;

Grant all priviliges on opennebula.* to ‘oneadmin’ identified by ‘oneadmin’;

quit;

iii) Before installing OpenNebula, configure mysql support.

cd ~/opennebula-3.1.90 [change your folder to opennebula source]

scons sqlite=no mysql=yes

iv) Install openebula in /srv/cloud/one accessible by group cloud and as user “oneadmin”.

./install.sh -u oneadmin -g cloud -d /srv/cloud/one

v) Create a profile file (~/.bash_profile) to set Environment Variables required to start and use services rendered by "one".

nano ~/.bash_profile

export ONE_LOCATION=/srv/cloud/one

export ONE_AUTH=$ONE_LOCATION/.one/one_auth

export ONE_XMLRPC=http://localhost:2633/RPC2

export PATH = $ONE_LOCATION/bin:/usr/local/bin: /var/lib/gems/1.8/bin /: /var /lib /gems/1.8/: $PATH

vi) Execute the profile file and set the environment variables.

source ~/.bash_profile

vii) Create and store OpenNebula user (oneadmin) and password in a file.

mkdir ~/.one

echo "oneadmin:<Password>" > ~/.one/one_auth

viii) Make required changes in OpenNebula configuration file (~/etc/oned.conf).

nano ~/etc/oned.conf

- Comment following line

  #DB = [backend = "sqlite"]

- Set SQL as mysql-uncomment #lines 61 through 66 or near by

  DB = [backend = "mysql",

  server = "localhost",

  port = 0,

  user = "oneadmin",

passwd = "oneadmin",

db_name = "opennebula"]

- Add below lines just below first TM_MAD definition.

  TM_MAD = [

  name = “tm_nfs”,

  executable = “one_tm”,

  arguments = “tm_shared/tm_shared.conf”]

ix) Start Nebula

one start { Note: it should start with no error messages}

4.8 Administering OpenNebula

i) Adding a host

Checkpoints:

- Check /etc/hosts file of OneHost and VMHost for Hostname entries.
- Try ssh VMHost from $ prompt and you should be able to login with no password.

Command: onehost create <hostname> --im <im_mad> --vm <vmm_mad> --net dummy

where,

im_mad specifies “Information drivers” -used to monitor the host.

vmm_mad specifies “Virtualization Drivers” -used to manage VMs.

e.g. onehost create omg --im im_kvm --vm vmm_kvm --net dummy

ii) Listing Registered host(s)

Command: onehost list

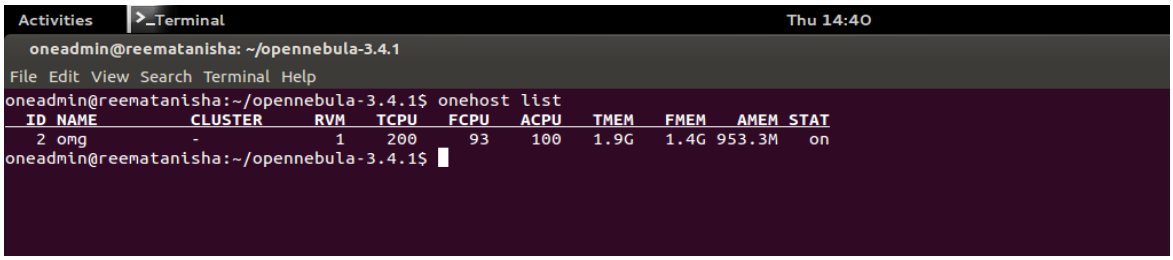

Figure 4.5: Registered Hosts

iii) Detailed information about the registered host

Command: onehost show <host ID> /<host_Name>

e.g onehost show 2

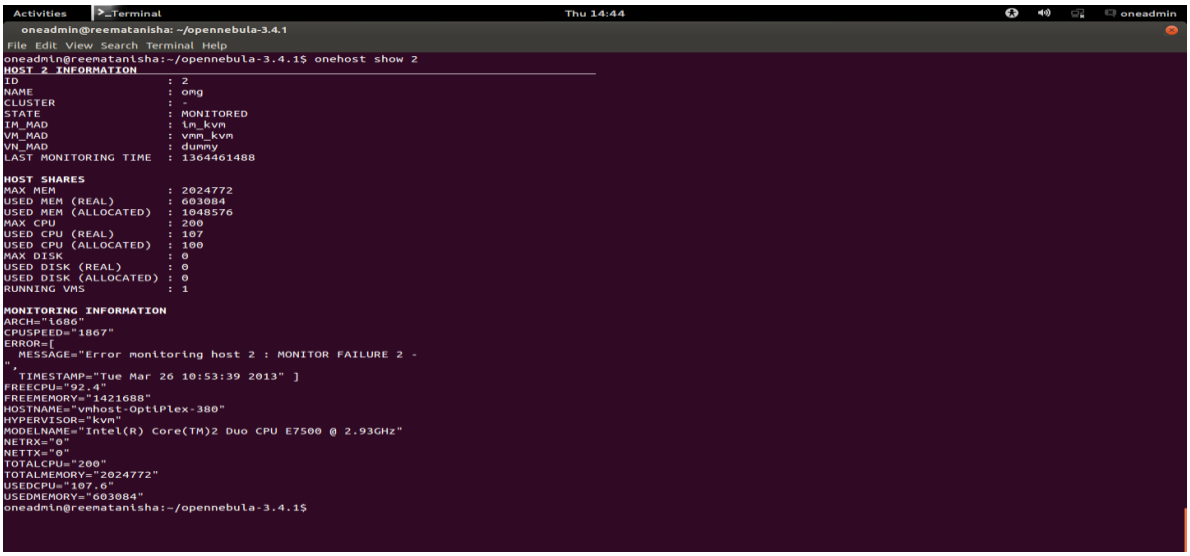

Figure 4.6 : Detailed Information of Host

4.9 Creating a Virtual Machine

i) Configuring a FileSystem Datastore.

The first step to create a filesystem datastore is to set up a template file (ds.conf) for it. The datastore type is set by its drivers, in this case be sure to add DS_MAD=fs. The other important attribute to configure the datastore is the transfer drivers. These drivers determine how the images are accessed in the hosts. Various attributes of configuration file are explained as below:

Table 4.2: Attributes of Datastore Configuration File[33]

Attribute	Description
NAME	The name of the datastore
DS_MAD	The DS type, use fs for the Filesystem datastore
TM_MAD	Transfer drivers for the datastore: shared, ssh or qcow2
SAFE_DIRS	If you need to un-block a directory under one of the RESTRICTED_DIRS. A space separated list of paths.

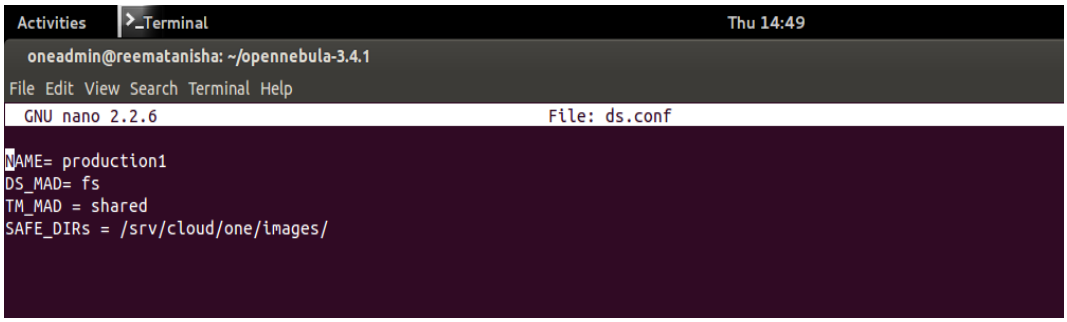

Figure 4.7: Datastore Configuration File

ii) Creating a datastore

Command: onedatastore create ds.conf

where,

ds.conf – Datastore configuration file

{Note: On successful execution of command, datastore will be created and an ID will be assigned }

iii) Listing available datastores.

Command: onedatastore list

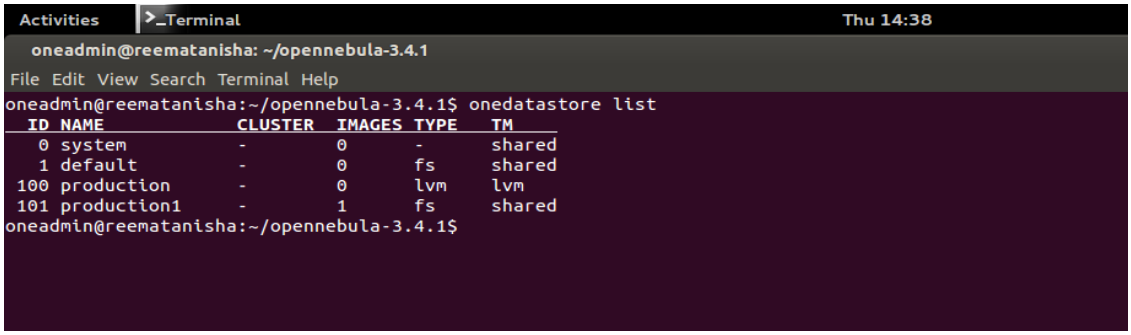

Figure 4.8: List of Available Datastores

iv) Detailed information about the registered datastores

onedatastore show <Datastore ID>

e.g onedatastore show 101

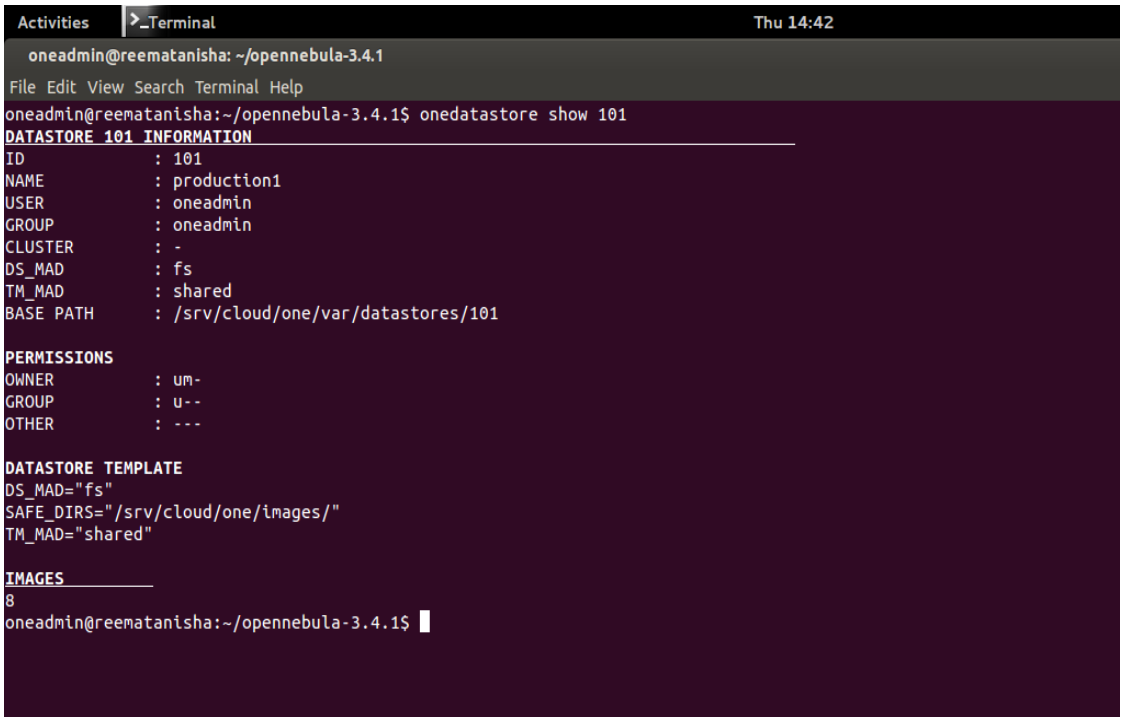

Figure 4.9: Detailed Information of Datastore

v) Configuring an image definition file

An OpenNebula image is a resource containing an operative system or data, to be used as a virtual machine disk. The first step is to define an image template file (vmImage.img). Various attributes of the template are explained as below:

Table 4.3 : Attributes of Image Template [33]

Attribute	Description
NAME	Name that the Image will get. Every image must have a unique name.
TYPE	Type of the image i.e. OS,CDBLOCK,DATABLOCK
PATH	Path to the original file that will be copied to the image repository.
PUBLIC	Public scope of the image. If omitted, the default value is NO.

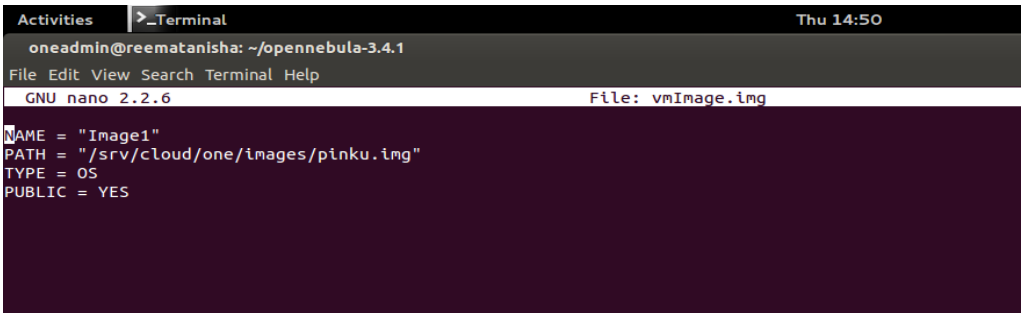

Figure 4.10: Image Definition File

vi) Submitting an image

After creating the image template, submit it using the oneimage create command.

e.g. oneimage create vmImage.img -d 101

where,

vmImage.img – Image template file

101 – Datastore ID

{Note: one successful execution of command, image will be created and an image ID will be generated}

vii) Listing of available image(s)

To check the available images in the repository oneimage list command is issued.

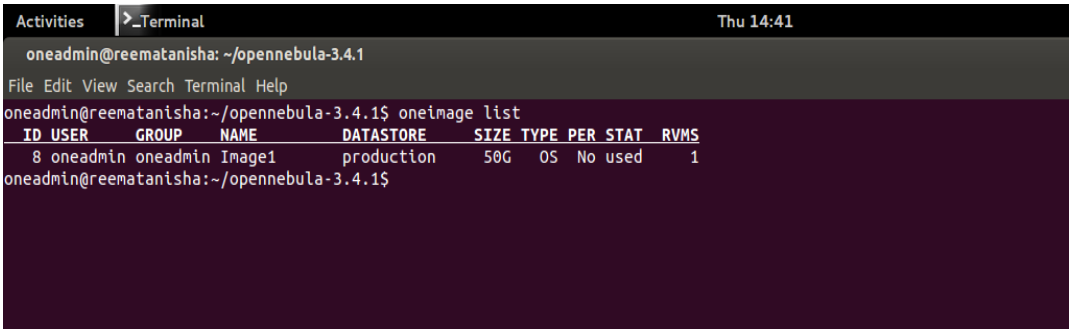

Figure 4.11: Images Available in the Repository

viii) Detailed information about the registered images

Command: oneimage show <Image ID>

e.g oneimage show 8

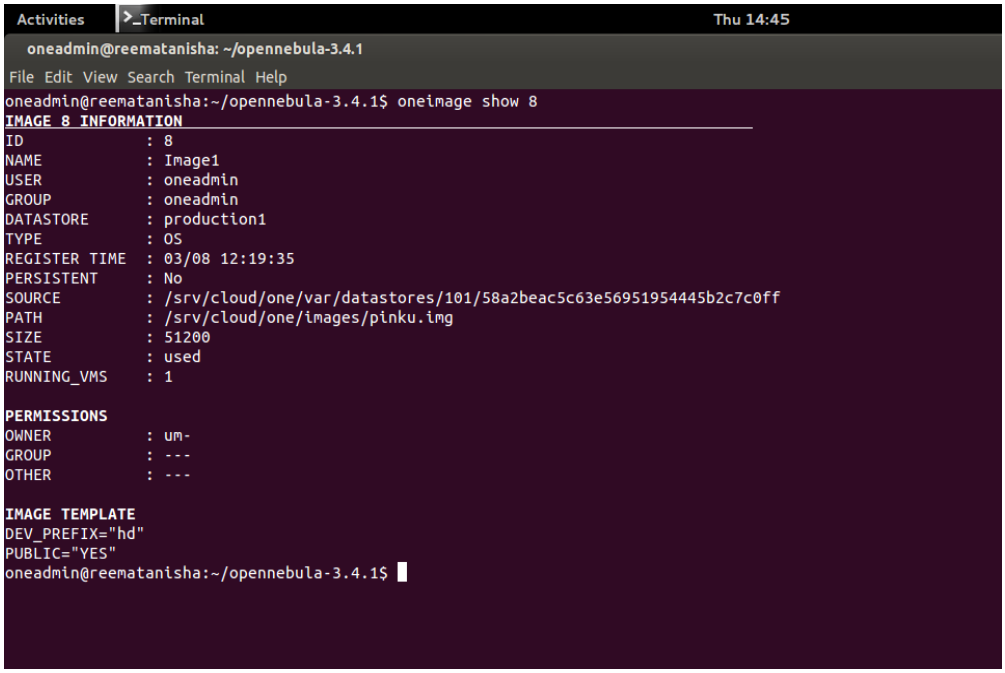

Figure 4.12: Detailed Information of an Image

ix) Configuring a Virtual Network

OpenNebula allows the creation of virtual networks by mapping them on top of the physical ones. All Virtual Networks are going to share a default value for the MAC prefix, set in the oned.conf file.

There are two types of Virtual Networks in OpenNebula:

- Fixed: This defines a fixed set of IP-MAC pair addresses
- Ranged: This defines a class network.

Various attributes of network template are explained as below:

Table 4.4 : Attributes of Network Template File[33]

Attribute	Description
NAME	Name of the Virtual Network.
TYPE	Type of the virtual network
PUBLIC	Public scope of the image. If omitted, the default value is NO
BRIDGE	Name of the physical bridge
NETWORK_ADDRESS	Base network address to generate IP addresses.

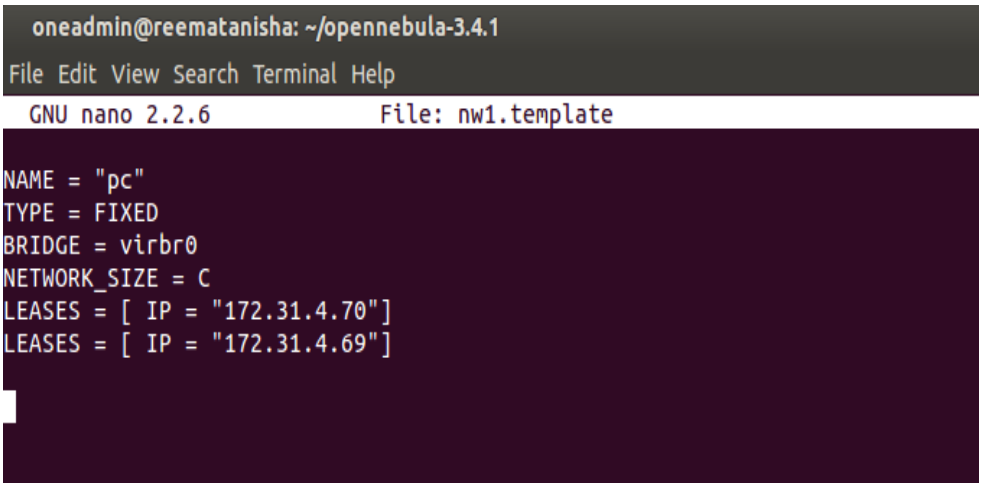

Figure 4.13: Network Template

x) Adding Virtual Networks (VN)

Once a template for a VN has been defined, the onevnet command can be used to create a virtual network.

onevnet create nw1.template

where,

nw1.template - network template file

xi) Listing of available Virtual Networks

Command: onevnet list

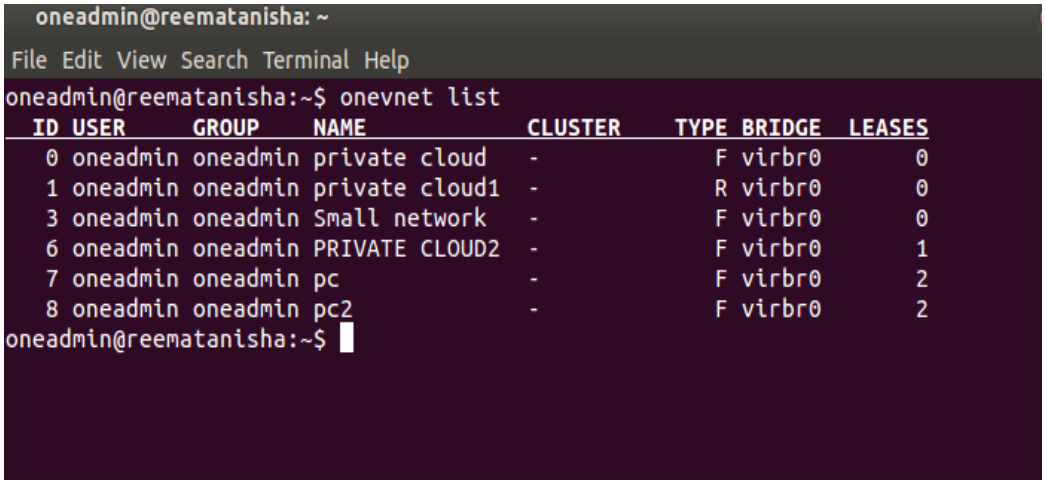

Figure 4.14: List of Virtual Networks

xii) Detailed information about a specific virtual network

Command: onevnet show <Network ID>

e.g. onevnet show 7

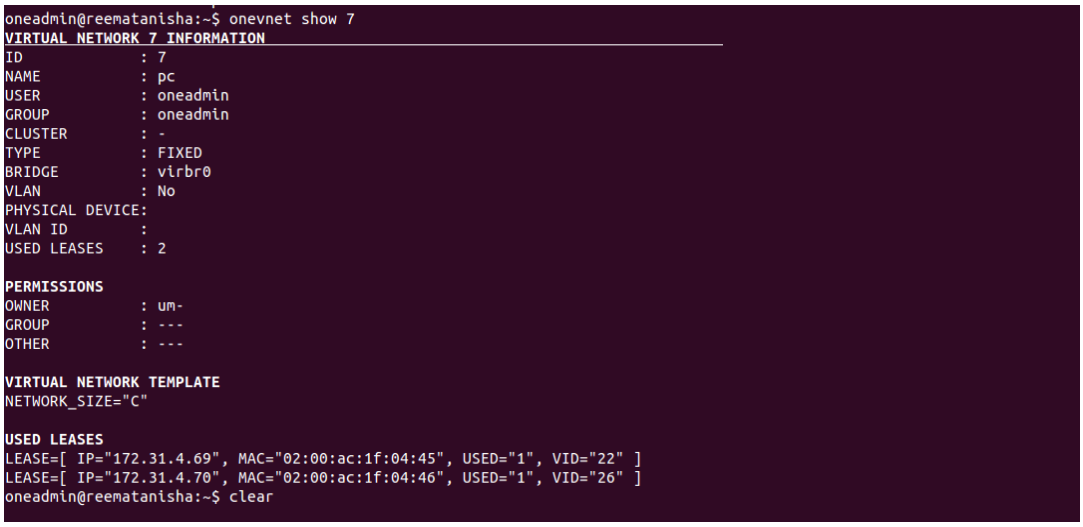

Figure 4.15: Detailed Information of a Network

xiii) Configuring a VM template file

The foremost step prior to deploying a VM is to define the template file (pc1.one). A template file consists of a set of attributes that defines a Virtual Machine. Various attributes of template file are explained as below:

Table 4.5: Attributes of VM Template [33]

Attribute	Definition
NAME	Name that the VM will get for description purposes
CPU	Percentage of CPU divided by 100 required for the Virtual Machine
DISK	
NIC	Defines the network interface of a VM
GRAPHICS	Whether the VM should export its graphical display and how

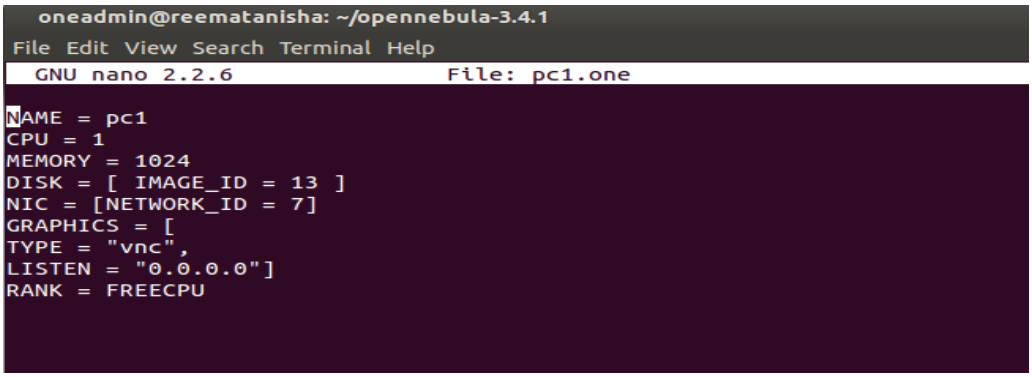

Figure 4.16: VM Template

xiv) Deploying a VM

onevm create pc1.one

where,

pc1.one – VM template file

xv) Listing deployed VM(s)

Command: onevm list

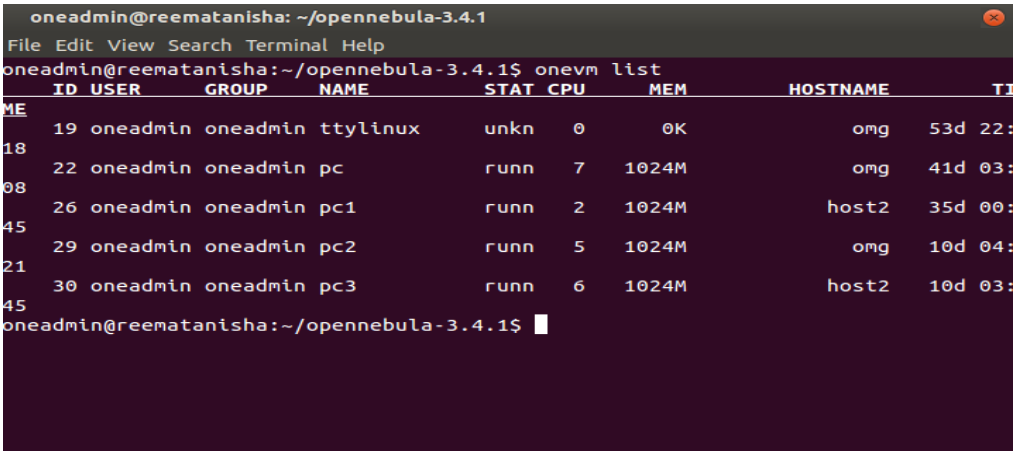

Figure 4.17: Registered VM's

xvi) Detailed information about VM

onevm show <VM ID>

e.g. onevm show 26

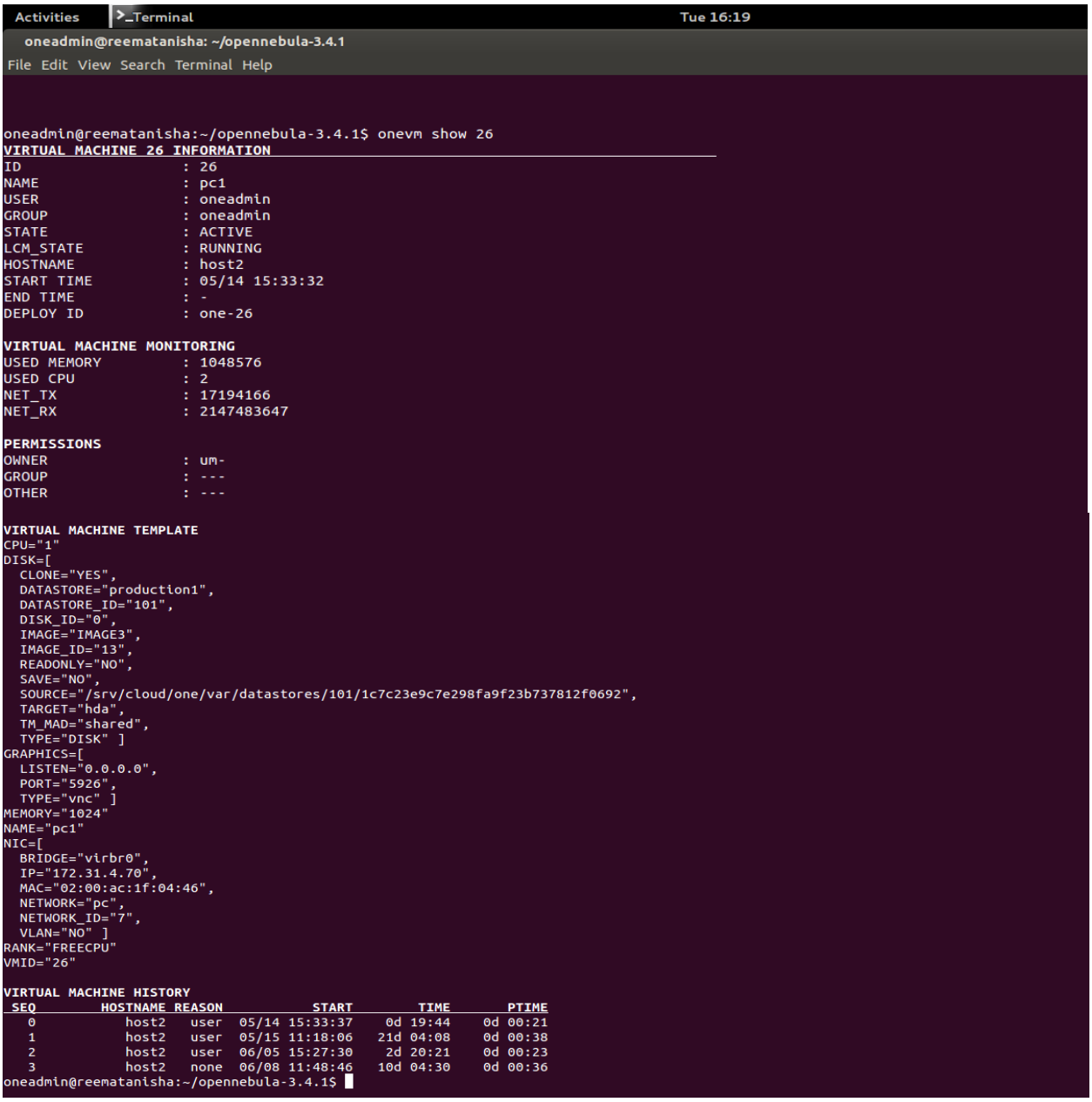

Figure 4.18: Detailed Information of VM ID 26

Chapter 5

Proposed Work

In order to ensure file security on cloud, we have designed a hybrid cryptosystem. We assume that the remote server is trusted, so files are encrypted by server and finally encrypted files are stored at the server end. The proposed hybrid cryptosystem uses a combination of:

- Blowfish Algorithm coupled with file splitting and merging mechanism
- SRNN Algorithm

The performance of symmetric algorithm is integrated with security of asymmetric algorithm. The algorithms used in proposed hybrid cryptosystem are explained below.

5.1 Encryption Algorithms Used

5.1.1 Blowfish Algorithm

Blowfish was designed [34] in 1993 by Bruce Schneider as a fast alternative to existing encryption algorithms. Blowfish is a symmetric key block cipher that uses a 64 bit block size and variable key length. It takes a variable-length key from 32 bits to 448 bits. Blowfish is one of the

fastest block ciphers which has developed up to date. It uses 16 rounds of iterative encryption and decryption functional design.

Blowfish symmetric block cipher algorithm encrypts block data of 64-bits at a time.it will follows the feistel network and this algorithm is divided into two parts.

- Key-expansion
- Data Encryption

5.1.1.1 Key-Expansion

It will convert a key of at most 448 bits into several sub-key arrays totaling 4168 bytes. Blowfish uses large number of sub-keys. These keys are generated earlier to any data encryption or decryption.

The p-array consists of 18, 32-bit sub-keys:

P1, P2,……,P18

Four 32-bit S-Boxes consists of 256 entries

S1,0, S1,1,……. S1,255

S2,0, S2,1,…….. S2,255

S3,0, S3,1,…….. S3,255

S4,0, S4,1,.....S4,255

5.1.1.2 Data Encryption Process

Blowfish cipher uses 18 each of 32-bit sub arrays commonly known as P-boxes and four Substitution boxes each of 32 bit size and having 256 entries each. It uses a Feistel cipher which is a general method of transforming a function into another function by using the concept of permutation. The working of blowfish cipher [35] can be illustrated as follows

- It splits the 64 bit block into two equal blocks having 32 bit size each.
- Left block is XORed with first sub array P1 and thus obtained result is fed in to a function called F-function .Inside the F-function substitution operations are carried out which in turn converts 32 bit blocks in to another 32 bit blocks.
- Thus resulted 32-bit entries are XORed with the right half and the result obtained is swapped as the left half for the next round.
- After the successful completion of each round, right half becomes the new left half or vice versa and Fiestal structure is followed up to 16 rounds
- The resultant left and right halves are not swapped but XORed with the seventeenth and eighteenth P-arrays.

The Fiesal structure of blowfish algorithm is shown in the following figure:

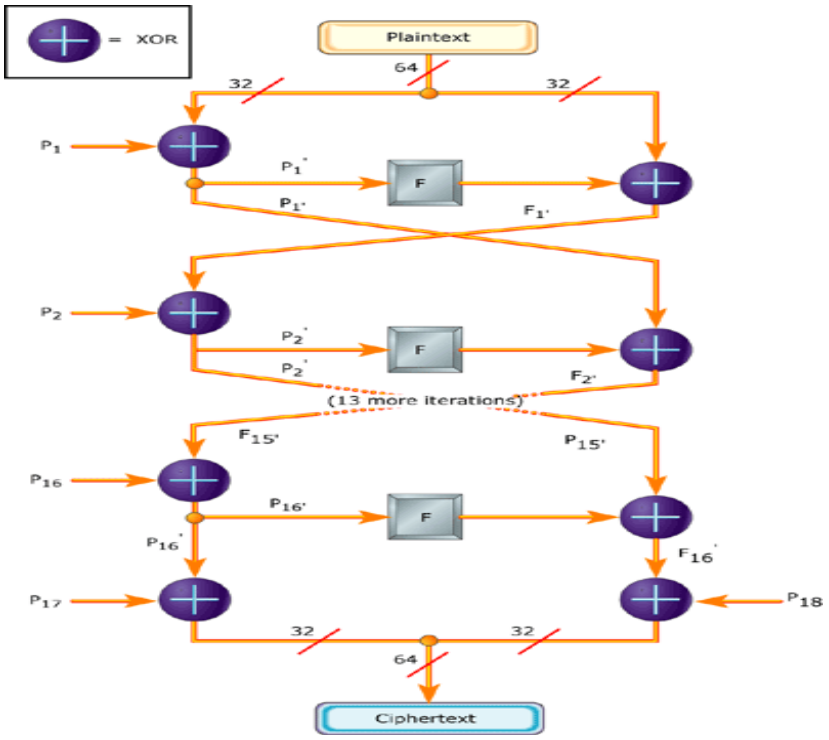

Figure 5.1: Encryption Phase of BlowFish [35]

Decryption is exactly the same as encryption, except that P1, P2,..., P18 are used in the reverse order. This is because XOR is commutative and associative.

5.1.2 RSA Algorithm

The Rivest-Shamir-Adleman (RSA) [15] cryptosystem is one of the best known public key cryptosystems for key exchange or digital signatures or encryption of blocks of data. RSA uses a variable size encryption block and a variable size key.

The key-pair is derived from a very large number, n, that is the product of two prime numbers chosen according to special rules; these primes may be 100 or more digits in length each, yielding an n with roughly twice as many digits as the prime factors.

For encryption, we use

$$C = M^e \bmod(n)$$

For decryption, we use

$$M = C^d \bmod(n)$$

Where C is the cipher, M is the message, e is the public key, *d* is the private key, and n is the modulus.

5.1.3 Short Range Natural Number (SRNN)

The SRNN (Short Range Natural Number) algorithm [15] is similar with RSA with some modification. SRNN algorithm is also a public key cryptography algorithm. In this algorithm we have extremely large number that has two prime factors (similar to RSA) .In addition of this we have used two short range natural numbers in pair of keys. This modification increases the security of the cryptosystem. So its name is short range natural number public key algorithm.

5.1.3.1 Key Generation Process

The key generation process [15] of SRNN is as follows:

- Generate two large random primes, p and q, of approximately equal size such that their product n = p x q is of he required bit length, e.g. 1024 bits.
- Compute n= p x q.
- Compute phi = (p-1) (q-1).
- Choose an integer e, 1 < e < phi, such that gcd (e, phi) = 1. Compute the secret exponent d, 1<d<phi, such that (e x d) mod phi = 1.
- Pick short range natural number u randomly such that u < phi-1.
- Pick another short range natural number a randomly such that phi > a > u. And compute u^a
- Find d such that; e * d mod ((p-1)*(q-1)) = 1
- The public key is (n, e, u^a) and the private key is (d, a, u). The values of p, q, and phi should also be kept secret.

5.1.3.2 Encryption Process

In encryption process, recipient does the following.

- Obtain the recipient public key (n, e, u^a) .
- Represent the plaintext as positive number m.

- Compute the cipher text c =$(mu^a)^e$ mod n.
- Send the cipher text to recipient.

5.1.3.3 Decryption Process

In decryption process, recipient does the following,

- Using private key (d, a, u)

  $$m = (v^e c)^d \bmod (n)$$

  Where v= u^{phi-a} mod (n), m denotes the message, n denotes the modulus, c denotes the cipher text.
- Extracts the plaintext from the integer representative m.

5.1.4 File Splitting and Merging Mechanism

File is sliced in to number of slices [29], so that each slice is encrypted with different key. The slices are encrypted with different key. This approach involves Cryptographic enciphering and deciphering along with file splitting and merging mechanisms. In this approach, a file which has secret data is sliced into desired number of pieces upon user's specification and then the cryptographic encryption phase is carried out. In order to achieve more security, we can adopt more than one cryptographic scheme which definitely ensures nil suspicion and more security. In this approach, we differentiate the cryptographic scheme by providing different key for each encryption of sliced files; provided the key should be given correctly at the time of decryption to avoid erroneous results.

In the file joining phase, en-ciphered files thus obtained from the different en-ciphering techniques are merged and hence transmitted to reception side as a single file which makes the file infeasible to breach and suspicion less to get to know that varying crypto schemes are adopted and hence, the data security is maximized. At the receiver's end, again files are sliced and decrypt it using the same algorithm and then join all split files together to retrieve the original message.

5.2 Hybrid Cryptosystem

The proposed hybrid cryptosystem, which is used to maintain security of the files has two phases:

- Encryption Phase
- Decryption Phase

5.2.1 Encryption Phase

- On the specification of the user, the file being encrypted will be sliced in to n slices, according to the number defined.
- Then the user will enter key (Blowfish Key) for each slice.
- The key will be encrypted by SRNN public key and corresponding SRNN private key will be generated.
- The Blowfish key, given by the user, will be used to encrypt the corresponding slices. After encryption, we have encrypted files and encrypted keys.

The encryption phase is illustrated in the following figure:

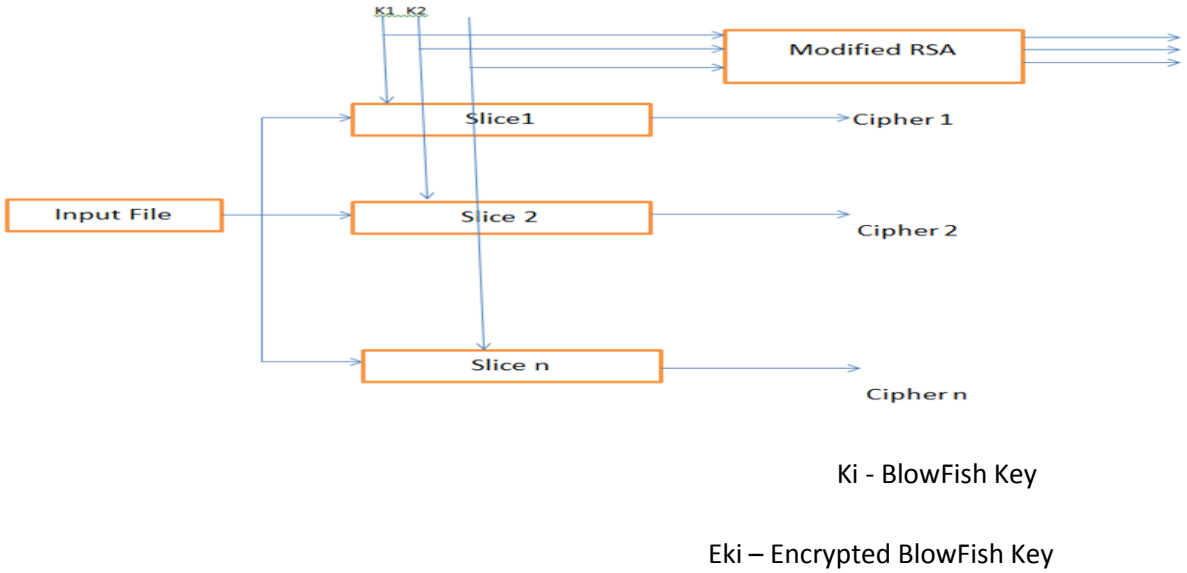

Figure 5.2: Encryption Phase

5.2.1.1 File Splitting and Merging Mechanism

The files are sliced at encryption phase and merged at decryption phase.

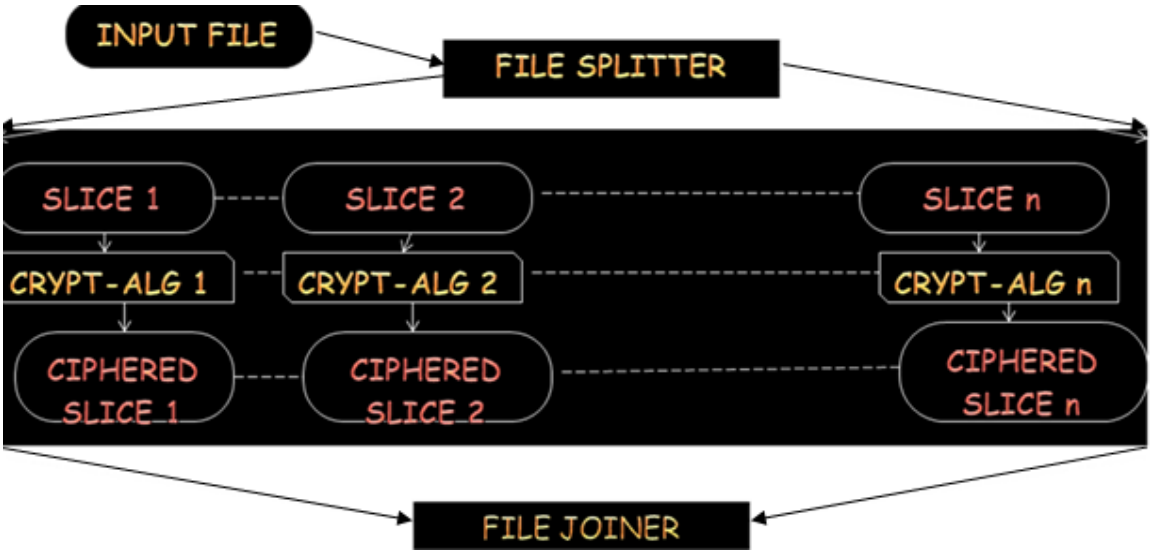

Figure 5.3: File Splitting and Merging Mechanism [29]

During encryption, the original file is sliced and slices are encrypted with different key corresponding to the slice. During decryption, the files are decrypted with key corresponding to the slice and finally the sliced are merged to the final original file.

5.2.2 Decryption Phase

At the decryption end,

- According to the no. of the slices determined during the encryption phase , the user will provide SRNN private keys. Using the corresponding SRNN private key, Blowfish key is decrypted at the server end.
- Using the corresponding decrypted blowfish key, encrypted file slices are decrypted. The corresponding key should be same that was given at the time of encryption.
- The decrypted slice will be merged to the original file.

The decryption phase is illustrated in the following figure.

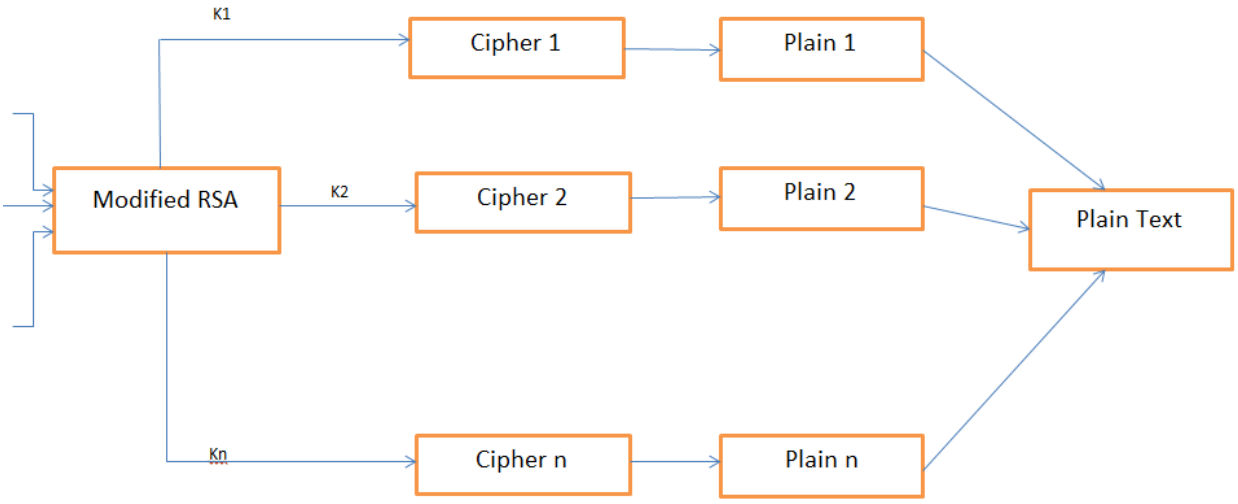

Figure 5.4: Decryption Phase

5.3 Cloud Computing Security Architecture

In order to ensure file security on cloud, the above proposed hybrid cryptosystem is deployed on cloud. We assume cloud server is trusted, but in order to prevent tampering/misuse of data by intruder or data leakage or other security concerns, the data stored at server is in the encrypted form.

We broadly classify the scheme deployed on cloud in three phases:

- Registration Phase
- Uploading Phase
- Downloading Phase

We used OpenNebula toolkit to set up cloud environment. In OpenNebula we have one front node and n cluster nodes. The VM's are deployed from front node to the corresponding cluster node.

5.3.1 Registration Phase

In the Registration Phase, the client registers himself in order to upload and view his files. In the process of registration, the client sends its request to front node and the front node assigns the VM of the cluster node, which has minimum load (no of request) to the client. At the end of registration the client is registered with IP address of corresponding

VM. Whenever he again issue his request, the request is transferred to its corresponding VM. The encrypted files, encrypted blowfish keys, public SRNN keys are stored at his registered VM.

5.3.2 Uploading Phase

The steps in the uploading phase are as follows:

Step: 1: The client will send request to front node to authenticate himself.

Step 2: The front node authenticate user and password and finally send the corresponding IP address of the VM to which user was registered.

Step 3: The files are uploaded by the client to the server. We assume server is trusted but the data stored on the server is in the encrypted form as it will protect the data from misuse or tampering.

Step 4: The files being uploaded to the server follows proposed hybrid cryptosystem.

Step 5: The encrypted slices and Blowfish encrypted keys are stored in VM's data store.

Step 6: The SRNN private keys are send to user and finally they are deleted form the server so that only the authenticated user is able to view his uploaded file.

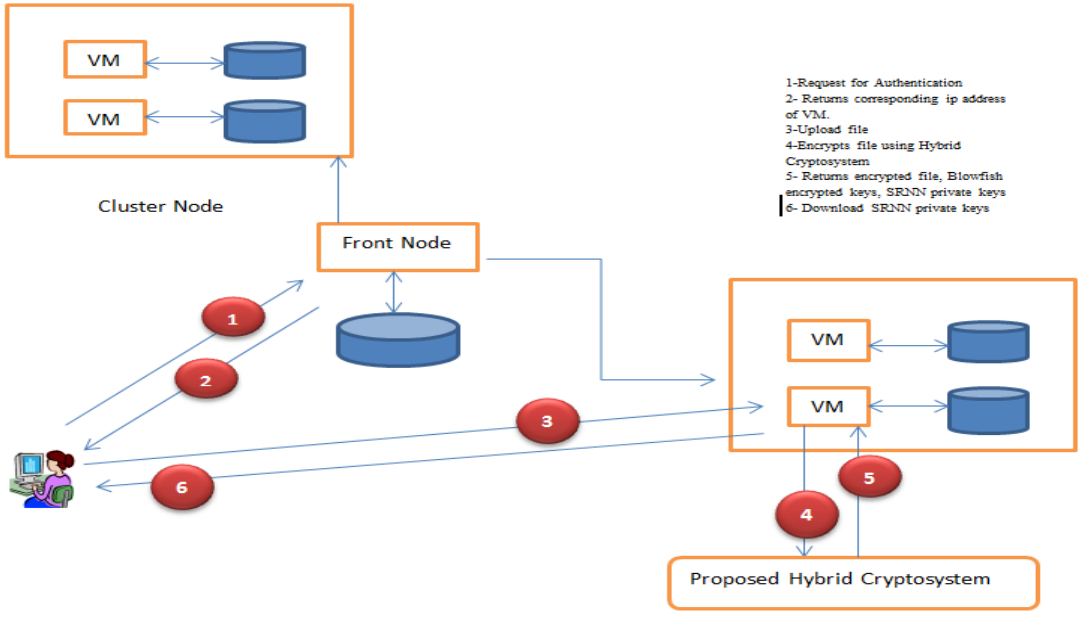

Figure 5.5: Uploading Phase

Following the above scenario, the files can be uploaded by the client to the server. At the end the server is left with encrypted files and encrypted keys.

5.3.3 Downloading Phase

The steps in the downloading phase are as follows:

Step: 1: The client sends request to front node to authenticate himself.

Step 2: The front node authenticates user and password and finally send the corresponding IP address of the VM to which user was registered.

Step 3: The client uploads SRNN private keys for the corresponding n slices.

Step 4: The SRNN private keys decrypt Blowfish encrypted keys and finally the encrypted slices are decrypted by Blowfish keys.

Step 5: The decrypted files are merged to get original file.

Step 6: The decrypted file is downloaded and viewed at client end.

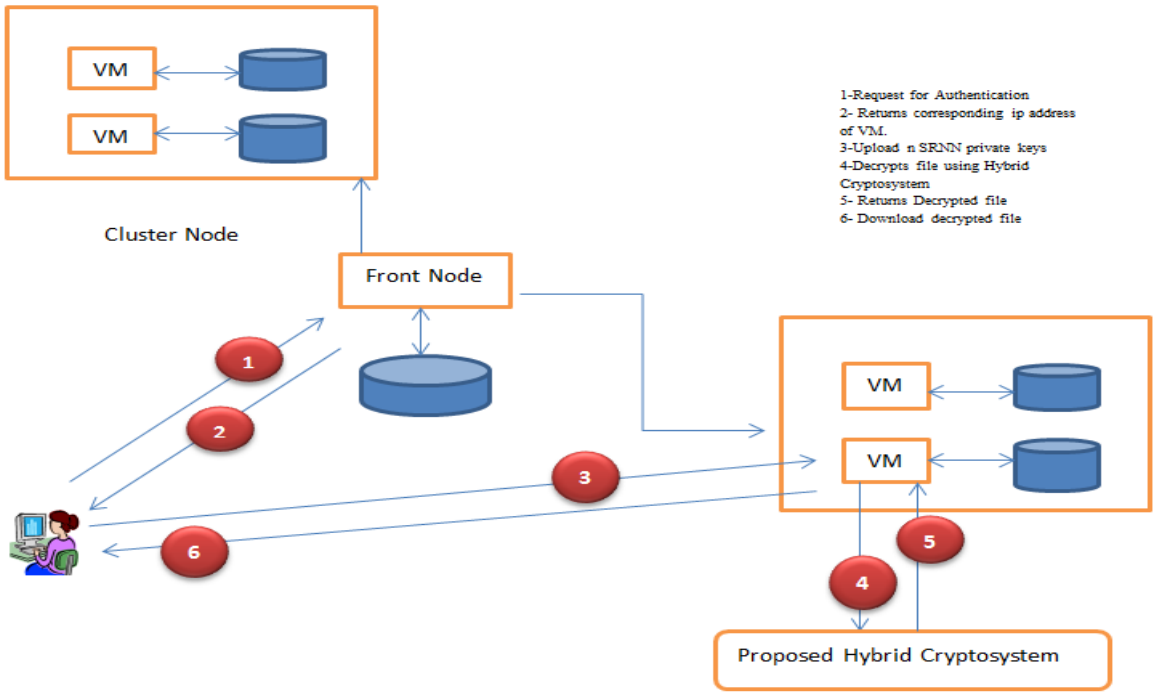

Figure 5.6: Downloading Phase

Following the above scenario, the files can be downloaded by the client from the server. The above scheme is tested on various type of file which are: image, audio, text, pdf and word files.

Chapter 6

Results

The cloud environment used for deploying the proposed hybrid cryptosystem is created by OpenNebula toolkit.

6.1 Registration Phase

In registration phase, the client sends its request to front node and the front node assigns the VM of the cluster node, which has minimum load (no of users) to the client. File shown below, is used to check the VM with least load at that instant of time.

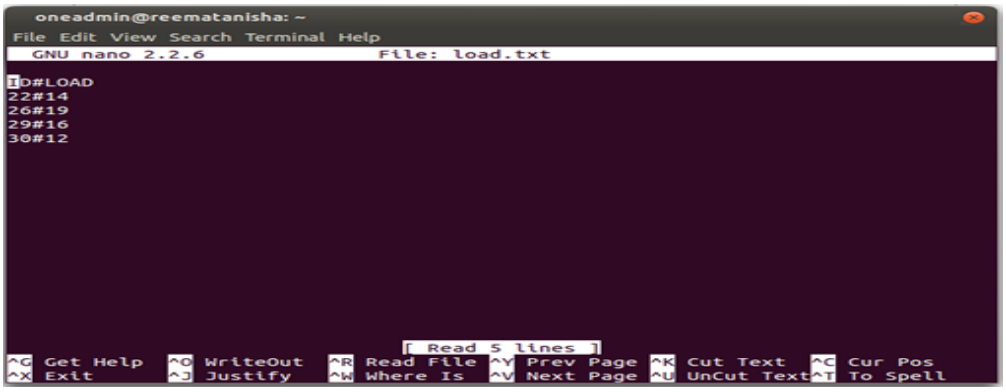

Figure 6.1: Load File

The above file serves as a database which shows the number of user registered with the VM. The new user is registered with the VM with minimum load and the load of registered VM is incremented by 1. At the time of termination of user account from cloud, the corresponding load is decremented by 1.To register himself the registration frame is used, which has been shown below.

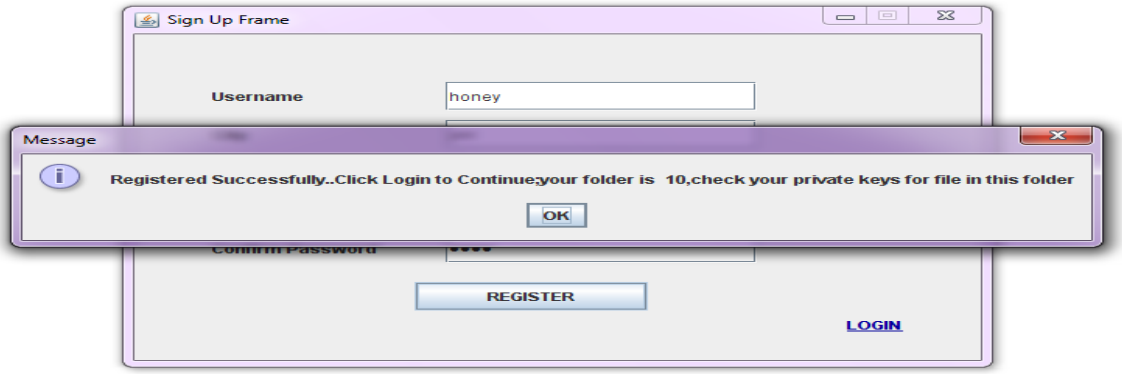

Figure 6.2: Registration Frame

In the above frame, the user registers himself. The user will fill the required fields, the fields are updated at the server. A unique id is generated by the front node. This unique id serves as a directory both at client and corresponding VM. At client end, the private keys corresponding to SRNN are saved in this directory. At the VM, the public keys corresponding to SRNN, encrypted keys and encrypted slices are saved in this directory Also the SRNN public keys are saved at corresponding VM. In the directory, a database is created at VM that contains the names of the files uploaded by the client. At the end of registration the client is registered with IP address of VM with minimum load. Whenever he again issue his request, the request is transferred to its corresponding VM.

6.2 Authentication Phase

In authentication, the client's login credentials are verified by the front node and finally the corresponding IP address of VM is send to the client. The login frame is used to enter the login credentials.

6.2.1 Login Frame

If the user is not registered, then first he will register himself using register button. If the username and password matches then the user will proceed to next frame, otherwise an error notification is generated.

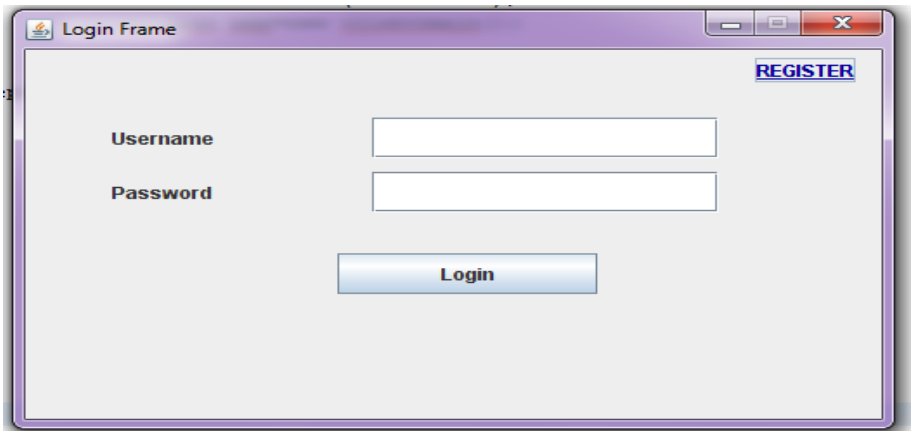

Figure 6.3: Login Frame

6.2.2 Selection Frame

Figure 6.4: Selection Frame

The selection frame allows the client to upload a file or view a file that has been previously uploaded.

6.3 Uploading Phase

In uploading phase the files are uploaded by the client and determine the number of slices that needs to be done. The file is sliced into n slices and encrypted by its corresponding key given by user. In this phase public, private keys that encrypt the keys are also generated. At the end, we are left with public, private, encrypted keys and encrypted slices.

6.3.1 Upload Frame

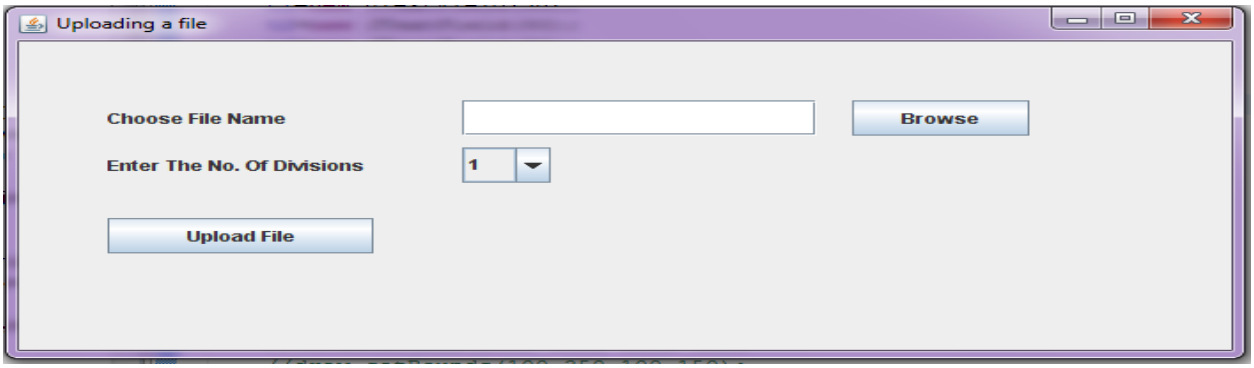

Figure 6.5: Upload Frame

In this frame, client will upload the file. The path is determined using browse button. The client will specify the division from the combo box. The division can range from 1 to 6. 5 types of files are supported that is image, audio, text, word and pdf. Then the file is uploaded to the VM using upload button. After uploading, the client gets the notification and he further proceeds to enter the key for each slice.

6.3.2 Encryption Frame

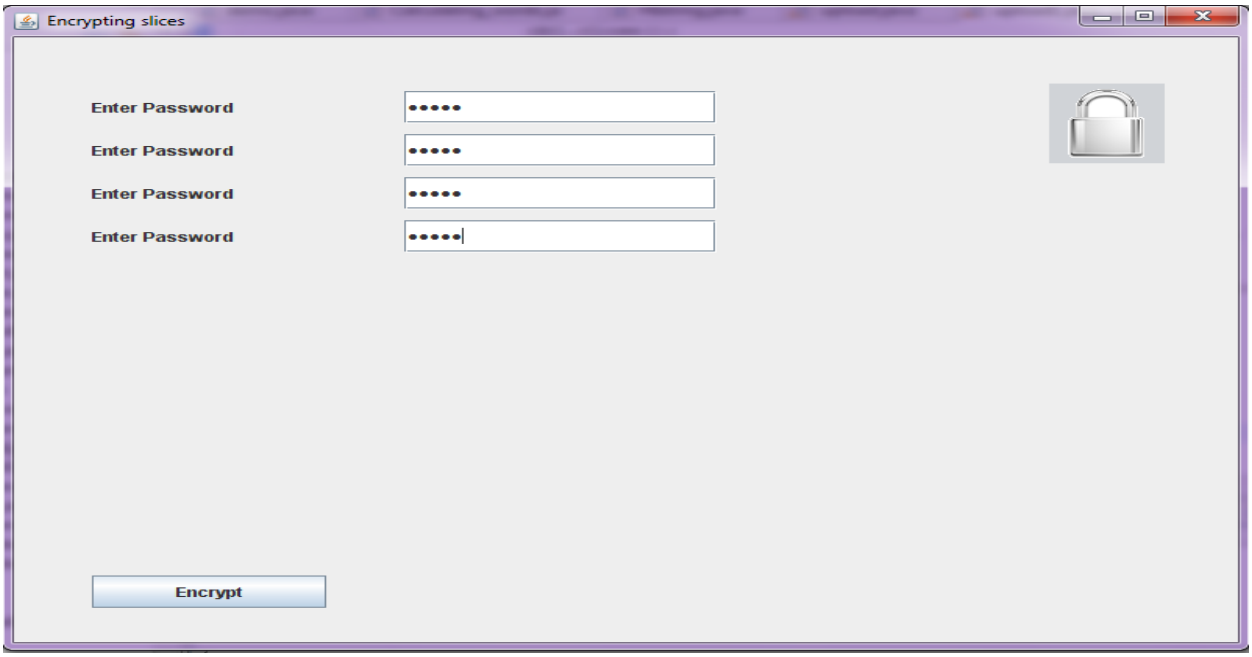

Figure 6.6: Encryption Frame

Here according to the number of slices defined in the previous uploaded frame, the client has to enter the key corresponding to each slice. The key is then encrypted by using public key of SRNN which is unique to each key of slice. The key is used to encrypt the slices. Encrypted slices are then stored at the VM datastore. The encrypted keys and public keys are also stored at VM and the private keys are stored at client end. Thereafter, notification is given that the file is successfully encrypted and stored at VM.

6.4 Downloading Phase

In this phase, the encrypted slices are decrypted using decrypted keys which are decrypted by client private key which are uploaded by client. Finally the client gets the original file.

6.4.1 Input Frame

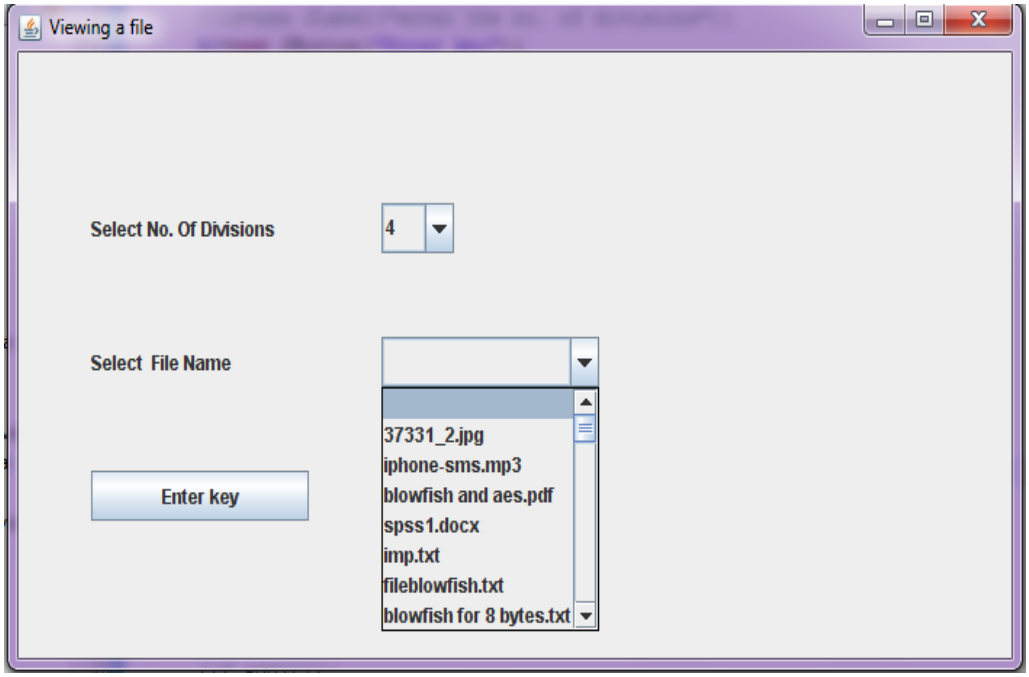

Figure 6.7: Input Frame

In this frame, the client will select the number of divisions. The user will select the corresponding filename that he/she want to view the file. These filenames are the names of files that have been previously uploaded by the client. Thereafter, the client will proceed to upload the private keys of the file.

6.4.2 Decryption Frame

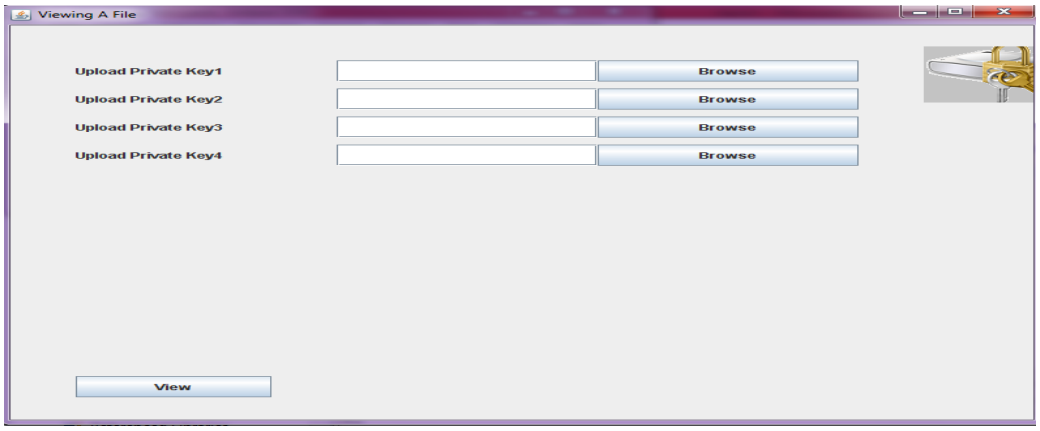

Figure 6.8: Decryption Frame

In this frame, the client will upload the private keys. These private keys are stored at the client end. VM will read the SRNN private key from the uploaded files; these private keys will decrypt the encrypted keys that is stored at VM. Thereafter, these decrypted keys will decrypt the encrypted slices at VM. Finally the decrypted slices will be combined to form original file that will be downloaded and viewed at client end that is shown in next frame.

6.4.2.1 Decrypted File

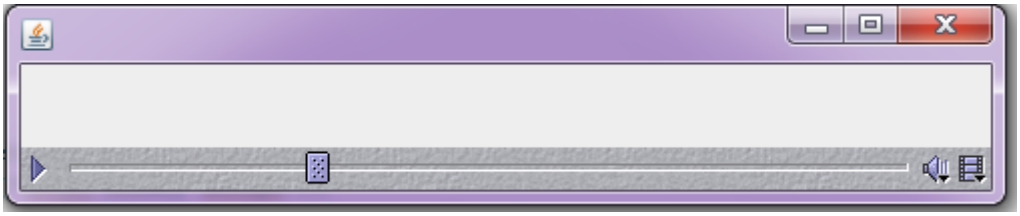

Figure 6.9: Decrypted Audio File

Figure 6.10: Decrypted Image File

The decrypted file is shown for audio and image file. The audio file plays through an mp3 player. Similarly, it can be done for pdf , text files and word files.

Chapter 7

Conclusion and Future Work

This chapter discusses the conclusions of the work presented in this thesis. The chapter ends with a discussion of the future direction which thesis work can take.

7.1 Conclusion

Cloud Computing has become the next generation architecture of IT Enterprise. Social media channels, corporate structures and individual consumers are all switching to the magnificent world of cloud computing. Although cloud providers offer security, but files are breached at some point of time by effective cryptanalysis, irrespective of its complex algorithmic design. Many cloud providers is restricted to a practice of following any one single encryption scheme and that too for a single iteration on a single file basis. So, a need for "practically strong and infeasible to get attacked" scheme becomes vital. The proposed hybrid cryptosystem serves the purpose of providing file security.

The symmetric algorithm (Blowfish) used in hybrid cryptosystem has best practice to avoid data misuse when compared with other algorithm. Secondly, in the case of encryption/decryption time, the blowfish has the best performance which proves it has the best CPU Clock Cycles in the terms of throughput. The asymmetric algorithm (SRNN) in hybrid cryptosystem uses a small range of natural numbers which are generated from session to session and thus, serves as a better solution by providing balance between both speed and security. The splitting of files and encrypting the files slices with different key provides a system which is "infeasible to get breached". Hence, proposed cloud architecture is highly efficient to ensure file security on cloud.

7.2 Future Work

Future opportunities could explore following:

- The proposed framework has been implemented on image, audio, pdf , word, text files which can be further enhanced to encrypt the video files.

- The user-level management of load can be enhanced by managing the load on request basis in proposed framework.
- An efficient approach is needed that manages the public, private keys when the number of slices increases. Moreover management of keys become a tedious task and additionally also it increases time to generate that number of SRNN public and private keys.

References

[1]P. Mell and T. Grance, "The NIST Definition of Cloud Computing", NIST, 2010.

[2]R . Buyya *et al.*, "Introduction to Cloud Computing", *Cloud Computing Principles and Paradigms,* John Wiley & Sons, 2011.

[3]*http://www.cloudtweaks.com/2012/04/cloud-computing-types-of-cloud-and-their-relevance-part-2.*

[4]B.Rimal *et al*., "A Taxonomy and Survey of Cloud Computing Systems", *Proc. Fifth International Joint Conference on INC, IMS and IDC,* pp: 44-51, held at Korea, August 25-27, 2009.

[5]N. Mangtani and S. Bhingarkar, "The appraisal and Judgment of Nimbus, OpenNebula and Eucalyptus", *International Journal of Computational Biology*, Vol. 3, Issue 1, pp: 44-47, 2012.

[6]V. Kumar *et al.*, "Cloud Computing: Towards Case Study of Data Security Mechanism", *International Journal of Advanced Technology & Engineering Research (IJATER),* Vol. 2, Issue 4, pp: 1-8, July 2012.

[7]A. Pillai *et al.*, "A Study on Open Source Cloud Computing Platforms", International *Journal of Multidisciplinary Management Studies*, Vol.2, Issue 7, July 2012.

[8]*http://www.techopedia.com/definition /28560/data-as-a-service-daas.*

[9]E. Mohamed and H. Abdelkader, "Enhanced Data Security Model for Cloud Computing", *Proc. Eighth International Conference on Infomatics and Systems*, Vol.2, No. 2, pp: CC-12-CC-15, held at Cairo, May 14-16, 2012.

[10] D. Chen *et al.*, "Data Security and Privacy Protection Issues in Cloud Computing", *Proc. IEEE International Conference on Computer Science and Electronics Engineering*, Vol. 1, pp: 647-651, held at Hangzhou, March 23-25, 2012.

[11]A. Mathur , "An ASCII value based data encryption algorithm and its comparison with other symmetric data encryption algorithms", *International Journal on Computer Science and Engineering* , Vol. 4, No. 9, pp: 1650- 1657, September 2012.

[12]E. Lackey, Red Hat Certificate System 8.0, 8.0.5 ed., 2009.

[13]J. Thakur and N. Kumar, "DES, AES and Blowfish: Symmetric Key Cryptography Algorithms Simulation Based Performance Analysis", *International Journal of Emerging Technology and Advanced Engineering*, Vol. 1, Issue 2, pp: 6-12, December 2011.

[14]N. Li, "Research on Diffie Hellman Key Exchange Protocol", *Proc. Second International Conference on Computer Engineering and Technology (ICCET)*, Vol. 4, pp: 634 -637, held at Chengdu, April 16-18, 2010.

[15]J. Yadav *et al*., "Modified RSA Public Key Cryptosystem Using Short Range Natural Number Algorithm", *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol. 2, August 2012.

[16]*http://en.kryptotel.net/encryption.html.*

[17]X. Wen *et al., "*Comparison of Open-Source Cloud Management Platforms: OpenStack and OpenNebula", *Proc. Ninth IEEE International Conference on Fuzzy Systems and Knowledge Discovery*, pp: 2457-2461, held at Sichuan, May 29-31, 2012.

[18]C. Yang *et al.,* "On Construction of Cloud IaaS Using KVM and OpenNebula for Video Services", Proc. *Forty one IEEE International Conference on Parallel Processing Workshops*, pp: 212-221, held at Pittsburgh, September 10-13, 2012.

[19] H. Tianfield, "Security Issues in Cloud Computing", *Proc. IEEE International Conference on Systems, Man, and Cybernetics*, pp: 1082–1089, held at Seoul, October 14-17, 2012.

[20] A. Behl, "Emerging Security Challenges in Cloud Computing", *Proc. World Congress on Information and communication Technologies*, pp: 217-222, held at Mumbai, December 11-14, 2011.

[21] R. Wadhawan, " Cloud Computing Security Issues in Infrastructure as a Service", *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol. 2, Issue 1, January 2012.

[22]K. Nafi, "A Newer User Authentication, File encryption and Distributed Server Based Cloud Computing security architecture", *International Journal of Advanced Computer Science and Applications,* Vol. 3, No. 10, 2012.

[23] D. Mukhopadhyay, "Enhanced Security for Cloud Storage using File Encryption", *Proc. International Conference on Distributed Computing and Internet Technologies*, 2013

[24] A. Mohta *et al.*, "Robust Data Security for Cloud while using Third Party Auditor", *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol. 2, Issue 2, February 2012.

[25] G. Zhao *et.al*, "Deployment Models: Towards Eliminating Security Concerns from Cloud Computing", *Proc. International conference on High Performance Computing and Simulation (HPCS)*, pp: 189-195, held at Caen, July 2010.

[26] D. Salama *et al.*, "Performance Evaluation of Symmetric Encryption Algorithms", *International Journal of Computer Science and Network Security*, Vol. 8, No.12, pp: 280-286, December 2008.

[27]C. Nayak, "Performance of Various Algorithm Used in Cryptography", *International Journal of Management, IT and Engineering*, vol. 2, issue 7, pp: 123-128,July 2012.

[28]D. Srinivasarao *et al.*, "Analyzing the Superlative symmetric Cryptosystem Encryption Algorithm", *Journal of Global Research in Computer Science*, Vol. 7, July 2011.

[29]G. Manikandan *et al.*, "A modified cryptographic scheme enhancing data", *Journal of Theoretical and Applied Information Technology*, Vol. 35, No.2, January 2012.

[30]Y.P. Singh *et al., "*On the security of Joint Signature and Hybrid Encryption", *Proc. Thirteenth IEEE International Conference on Networks*, Vol. 1, held at Kuala Lumpur, 2005.

[31]G. Toraldo, "OpenNebula 3 Cloud Computing", *OpenNebula and Why it Matters*, May 2012.

[32]G. Laszewski *et al.*, "Comparison of Multiple Cloud Frameworks", *Proc. Fifth IEEE International Conference on Cloud Computing*, pp: 734-74, held at Honolulu, June 24-29, 2012.

[33]*http://opennebula.org/*

[34]G. Manikandan*et al.*, "A Hybrid Approach for Security Enhancement by Modified Crypto-Stegno Scheme", *European Journal of Scientific Research*, Vol. 2, pp: 206-212, 2011.

[35] W. Stallings, "Cryptography and Network Security", Fourth ed., June 2010.