A TRUST BASED SECURITY OF RESOURCES IN CLOUD ENVIRONMENT

Thesis

Submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

by

USHA D

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA

SURATHKAL, MANGALORE - 575025

February, 2018

To my family

DECLARATION

By the Ph.D. Research Scholar

I hereby *declare* that the Research Thesis entitled **A TRUST BASED SECURITY OF RESOURCES IN CLOUD ENVIRONMENT** which is being submitted to the **National Institute of Technology Karnataka, Surathkal** in partial fulfillment of the requirements for the award of the Degree of **Doctor of Philosophy** in **Computer Science and Engineering** is a *bonafide report of the research work carried out by me*. The material contained in this Research Thesis has not been submitted to any University or Institution for the award of any degree.

USHA D

Reg. No.: 112032 CO11F02

Department of Computer Science and Engineering

Place: NITK, Surathkal.

Date: February, 2018

CERTIFICATE

This is to *certify* that the Research Thesis entitled **A TRUST BASED SECURITY OF RESOURCES IN CLOUD ENVIRONMENT** submitted by **USHA D.**, (Reg. No.: 112032 CO11F02) as the record of the research work carried out by her, is *accepted as the Research Thesis submission* in partial fulfillment of the requirements for the award of degree of **Doctor of Philosophy**.

(K. Chandrasekaran)
Research Supervisor

Chairman - DRPC
(Signature with Date and Seal)

ACKNOWLEDGMENT

I would like to take this chance to thank those people who have made this thesis possible.

My sincere gratitude to my adviser Prof. K. Chandrasekaran for accepting me into the PhD program and for keeping constant faith in me. No matter what new project I start up, he is always supportive and remarkably perceptive in advising in the right direction and correcting the vital details. He is the epitome of professionalism and perfectionism. Without his continuous encouragement, I would still be in my PhD dreams. He is a great mentor and I learned so much from him through his long academic journey. I could not have finished my thesis in time without his untiring help in making every sentence concise and correct.

Special thanks are due to my Research Progress Assessment Committee (RPAC) members Dr. P. Sam Johnson (MACS Dept) and Dr. Mohit. P. Tahiliani (CSE Dept) for their constant advise in improvising my research work and special thanks to Dr P Sam Johnson for continuously guiding me in brushing my mathematics for my research work.

Special thanks to Prof. P.Shanti Thilagam, Head of Department, Department of Computer Science and Engineering, for encouraging research related activities in the department and cultivating an open and free research environment.I would like to extend my sincere thanks for the Computer Science department faculty and staff for their untiring support in successful completion of my research work.

Special thanks to my parents and my daughter for their constant and untiring support during the whole tenure of my Ph.D program. Special thanks to my close friends who have been instrumental in uplifting my morale during my research work.

Place: NITK, Surathkal Usha D

Date: February , 2018

ABSTRACT

Cloud computing has been attracting the attention of several researchers both in the academia and the industry as it provides many opportunities for organizations by offering a range of computing services. Before cloud computing to become acceptable to everybody both the enterprises and individuals, several issues have to be solved. One of the most important aspects that need to be paid special attention is the cloud security. Trust management is one of the important components in the cloud security that needs special attention.

The trust management systems proposed for cloud computing by various researchers have been studied with special emphasis towards their capability and applicability in a practical heterogeneous cloud environment besides implementabilty. An effective trust management system helps cloud service providers and consumers to reap the benefits brought about by cloud computing technologies. Despite the benefits of trust management, several issues related to general trust assessment mechanisms, distrusted feedbacks, poor identification of feedbacks, privacy of participants and the lack of feedbacks integration still need to be addressed. Traditional trust management approaches such as the use of Service Level Agreement are inadequate for complex cloud environments. Due to the multiple vulnerabilities like identification, privacy, personalization, integration, security, and scalability in the existing models, it is proposed for a strong trust model which would create a strong trust between the entities or resources of the cloud. To build a strong trust a strong trust path is necessary between the entities where all the entities in cloud and cloud computing environment would trust each other and the entities that have communication would have valid trust on each other.

A mathematical model was proposed to calculate basic trust, dynamic trust and trust for migration.Basic trust was calculated using entropy. Based on the initial trust of the

system, a new trust for successful transactions was calculated as dynamic trust.The trust models proposed were implemented using Family Gene Genetic Algorithm. The algorithm gives an optimal solution for a large set of data. The implementation of proposed model using this adapted algorithm showed that the resources on a cloud with a strong trust value would always be available for performing any successful transaction.

We have proposed a end-to-end trust model which calculates trust based on four parameters namely: utilization, saturation, failure rate and availability. In this model we simulated the results using Monte Carlo method to check with the trust decision making policy. We found from the results that the trust decision is high or low based on the availability of the resources. Based on the trust model and the adapted algorithm the performance of the system using perceived factors were evaluated. The implementation on two different cloud platforms, namely Aneka and Opennebula showed that the model would give better results in terms of Process Time, System Time and Compute Time. Thus we conclude that our model proposes a strong trust path between the entities or resources of the cloud.

Keywords: Cloud Computing, Trust Management, Entropy, Family Gene Genetic Algorithm, Perceived Factors

Contents

List of Figures

List of Abbreviations

CC	Cloud Consumer
CSP	Cloud Service Provider
IaaS	Infrastructure-as-a-Service
PaaS	Platform-as-a-Service
SaaS	Software-as-a-Service
FIM	Federated Identity Management
PII	Personally Identifiable Information
SLA	Service Level Agreement
P2P	Peer-to-Peer
PKI	Public Key Infrastructure
CA	Certification Authority
CP	Certification Policy
STAR	Security, Trust & Assurance Registry
CAIQ	Consensus Assessments Initiative Questionnaire
CCM	Cloud Controls Matrix
CTP	Cloud Trust Protocol
SP/SPS	Service Policy/Service Practice Statement
FBCT	Family-gene Based model for Cloud Trust
TCP	Trusted Computing Platform
JVM	Java Virtual Machines
QoS	Quality of Service
FIFO	First-in-First-Out
MTCEM	Multi-tenancy trusted computing environment model
GA	Genetic Algorithm
FGA	Family Gene Genetic Algorithm

Chapter 1

INTRODUCTION

In this chapter, we describe what is trust and cloud computing. We start with the description of trust and types of trust and the description of cloud computing and various service models of cloud. We also describe the relation between trust and cloud computing.

1.1 Definition

Recently, cloud computing has been receiving much attention as a new computing paradigm for providing flexible and on-demand infrastructures, platforms, and software as services. According to National Institute of Standards and Technology (NIST) Mell et al. (2011), Cloud computing is a model for enabling ubiquitous, 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. Cloud computing offers service dynamism, elasticity and a wide variety of choices to enterprises. In today's competitive environment, enterprises cannot ignore these services. Flexible cloud computing services require one party CC rely on the actions of another party, i.e. CSP, therefore, trust has become a vital component of such services.

Trust is a complex social phenomenon. Based on the concepts of trust developed in social sciences Firdhous et al. (2012a), trust is a mental state comprising: (1) expectancy in which the trustor expects a specific behavior from the trustee, such as providing valid information or effectively performing cooperative actions; (2) belief in which the trustor believes that the expected behavior occurs, based on the evidence of the trustee's competence, integrity, and goodwill; (3) willingness to take risk, in which the trustor is willing

to take the risk for that belief Xiong and Liu (2003). Although intuitively easy to comprehend, the notion of trust has not been scholarly defined.

In order to use cloud services, an enterprise needs to give up control of its assets (i.e. data) to the CSP. Loss of control on stored data in cloud triggers uncertainty about data confidentiality, privacy, integrity and availability for CCs which adversely affects adaptability of cloud computing services. Enterprises have to remember that as compelling as cloud services are, it isn't without potential problems. Enterprises also must consider the possibility that data could be stolen or viewed by people who are not authorized to see it. CSPs take all user critical information and put on virtual servers while users may never know if stored information will be used against their consent. CSP can be forced by government agencies to reveal stored data. For individual users of famous CSPs like Microsoft, Amazon, Apple and etc., the bigger risk is to lose the access to their online accounts that store numerous amount of personal data such as pictures and videos, email correspondences and banking information because of accusation of terms of service violation. Although most of such cases can be resolved, it may take a long time for communication with CSP and meanwhile, users do not have access to their personal data. According to Armbrust et al. (2010), trust management is ranked among the top 10 obstacles to adopting cloud computing. Adaptability of cloud services depends on the establishment of trust on CSP to assure data security and guarantee cloud performance and behavior.

1.2 Overview of Cloud Computing

Cloud computing is a paradigm that provides flexible and on-demand infrastructures, platform and software as services. Cloud computing has emerged as a result of combining the benefits of grid computing and virtualization with those of service-oriented computing to utilize computer resources (data centers) and deliver computer resources as services. Cloud computing uses virtualization techniques to design and govern the services it offers to automate business logics. Cloud environments promise several benefits such as reduced expenses and simplicity to service providers and service requesters.

Cloud services are established based on five essential characteristics. The first characteristic is the on-demand self-service, which enables consumers to provision computing power, storage, networks and software in a simple and flexible way. Second is broad network access in which cloud service consumers can access available computing resources over the network. Third is resource pooling where computing resources are pooled to serve multiple cloud service consumers based on a multitenant model where physical and virtual computing resources are dynamically reassigned on demand, fourth is rapid elasticity where computing resources are elastically provisioned to scale rapidly based on the cloud service consumers need, and the last one is measured service where computing resources usage is monitored, metered, controlled, and reported to provide transparency for both CSPs and consumers.

1.2.1 Cloud Service Models

Cloud services have three different models. One of the service models is Infrastructure as a Service (IaaS) which provides raw storage space, computing, or network resources for the customers to run and execute any software that they choose. The other service model is Platform as a Service (PaaS) which the CSP provides the hardware and a toolkit and a number of supported programming languages to build higher level services. The users who are typically software developers host their applications on the platform and provide these applications to the end-users. The third service model is Software as a Service (SaaS) which the CC is the end-user who just have access to the complete applications running on a cloud infrastructure and offered on a platform on-demand.

The different infrastructure deployment models are distinguished by their architecture, the location of the datacenter where the cloud is realized, and the needs of the CC. Public clouds are one of the deployment models which run applications from different CCs who share the infrastructure and pay for their resource utilization on a utility computing basis. Private clouds are another deployment models which are built for the exclusive use of one CC, who owns and fully controls the cloud. The third deployment model in cloud computing is Community clouds in which CCs who have similar requirements, can share an infrastructure and configuration and management of the cloud. The last deployment model

is Hybrid clouds which consist of any composition of other deployment models.

Limitations of Cloud Computing There are many benefits of cloud computing like secure and affordable managed hosting, accessibility of data from anywhere at any-time, scalability, off-site back-up, etc. But there are some limitations like network connection, control of data security, integration, peripherals, generic which caters to multi-tenancy which may not be favourable for all the users.

1.2.2 Risks and Threats of Cloud Computing

Trust is one of the main concerns for the consumers to adopt Cloud computing Pearson and Benameur (2010). Based on the common elements between literature in this area, cloud computing risks and threats are:

Lack of Confidentiality : According to standard computing literature Pearson (2009), the IT Security depends on the Confidentiality of data. In cloud computing confidentiality is achieved by encryption. To achieve confidentiality the encryption schemes need to be secure for the long term. Also, confidentiality is threatened by decrypting data while using it. Furthermore, information leakage vulnerability in third-party compute Clouds pose threats to Confidentiality too.

Lack of Reliability : Availability of resources in cloud computing is one of the biggest concerns for the consumers. Availability not just consists of reachability but also success rate of transactions. However, the CSPs look at availability as a way to represent the level of reliability of the cloud services.

Lack of Identity Management : Federated Identity Management (FIM) is an important terminology in the case of federated clouds. FIM provides a tool for sharing resources and services among different enterprises while the directory services, authentication and authorization do not have same technologies. But it is possible that FIM has the enterprises to use authentication broker (a common trusted third party)as identity management provider. However, this kind of application can be considered as a threat for the security of the entire service inventory Pearson and Benameur (2010).

Lack of Privacy : From the CCs perspective, privacy is an important concern in cloud computing and entails the protection and appropriate use of the personal information of CCs, and the meeting of expectations of CCs about its use. For organizations, privacy

entails the application of laws, policies, standards, and processes by which Personally Identifiable Information (PII) of individuals is managed. Context is very important as privacy threats differ according to the type of cloud scenario. Some cloud application areas and services might face a very low privacy threat, for example, if the service is to process information that is public. It is possible only if the service handles personal information in the sense of collecting, transferring, processing, sharing or storing it, that there could be a privacy risk and privacy needs to be taken into account Pearson and Benameur (2010). Privacy becomes very important when multiple services need to be combined to enable a new service. For example, print on demand service in cloud which can be provided by combining a printing service with a storage service can cause a privacy threat since the information regarding the services might need to flow across service providerś boundaries.

Lack of Reputation : With the growing number of CSPs, the CCs are facing a challenge to select the best and most appropriate providers from numerous offers. In Subashini and Kavitha (2011), the author points out a typical scenario, where a CSP can offer a secure service while another may not, if the latter charges half the price, the majority of organizations will ask for the latter one as there is no real way to explore the difference.

Lack of Service Level Agreements (SLAs) Standard SLAs in the present Cloud market is also one of the obstacles that the consumers face while adopting the services offered by the CSPs. Consumers might face problems that occur from vendor lock in, insufficient security measures, data unavailability, hidden costs, and non transparent infrastructure. In most cases, SLAs are created to protect the vendors/providers and not the CCs. Most of the above mentioned problems are overlooked in current SLAs offered by the CSPs.

Lack of transparency : Providers of cloud computing technologies may be unlikely to share information about processes, operations, controls, and methodologies, especially related to IT general controls affecting the cloud environment. There are some transparent security principles help identify the types of information that should and should not be disclosed. Those that should be disclosed are: Common security features such as the use of firewalls and encryption of data in transmission or at rest should be disclosed because they are considered basic security features that most security people would expect to be in place anyway, performing disclosure when it is imperative due to a legal or regulatory

requirement, Security architectural details that may either help or hinder security management should be disclosed, governance responsibilities of the CC versus those of the CSP should be clearly articulated so that CCs are clear on what they must do themselves to help protect their data. Also, there are some principals for which disclosure is not recommended which are: do not disclose anything that could create risk to the datacenter or to the integrity of data stored in the datacenter, if disclosure could create potential harm for a CC or partner, it should be avoided, avoid disclosures that could create undue liability for the CSP, if disclosure would result in breach of a legal or regulatory requirement, it should be avoided.

To minimize the risks or threats, strong security is needed and trust is the first step to security. When a strong trust is implemented in the first place it reduces the risks like reputation or privacy which in turn increases the security of the system.

1.3 Trust

Trust is often measured/related to terms like cooperation, confidence, and predictability. According to Gambetta et al. (1988) trust is the probability that an entity will perform an action that is beneficial or at least not detrimental to us is high enough for us to consider engaging in some form of co-operation with it.

1.3.1 Trust Classification

Trust can be classified as:

Infrastructure Trust: The service provider provides a secure infrastructure like workstations, local area network, and servers, which can be implemented using various security protocols and other protective measures Cheng et al. (2012).

Delegation Trust: A service provider may trust a third party entity to take decisions on its behalf for resource sharing.

Services Provision Trust: Services Provision Trust describes trustors trust in providing services for trustees.

Entity certification: A third party certification to a cloud entity based on its trustworthiness.

Resources Access Trust: Resources Access Trust is a kind of trust relationship for the

purpose of accessing resources Cheng et al. (2012).

1.3.2 Importance of Trust in Cloud Computing

Trust is an important concept in distributed computing environments and plays a critical role in ensuring and enhancing system security and adaptability. There are a number of challenges that affect trust in different distributed systems for example in social networks, hackers are a serious threat since they try to access the user accounts and use them as a trusted source to spread malware. Also, tools that help in managing, viewing, querying, transferring and displaying personal data in the system and provide searching and mining profile data can be considered as another trust issue in social networks. In P2P networks, security implications arise from abusing trust between peers. One of the trust issues exists in peer to peer networks is the distributed denial of service (DoS). Also, attackers can make use of the querying nature of P2P networks to overload the network by sending a massive number of queries to peers, make the portions of the network inoperable. Since in P2P network the peers should contribute in resource distribution process, peers data stream may be compromised by fellow peers who assist in transmitting the data in the system, and sometimes free to freeload off other peers.

Cloud Computing supports four deployment models which are public clouds, private clouds, community clouds and hybrid clouds. In a private cloud, trust is not applicable if the third party is not involved. However, public clouds can introduce many security risks since controlling data in this deployment model is very challenging. Trust in community cloud depends on the role of the third party. If there is a third party involved, the trust risks are the same as the corresponding case in a private cloud. Otherwise, if the community cloud is managed by the organizations in the community, trust risks are limited to the trust relationships that are discussed and agreed between community members. In Hybrid clouds, since both the private clouds and public clouds are involved, all the trust issues related to public cloud shift to the hybrid cloud, too.

1.3.3 Objectives of the Research

1. To design and develop a conceptual and mathematical model for trust in cloud environment to secure accessibility and utility of resources.

2. To develop a Family Gene based approach for selecting available resources based on the trust values obtained from the mathematical model developed.

3. To design an end-to-end trust evaluation framework to secure accessibility, utility and computability of the available resources in cloud environment.

4. To evaluate the performance of designed trust model based on various perceived factors.

1.3.4 Problem Definition

Design and develop a trust based security solution for ensuring secure and optimal usage of resources in cloud environment.

1.3.5 Thesis Outline

Chapter 2 contains a complete details of trust, trust management and the relationship between trust management and cloud computing.

Chapter 3 contains a complete Literature Survey of the existing Trust Models in Cloud Computing paradigm. It also emphasizes on the drawbacks of the existing Models.

Chapter 4 contains the detailed discussion about the proposed mathematical models and a conceptual diagram of the work proposed. 3 Principles are defined in the proposed mathematical model in this Chapter.

Chapter 5 contains the mathematical proof of the 3 principals defined in Chapter 3 and some examples are illustrated for the easy understanding of the proposed Mathematical equations.

Chapter 6 includes the experimentation and implementation details of the mathematical model using an advanced optimization algorithm called Family Gene Genetic Algorithm. The chapter contains a detailed discussion about the adaptation of the Family Gene Genetic Algorithm for the implementation.

Chapter 7 highlights the End-to-End trust evaluation for secure accessibility of the available resources in cloud environment.

Chapter 8 introduces the concept of perceived factors and also describes the details of the selected perceived factors for evaluating our system performance.

Chapter 9 gives various conclusions of the research work done, limitations of the work and provides directions for future work.

1.4 Summary

Cloud computing is the fast emerging technology in current world. As this technology gives a pay-per-use of the resources without much investment from the user side its an easy tool for the user. Cloud can be deployed as private, public or hybrid model based on the security requirements of the user. Cloud also has several service models like Software-as-Service, Platform-as-service, Infrastructure-as-Service and several other types like Security-as-Service based on the usage requirement of the user. With this easy model catering to the user requirements also comes multiple risk factors like security, privacy, lack of and so on. Security being the major concern in cloud environment many researchers have proposed multiple security models.

Chapter 2

TRUST MANAGEMENT

In this chapter, we describe what is trust , trust management,and trust in cloud computing. We start with the description of trust, trust mechanisms and factors affecting trust We also describe the relation between trust and cloud computing.

2.1 Definition

Trust is a complex social phenomenon. Based on the concepts of trust developed in social sciences by Firdhous et al. (2012a), trust is a mental state comprising: (1) expectancy in which the trustor expects a specific behavior from the trustee such as providing valid information or effectively performing cooperative actions; (2) belief in which the trustor believes that the expected behavior occurs, based on the evidence of the trusteeś competence, integrity, and goodwill; (3) willingness to take risk, in which the trustor is willing to take the risk for that belief as explained by Xiong and Liu (2003). Although intuitively easy to comprehend, the notion of trust has not been scholarly defined.

2.2 Trust Management

Trust is often measured/related to terms like cooperation, confidence, and predictability. According to Gambetta et al. (1988) trust is the probability that an entity will perform an action that is beneficial or at least not detrimental to us is high enough for us to consider engaging in some form of co-operation with it.

Trust Management is known as degree of trustworthiness, quality and reliable entity engagement management. It is a strategy of developing trust between the entities in the system so that there is no detrimental effect on the system.

Trust management is originally developed by Blaze et al. (1996) to overcome the issues of centralized security systems, such as centralized control of trust relationships (i.e., global certifying authorities), inflexibility to support complex trust relationships in large-scale networks, and the heterogeneity of policy languages. Policy languages in trust management are responsible for setting authorization roles and implementing security policies. Authorization roles are satisfied through a set of security policies, which themselves are satisfied through a set of credentials. These techniques are considered as policy-based trust management because they rely on policy roles to provide automated authorizations. Later, trust management inspired many researchers to specify the same concept in different environments such as e-commerce, P2P systems, Web services, wireless sensor networks, grid computing, and most recently cloud computing.

2.2.1 Trust and Trust Management

It is known that trust is a belief and that there will be no risk between the two entities performing any action. The basic principle for any successful relationship is the base value of trust among the entities involved. Trust is one of the obligatory qualities in any relationship. It is due to this trust that any entity could cooperate beyond a system of formal and legal rules.

The basic nature of trust is found as the tension between depending upon another and instituting controls to make sure that other performs. The higher the risk the higher would be the loss. In human science or information technology, the trust plays a vital role in reconciling away fears and the willingness to become vulnerable to the other without controlling the other as stated by Habib et al. (2011).

Trust is usually defined as a relationship between a truster and trustee. Trust forms the basis for the trustee to use or manipulate resources owned by a truster. It is always assumed that trust is inversely proportional to the degree of risk in any transaction. In many current business relationships, trust is based on a combination of judgment or opinion based on face-to-face meetings or recommendations of colleagues, friends, and business partners.

Trust is a complex subject relating to belief in the honesty, truthfulness, competence, reliability, etc., of the trusted person or service. The significance of incorporating trust in distributed systems is that trust is an enabling technology. Its inclusion will enable Internet

commerce and seamless, secure agent-based applications. Despite the need to standardize trust and its related concepts, many researchers simply use and assume a definition of trust in a very specific way relating to topics such as authentication, or the ability to pay for purchases.

A trustworthy entity will typically have a high reliability and so will not fail during the course of an interaction, will perform a service or action within a reasonable period of time, will tell the truth and be honest with respect to interactions, and will not disclose confidential information. We define trust as "the firm belief in the competence of an entity to act dependably, securely, and reliably within a specified context" Grandison and Sloman (2000).

A trustor trusts a trustee with respect to its ability to perform a specific action or provide a specific service within a context. The entities involved in a trust relationship will be distributed and may have no direct knowledge of each other, so there is a need for mechanisms to support the establishment of trust relationships between distributed entities.

2.2.2 Trust Mechanisms

Trust models are the techniques that are used for evaluating trust in cloud services. They can be categorized in certain categories named trust mechanisms. The categories of trust models based on trust mechanisms as explained by Firdhous et al. (2012a) are:

Reputation Based: Trust and reputation are different from each other. Trust is the subjective expectation of one entity about another within a specific context at a given time as stated by Blaze et al. (1996). Reputation, on the other hand, is what is believed about an entity standing by the community. This belief can be derived from direct or indirect experiences collected in previous interactions between entities. It is important to note that trust can be used to determine the reputation of an entity, and vice versa.

Trust is considered between two entities, however the reputation of an entity is the aggregated opinion of a community towards that entity. In another word, an entity that has a high reputation is trusted by many entities in that community. An entity can use reputation to calculate the trust level of the trustee. In cloud computing, reputation is very important since it will impact cloud users. Therefore, CSPs are trying to achieve a higher reputation. Reputation is shown by a comprehensive score that is based on the overall opinion and

score for the major aspects of performance. At the very first time when a user wants to choose a cloud service, reputation of the CSP who offer the service is very important, but it may not be important in later stages as performance and reliability of the service can establish trust between user and CSP.

This category of trust models contains those trust models that collect the feedback and opinions from other CCs to evaluate trust on cloud services. The trust model selects the most reliable and trusted CSP by evaluating the CCs feedback.

Authentication Based: Encryption and Key Management are important technologies that can help secure applications and data in the cloud. PKI is a technology that introduces a trust mechanism to support digital signature, key certification, and validation, attribute certification and validation.

A public key certificate also contains a certificate policy (CP) extension. The certificate means that the issuing CA who conforms to the specified CP asserts that the subject CA has the certified public key, and the subject CA also adheres to the specified CP. As a result, to infer Alice belief in CA key and Bob key, she must trust that CP in the sense that any CA conforming to that CP will generate valid public key certificates as suggested by Huang and Nicol (2013).

Since PKI is currently practiced, trust in a certification authority (CA) with respect to issuing and maintaining valid public key certificates is based on the CA conformance with certain certificate policies. Certificate policies play a central role in PKI trust.

SLA Based: A service level agreement (SLA) is a legal contract between a cloud user and a CSP. SLA is a service level agreement. It is one of the approaches to establishing trust on CSPs. The entities that are providing services are required to follow standardized SLA, e.g., proposed by Cloud Computing Use Cases community as detailed by Wang et al. (2010a). SLA validation as stated by Haq et al. (2010), and monitoring schemes are used to verify the quality of CSPs and CCs are responsible for monitoring SLA violations. Since SLA compensation clauses are developed by the CSPs, CCs do not have enough chance to apply for compensation if an SLA violation happens and this is a problem due to lack of standardized SLAs for the stakeholders in the cloud computing market. However industry driven initiative has addressed this problem but still, it is not fully implemented. There

are a number of other issues with SLA based trust. First, SLA focuses on the visible elements of cloud service performance and does not address invisible elements such as security and privacy. Second, many cloud users do not have enough capability to perform SLA verification on their own and they need a professional third party help to provide these services. In a private cloud, the trusted broker or trust authority who is trusted in the trust domain of the private cloud can provide the users the services of SLA verification. In a hybrid cloud, a user within a private cloud might still rely on the private cloud trust authority to conduct SLA verification; however, in a public cloud, individual users and some small organizations without technical capability may use a commercial professional cloud entity as a trust broker. Trust establishment under this category is based on contracts and agreements signed by CSPs for the delivery of different services to CCs. SLA provides the basis for trust establishment. Various security concerns and quality of service attributes are included in contracts and agreements to establish trust on CSP.

Domain Based: Basic idea in the domain based trust model is to divide the Cloud into a number of autonomous domains and distinguish two types of within-domain and inter-domain trust relationships respectively. Within-domain trust values depend upon the transactions between the entities that are in the same domain. If an entity needs to compute the trust value for some other entity, it checks the direct trust table but if the direct trust value is not found, then it looks for the recommended trust values from other entities as explained by Kanwal et al. (2013).
The inter-domain trust relationship is using the trust relationship between the domains. There is an authentication mechanism for each domain which trusts the authentication mechanisms of other domains. If an entity is authenticated by one domain, then its authentication is acceptable by all other domains.

Platform Based: Platform based trust models consist of policies that ensure applications are executing on platforms that meet a specified trust assurance level and evaluate the confidence of CCs on using cloud services bunch on a specific platform. Therefore, by using this trust model, CCs can trust a CSP to use the offered platform as stated by Kanwal et al. (2013).

2.2.3 Factors Affecting Trust

Trust cannot be specified in a crisp value. To arrive at calculative value some basic factors always have an exclusive influence. Some of the few factors are listed below.Some of the key factors that are common to trust irrespective of the platform are:

i. Trust plays an important role in uncertainty and risky environment.

ii. Trust is the base platform on which certain decisions are made.

iii. Trust is built using prior knowledge and experience.

iv. Trust is a subjective notion based on opinion and values of an individual.

v. Trust is dynamic and new knowledge and experience will be overriding over the old ones.

vi. Trust is context-dependent.

vii. Trust is multi-faceted.

The major components that are affected by the barriers in cloud trust as explained by Ko et al. (2011); Khan and Malluhi (2010) are:

Security: Mechanisms (e.g. Encryption) which make it difficult or uneconomical for an unauthorized person to access some information.

Privacy: Protection against the exposure or leakage of personal or confidential data (e.g. Personally identifiable information (PII)).

Accountability: Defined as the obligation and or willingness to demonstrate and take responsibility for performance in light of agreed-upon expectations. Accountability goes beyond the responsibility by obligating an organization to be answerable for its actions.

Auditability: The relative ease of auditing a system or an environment. Low degree auditability means that the system has poorly-maintained (or non-existent) records and systems that enable efficient auditing of processes within the cloud.

Various technologies that are used to establish trust by the cloud providers are as given below Habib et al. (2011):

Service Level Agreement: Currently the way to build trust between cloud providers and cloud consumers is the execution of SLAs. SLA validation and monitoring schemes are used to quantify what exactly a cloud provider is offering and which assurances are actually convened. In cloud computing environments, customers are more responsible for

monitoring SLA violations and informing the providers for compensation.

Auditing: Cloud providers use different auditing standards to assure cloud users about the different services offered by them and various platforms. These rules are not adequate enough to reduce the users security concerns. Due to the unwillingness of the cloud service providers to reveal the internal audits transparency in cloud poses to be a problem.

Ratings & Measurements: Cloud providers are rated based on a questionnaire that needs to be filled in by current cloud consumers.Cloud Commons aims to combine consumer feedback with technical measurements for assessing and comparing the trustworthiness of cloud providers.

Self-assessment Questionnaires: The Cloud Security Alliance (CSA) proposed a detailed questionnaire for providing security control transparency called the Consensus Assessment Initiative (CAI) questionnaire. This questionnaire is a methodological way for assessing the capabilities and competence of cloud providers in terms of different attributes like compliance, information security, and governance.

2.2.4 Applying Trust in Cloud Computing

In order to gain trust on CSPs, transparency and accountability play an important role. Security, Trust & Assurance Registry (STAR) is a free publicly accessible registry program which is launched by Cloud Security Alliance (CSA) to increase the cloud transparency. This program helps CSPs to publish self-assessment of their security controls, in either a Consensus Assessments Initiative Questionnaire (CAIQ) or a Cloud Controls Matrix (CCM). CAIQ contains over 140 frequent questions that are useful for cloud users or auditors. CCM is a framework describing how a CSP aligns with the CSA security guide. STAR is a useful source for users who are seeking for cloud services and the information offered is a CSPs self -assessment. Cloud Trust Protocol (CTP), is a request-response mechanism for a cloud user to obtain specific information about the elements of transparency which includes aspects of configuration, vulnerability, audit log, service management, service statistics, and so forth and are applied to a specific CSP. The primary purpose of the CTP and the elements of transparency is to generate evidence-based confidence that everything that is claimed to be happening in the cloud is indeed happening as described, and nothing else. CTP helps user internal observations of cloud service operations by pro-

viding an interesting channel between cloud users and CSPs. One of the weaknesses of STAR and CTP is that its information is provided by CSP itself and if CSP is dishonest, it can filter out or change data which would violate data reliability based on trust judgment.

Trust is an important concept in distributed computing environments and plays a critical role in ensuring and enhancing system security and adaptability. There are a number of challenges that affect trust in different distributed systems for example in social networks, hackers are a serious threat since they try to access the user accounts and use them as a trusted source to spread malware. Also, tools that help in managing, viewing, querying, transferring and displaying personal data in the system and provide searching and mining profile data can be considered as another trust issue in social networks. In P2P networks, security implications arise from abusing trust between peers. One of the trust issues exists in peer to peer networks is the distributed denial of service (DoS). Also, attackers can make use of the querying nature of P2P networks to overload the network by sending a massive number of queries to peers, make the portions of the network inoperable. Since in P2P network the peers should contribute in resource distribution process, peers data stream may be compromised by fellow peers who assist in transmitting the data in the system, and sometimes free to freeload off other peers.

Cloud Computing supports four deployment models which are public clouds, private clouds, community clouds and hybrid clouds. In a private cloud, trust is not applicable if the third party is not involved. However, public clouds can introduce many security risks since controlling data in this deployment model is very challenging. Trust in community cloud depends on the role of the third party. If there is a third party involved, the trust risks are the same as the corresponding case in a private cloud. Otherwise, if the community cloud is managed by the organizations in the community, trust risks are limited to the trust relationships that are discussed and agreed between community members. In Hybrid clouds, since both the private clouds and public clouds are involved, all the trust issues related to public cloud shift to the hybrid cloud, too.

2.3 Summary

Security becomes vulnerable when the trust on the system is low. So trust places an important role in the security aspect. Trust management is the way trust is organised so that there is no detrimental effect on any of the entities of the system. Trust can be adapted in cloud in maintaining the privacy of user credentials or data.Trust is also developed by a feedback mechanism in a system. When a trust is evaluated, also with it risk factor is calculated. Trust helps in minimising the risks of the system. Thus trust plays an important role in a cloud environment. Management of this trust in appropriate areas of cloud reduces the security to some extent.

Chapter 3

LITERATURE REVIEW

In this chapter, we describe all the trust models developed by various researchers so far. We also highlight the various disadvantages of all the trust models. At the end, we summarize the models with advantages and disadvantages in a table and throw focus on the issue of our research.

3.1 Trust as Cloud Security

Security is one of the most important areas to be handled in the emerging area of cloud computing. If the security is not handled properly, the entire area of cloud computing would fail as cloud computing mainly involves managing personal sensitive information in a public network. Also, security from the service providers point also becomes imperative in order to protect the network, the resources in order to improve the robustness and reliability of those resources.

Trust is one of the important aspect of security. Trust is a mental state comprising of expectancy of the trustor of a non detrimental behaviour from the trustee, the belief of trustor on the trustees competence, integrity, and goodwill based on some evidence and the willingness of the trustor to take risk based on that belief. In the current scenario, trust is calculated based on reputation or SLA verification or Cloud transparency or by formal audits of the service by a third party. Trust by reputation is useful to create initial trust, but fails on later stages. Trust by SLA verification builds trust for the services visible to users but fails to build trust for invisible factors like privacy. Trust by transparency is built by service providers themselves so trusting them is a major concern.

So a strong trust is needed for the user to trust the services of the service provider. Multiple trust models are developed based on the requirement to cater to the needs of the

user to trust the cloud services. Our next section 3.2 throws light on the various trust models.

3.2 Trust Based Approaches in Cloud Computing

Trust is an eminent factor in any network. Trust management comprises collecting the information necessary to establish a trust relationship and to dynamically supervise the existing trust relationship. The various models for describing trust and trust establishment in Cloud Environment are listed below.

Khan and Malluhi (2010) have analyzed the trust in the cloud system in terms of security and privacy. These authors have identified control, ownership, prevention and security as the key aspects that decide users level of trust on services. Diminishing control and lack of transparency are identified as the issues that diminishes the user trust on cloud systems. These authors have forecast that remote access control of the resources, transparency in cloud provider's actions and providing security for users would enhance the trust of users in the services and service providers.

Yu et al. (2006) have discussed the security issues that SaaS might create on user data as the remotely installed software will have unrestricted access to the user data. These authors have presented a mechanism to separate software from the data so that it is possible to create a trusted binding between the entities. The mechanism involves four parties, namely the resource provider, software provider, data provider and the coordinator. The resource provider hosts both data and software and provides the platform to execute the software for data. The software provider and data provider are the owners of the software and data respectively. The coordinator brings the other parties together while provider facilitates the ancillary services such as searching for resources and providing an interface to execute the application on the data. These authors do not address the question of trust on the proposed platform as this would be another application or service hosted on the cloud. Both application providers and data provider need some kind of better assurance as now they are entrusting their data and software to a third party software.

Sato et al. (2010) have proposed a trust model of cloud security in terms of social security. The social security is divided into three sub areas, namely; multiple stakeholder problem, open space security problem, and mission critical data handling problem. The

multiple stakeholders problem addresses the issues created by multiple parties like the client, the cloud service providers, and third parties. The client assigns the operations to cloud providers as written in the Service Level Agreement (SLA). Even if the client would like to have the same type of policies that it would apply if the resources were hosted on site on the delegated resources, but the provider's policy may differ from that of the client. The providers are bound only by the SLA signed between the parties. The SLA plays the role of glue between the policies. The open space security problem addresses the issue of loss of control on where the data is stored and how they are physically managed once control of data is delegated to the cloud provider. The mission critical data handling problem looks at the issue of delegating the control of mission critical data to a service provider.

Researchers have developed a trust model named cloud trust model to address the problems. Two trust layers named internal trust layer and contracted trust layer have been added to the conventional trust architecture. Internal trust layer acts as the platform to build the entire trust architecture. Internal trust layer is installed in the in house facilities and hence under the control of the local administration. Id and key management are handled under the internal trust. Also any data that is considered critical or needs extra security must be stored under this layer. Contracted trust has been defined as the trust enforced by an agreement. A cloud provider gives the trust to a client based on the contract that is made up of three documents known as Service Policy/Service Practice Statement (SP/SPS), Id Policy/Id Practice Statement (IdP/IdPS) and the contract. A cloud system, thus installed is called a secure cloud by the authors.

Li et al. (2010a) proposed a domain-based trust model to ensure the security and interoperability of cloud and cross-cloud environment. They also suggested some trust based security strategies for the safety of cloud customers and providers.

The family gene based cloud trust model proposed by Wang et al. (2010b,c) is basically based on the study of various basic operations such as user authentication, authorization management and access control, and proposed a Family-gene Based model for Cloud Trust (FBCT) integrating these operations.

CARE resource broker integrated trust model proposed by Manuel et al. (2009) calcu-

lates trust based on three components, namely, Security Level Evaluator, Feedback Evaluator and Reputation Trust Evaluator. Security level evaluation is carried out based on authentication type, authorization type and self-security competence mechanism. Feedback evaluation has three different stages, namely feedback collection, feedback verification and feedback updating. The reputation trust evaluator computes the trust values of the cloud resources based on the capabilities of computational parameters and network parameters.

Shen et al. (2010); Shen and Tong (2010) have proposed a system of integrating Trusted Computing Platform (TCP) into the cloud computing system which improves the security and dependability of cloud. The TCP is used in authentication, confidentiality and integrity in a cloud computing environment. The model has been developed as software middleware known as the Trusted Platform Software Stack (TSS).

SLA based trust model proposed by Alhamad et al. (2010) consists of the SLA agents, cloud consumer module, and cloud services directory. The SLA agent groups the consumers to classes based on their requests, designs SLA metrics, negotiates with cloud providers. Cloud consumer module requests the execution of services. Cloud services directory advertises the cloud providers services and helps consumers find the appropriate providers. The authors have proposed only the model and no implementation or evaluation has been developed or described. Hence the each and every module will have to be evaluated for their functionality and the effectiveness and finally the overall model will have to be evaluated for its effectiveness.

Multi-tenancy trusted computing environment model (MTCEM) proposed by Li et al. (2010b) is a two-level hierarchy which supports the security duty separation and also supports three types of stakeholders namely, CSP, customers and auditors. CSP responsibility is to keep infrastructures trusted while the customer assumes responsibility starting from the guest OS, which are installed by the customer on the Virtual Machines provided by the CSP. The auditor monitors the services provided by the CSP.

Yang et al. (2010) study states that the existing trust models ignore the existence of a firewall in a network. These authors have proposed a firewall based trust model in the Cloud. Their paper gives the detailed design calculations of the proposed trust model and practical algorithms of measuring and updating the value of dynamic trust. The model has

the following advantages compared to other models:

i. There are different security policies for different domains.

ii. The model considers the transaction context, the historical data of entity influences and the measurement of trust value dynamically.

iii. The trust model is compatible with the firewall and does not break the firewall's local control policies.

Watermark-aware trusted environment model proposed by Fu et al. (2010) is made up of two components, namely the administrative center and the cloud server environment. The administrative center inserts watermark and tailors the Java Virtual Machines (JVM) and the trusted server platform includes a series of cloud servers deployed with the customized JVMs and is used to handle security due to running software on a cloud.

Ranchal et al. (2010) have proposed a system without the involvement of a trusted third party based on the study conducted on identity management in the cloud. The proposed system is based on the use of predicates over encrypted data and multi-party computing.

Security framework model proposed by Takabi et al. (2010) consists of three main entities, namely cloud customers, service integrators and service providers. The Service Integrator acts like a bridge between the customers and service providers. The Service Integrator module consists of security management module, trust management module, service management module and heterogeneity management module. The heterogeneity management module manages the heterogeneity among the service providers. In overall this is a very comprehensive framework. But the authors have not discussed the interoperability issue of each component in the framework or implemented a prototype to evaluate the function and efficiency of the components or the overall framework.

A reputation system based on a fuzzy-logic was developed by Song et al. (2005) which has the ability to handle uncertainty, fuzziness, and incomplete information. The proposed system uses fuzzy logic inference rules to calculate local trust scores and to compute global reputation.

Filali and Yagoubi (2015) developed a general trust model based on QoS selection and Certain Trust Model which uses QoS parameters like direct trust, user feedback, user preference, etc. to calculate trust of the service provider.

Sun et al. (2011) proposed Trust Management Model based on fuzzy set theory called TMFC where direct trust was classified into two types due to difference in their trust assessment.

Mohammadnia and Shakeri (2014) proposed HITCloud model to handle some of the security issues of the cloud like data integrity, privacy using a feedback mechanism. The feedback from users will be filtered according to their reliability and accuracy of accomplishment, which in turn will be calculated based on node trust and region trust.

Fan and Perros (2013) proposed Reliability-based Trust Management for Cloud Services which is based on the feedback. Users who have no prior experience with service provider can submit their trust feedback to the trust management system to make a decision to use the service of the provider or not. The feedback from users is filtered according to their reliability, which is calculated based on familiarity and consistency.

Habib et al. (2011) proposed multi-faceted trust management architecture for selecting appropriate cloud service providers based on a calculated trust which in turn is dependent on the customer attribute value.

Muchahari and Sinha (2012) proposed a new trust management architecture which consists of cloud service registry and discovery which helps to register and locate a service provider based on the three service models namely, Infrastructure-as-a-service, Platform-as-a-service, Software-as-a-service. Based on the different trust values of the models the selection of service providers is done.

Zhao et al. (2013) proposed pool oriented resource trust management which calculates the trust of pool of resources that would be used for services. The trust thus calculated is verified and a protocol is developed to initiate communication between the resources in the pool.

Bennani et al. (2014) proposed a trust based solution to evaluate the Hybrid service model for data credibility. They have proposed two algorithms which handle trust evaluation for both private cloud and public cloud.

Anisetti et al. (2014) have proposed certification based trust model to handle assurance techniques which can manage trust information during production and also handle the third party trust to manage the entire assurance technique.

Firdhous et al. (2012b) have proposed a hysteresis based robust trust computing mechanism that computes trust value using a non-linear equation which has more than one state at a given time.
Firdhous et al. (2011b) have proposed a trust system based on server response time where the trust computation score lies between -1 and 1 for different levels of services in terms of response time and confidence levels.

Wang et al. (2013) have proposed a group signature based trust management for Iaas cloud model. The proposed architecture helps to the resource pool oriented trust management in a cloud infrastructure. A protocol is also devised to synchronize the interaction and behavior of trusted resources.

Firdhous et al. (2011a) have proposed a trust system based on the response time. The trust system computes a trust between 0 and 1 for different levels of services and continues to improvise the calculated trust values based on the performance of the system.

3.3 Research Gaps and Motivation

An extensive literature survey reveals some of the drawbacks found in the various trust models explained in the above section. The issues are listed as below:

Trust calculated by Sato et al. (2010) proposed model is internal to the organization. The Cloud Service Provider (CSP) has nothing to do with the security of the resources. So the organization has to have a private cloud to secure its data which is not possible with small/medium organizations.

The Family Gene based trust models proposed by Wang et al. (2010b,c), is just proposed for authentication and is tested by simulation. The model does not deal with security aspects either of data or of resources. A real time implementation is not done.

In CARE resource model proposed by Manuel et al. (2009), the conventional scheduling is done through FIFO. So computation/process starves for the necessary resources. The priority of resources for the critical jobs is not taken care.

Shen et al. (2010); Shen and Tong (2010) have proposed trusted computing technology for trust evaluation. The basic disadvantage of this model is that the underlying architecture is based on Trusted Computing Platform [TCP] which is difficult to integrate in cloud computing with respect to hardware.

Alhamad et al. (2010) have proposed SLA based trust model and no implementation or evaluation has been developed or described. This model is a reputation based trust that has a disadvantage that the user with high scores for reputation can cheat user in fewer transactions even though they receive negative feedback. This model has a centralized architecture, so all the services and reputation information has a single point of failure.

In the Role Based Trust model the trust is based on the roles, ID used for TCP, standard certificate for assurance. The hardware maintains a master key for each machine and it uses master keys to generate unique sub key for every configuration of the machine. The data encrypted for one configuration cannot be decrypted in another configuration of the same machine. If the configuration of the machine changes the session key of the local machine will not be useful.

The Active Bundle Scheme proposed by Ranchal et al. (2010) is based on Identity Management model approach which is independent of a third party, it is less prone to attack as it reduces the risk of correlation attacks and side channel attacks, but it is prone to a denial of service as an active bundle may also be not executed at all in the remote host.

From literature survey, it is very clearly known that researchers have till now not considered the availability/non-availability of resources for any transaction. Thus a strong Trust model is needed to calculate Trust in Cloud Environment based on the availability of resources as the resources are the main basis for any transaction in Cloud.

From Table 3.1 it is very clearly known that researchers have till now not considered the availability/non-availability of resources for any transaction. Thus a strong Trust model is needed to calculate Trust in Cloud Environment based on the availability of resources as

Table 3.1 Comparative summary of previous work done in the Cloud

Authors	**Type**	**Identity Mgnt/ Authentication**	**Data Security**	**SLA Support**	**Heterogeneous System Support**	**Implemented**	**Comments**
Sato et al. (2010)	Social Security Based	Discussed	Discussed	Discussed	No	No	No concrete proposal. Only discussed the issues.
Wang et al. (2010b,c)	Family Gene Based	Discussed	No	No	No	No	Model has been tested using simulation
Manuel et al. (2009)	Integrated with CARE Resource Broker	Yes	Yes	No	Yes	No	Model has been tested using simulation
Shen et al. (2010); Shen and Tong (2010)	Built on trusted platform service	Yes	Yes	No	Yes	No	Only a model has been proposed.
Li et al. (2010b)	Built on Trusted Computing Platform	No	No	No	No	Prototype Implemented	Concept has been proved with a prototype.
Yang et al. (2010)	A collaborative trust module of Firewall-through	No	No	No	Yes	No	Model has been tested using simulation.
Fu et al. (2010)	Watermark based security	No	No	No	No	Prototype Implemented	Concept has been proved with a prototype.
Ranchal et al. (2010)	Based on active bundles scheme	Yes	No	No	Yes	Prototype Implemented	Concept has been proved with a prototype

the resources are the main basis for any transaction is Cloud. Hence a new Trust Model is proposed to handle some of the problems.

3.4 Summary

Trust is the first step towards security of any resource in cloud environment. In this chapter we have thrown light on several trust models developed by various researchers for calculating the Trust. This chapter also highlights the disadvantages of the various models discussed. The table 3.1 throws a light on various models based on some important parameters like authentication, Data Security, SLA support and so on.

Chapter 4

CONCEPTUAL FRAMEWORK OF TRUST MODEL

In the previous chapter, we discussed about some of the disadvantages of the various trust models and also highlighted the issue of our research. In this chapter, we describe a framework for trust evaluation in a diagrammatic form.

A strong trust ensures the security of the resources as well as authentication of the users. A strong trust enhances a secure path between the user and the resources. To have this strong trust path a model depicting the trust path between the users and the resources of the service provider is needed. In below section, we have proposed a model which represents a strong trust path for the utility of the resources by the user.

4.1 Genesis of the Conceptual Framework

From the literature review, it is very well known that though many trust models were developed still the models were prevalent to only certain scenarios. The models didn't support heterogeneous environment.Many models like: MCTEM model which supported duty separation, SLA based model which classified the users based on their requests, CARE resource broker model which calculated trust in three different types, Watermark-aware model which used Java Virtual Machines to insert a watermark, Reputation system based on fuzzy logic which incorporated fuzzy inference rules to calculate reputation, Trusted Computing model which improved the dependability on cloud were proposed and developed to calculate trust based on multiple factors like feedback, inference rules,security strategies, reputation and so on.

But the basic disadvantage of these models were, they were platform and application dependent. They did not support heterogeneity. The models also never spoke of how trust can be established to secure the resources of the service provider for utility of user based on availability. A strong trust on the resources of the service provider would always enhance the trust of the user on the service provider. To build a strong trust, a strong trust path is needed between the two important entities of the cloud namely: user and resource of the service provider. The below section explains the way to build a strong trust in terms of trust path between the user and resources of the service provider.

4.2 Conceptual Model and its Working

From the above section, we have understood that no model is proposed to build a strong trust path between the user and the resources. A strong path envisages the security in terms of authentication of the user and resource security based on availability for utility.Below we propose a diagrammatic representation of the proposed model which depicts the trust path between the entities.

The Figure 4.1 is conceptual diagram of the proposed model. The diagram gives a pictorial representation of Trust Management between cloud user and cloud service provider.The diagram has components like Trust Admin, Trust Feedback, Dynamic Trust Calculator, Trust Selection Algorithm, Trust Calculative Model, Effective Trust Value and Trust Moderator. The detailed description of these components is given below:

Trust Admin: Basic Trust value is needed to enter any domain. Trust Admin initializes the minimum/basic trust required to enter the system using the SLA. Trust Admin also administers the aver-all basic trust of the user/ resources in the system. Trust Admin examines the basic trust value needed by the user to request for resources and the basic trust value of the resources to be allocated to the requested user.In migration, Trust Admin plays an important role in calculating the basic dynamic trust of the user/resources using the Trust Feedback component.

Trust Calculative Model: Many Trust models are developed, but are always very spe-

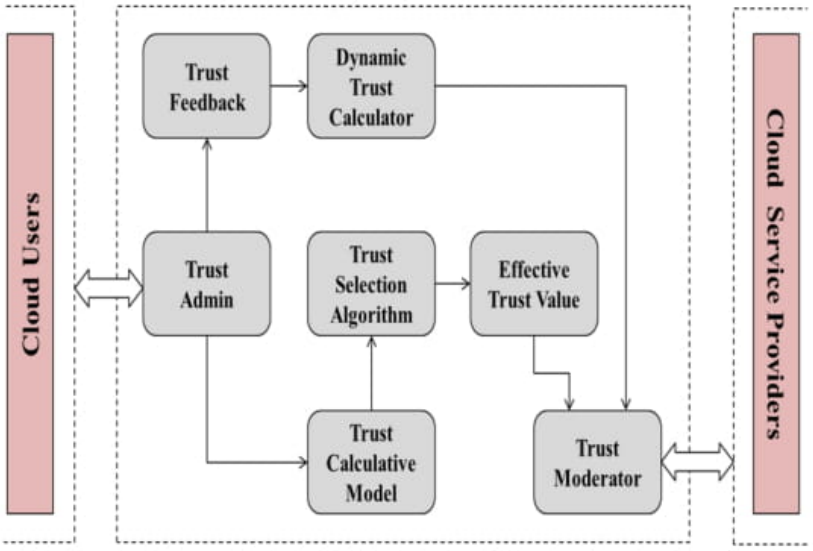

Figure 4.1 Conceptual Diagram of proposed Trust Model

cific with respect to a scenario.Trust models are needed to calculate the trust value required by the user and the trust value for resources of CSP for a successful completion of the transaction. Trust models are mathematical models enabling the user/CSP for calculating the trust which generally ranges between 0 & 1. We have proposed a mathematical model to calculate the trust value by entropy and also dynamic trust value used for continuous evaluation of the resources and user of the system. Our trust model is a generic model.The detailed description is given in Chapter 5.

Trust selection Algorithm: The mathematical values derived from the mathematical model ae to be evaluated for a real scenario to check the correctness of the proposed model. So a selection algorithm is used to justify the trust values generated by the mathematical model.

It is known from the researcher Noraini and Geraghty (2011) that Genetic Algorithm is the best selection algorithm which gives near optimal solutions which are suitable for any practical problems where input data are approximate, but the basic disadvantage of genetic algorithm is it does not yield exact optimal solutions when the population size is considerably large. So the algorithm used for trust selection is the Family Gene Genetic Algorithm as suggested by Jianhua et al. (2006), which gives the best optimal solution for a large

population.The detailed description is given in the Chapter 6.

Effective Trust Value: The trust values generated by Trust Calculative Model and justified by Trust Selection Algorithm are finalized here so as to assign the trust value required to allocate resources based on availability for the transactions to be performed by the customer/user. Once the thus calculated trust is assigned, the user can access the resources that are termed available due to the trust value for any transaction as required by the user.

Trust Moderator: Trust Moderator assigns the final trust value to the existing customer who has requested for the specific resources and also assigns trust during migration of the customer.

Trust Feedback: The basic trust value is stored in a central table called Trust Feedback for future considerations. In a federated cloud environment, when the user wants to migrate to a different CSP, rather that reevaluating the new trust of the user, the basic initial trust from the central table is retrieved. Then a new trust value is calculated using the mathematical proposed for migration and thus the basic initial trust value for the user is assigned with the new CSP for further functioning.

Dynamic Trust Calculator: In case of migration the new trust value is calculated using the Principle-3 of the Trust Calculative Model and the new trust of migration is sent to the Trust Moderator for the assignment of the trust.

4.2.1 Working Process of the Proposed Trust Model

In the above section we have described the proposed Conceptual Diagram of Trust model developed. The working process between the components in the proposed Conceptual Diagram is as shown below:

The Figure 4.2 is a sequence diagram which describes the sequences of steps performed in calculating trust by a service provider to processes the request of the customer for available resources to complete his transaction. Trust Admin defines the initial trust required to enter the system. Trust calculative model incorporates the mathematical model proposed

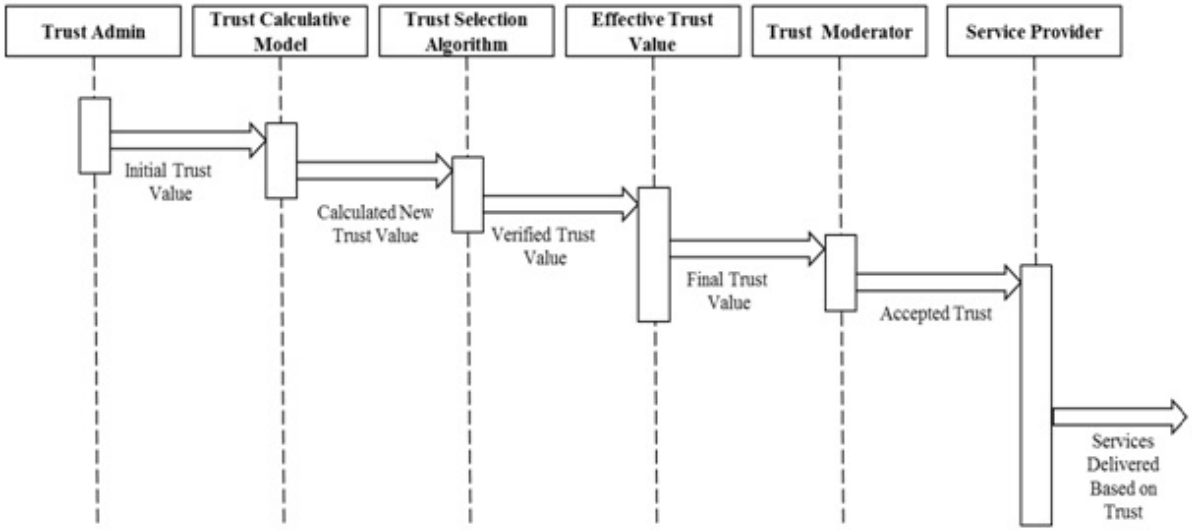

Figure 4.2 Working Process of the Proposed Trust Model

and generates trust values. Trust selection algorithm justifies the trust values generated by the trust calculative model and then these trust values are sent to effective trust value. In effective Trust Value based on the scenario the final trust value is generated and sent to Trust Moderator who in turn assigns the trust value for the customer and resource. Based on the trust value assigned by the Trust Moderator the Service provider processes the request of the customer for resources for the transaction. Trust Selection Algorithm Uses Family Gene Genetic Algorithm to justify the trust calculated by the Trust Calculative Model. Effective Trust Value finalizes the final trust value. This final trust value is assigned/passed on to the Service provider by the Trust Moderator who in turn assigns the resource to the requested customer based on this final trust value. In case of migration the trust value stored retrieved by the Trust Feedback and this is sent to the Dynamic Trust Calculator who in turn calculates the dynamic trust and the sends it to the Trust Moderator for final assignment of Trust by CSP.

4.3 Analysis and Results

Various trust models were developed based on reputation, watermarking schemes, feedback and so on. Mathematical models based on semantic methods, formal languages, etc. were developed to calculate the trust of the entities. In all the other proposed models the

trust was calculated to secure data, or authenticate the user, or to handle privacy, but failed to secure the resources of the service provider which are the essential entities of the cloud environment. None of the proposed models were implemented on a real time cloud platform to speak about the performance of the system.

We have proposed a model to build trust on the system based on the availability of the resources. If the resources are available for utility then the trust on the system increases as more computations are undertaken by the user. If the resource is not available the trust of the system decreases thus depicting that the resources are insufficient.Thus a strong trust path between the entities of the cloud represents that the requested resources by the user are available for his further computations which in turn increases the trust of the user on the service provider.

4.4 Summary

We have seen that various issues exist, though various trust models are developed. One of the issues being the security of the resources of the service provider. We have proposed a framework that acts as a flow chart which describes the sequence of steps that would be taken to build a strong trust path between the two entities of the cloud, namely: user and resource.This strong trust path enhances the security of the resources of the CSP as well as the trust of the user on the resources of CSP for further computation.

Chapter 5

MATHEMATICAL MODEL FOR TRUST IN CLOUD ENVIRONMENT

In the previous chapter, we have seen how the framework describes the path to be followed in trust evaluation to build a strong trust path which enables a strong trust based security of resources and user. In this chapter, we describe in detail the mathematical model needed to calculate the trust. Here we also focus on the calculation of dynamic trust and trust for migration.

5.1 Mathematical Model for Trust

We have found from the literature survey that though much research towards trust management in cloud has taken place, still the trust of resources based on their availability was never considered. So we propose to calculate the trust value based on usage values which in turn are calculated in terms of availability and non-availability.

The trust value calculation as proposed by Divakarla and Chandrasekaran (2016) is briefly described as below:

i. Trust relationship established between two entities is based on usage and the entities are represented as customer and resources. The notation for the relationship is given as {Customer: resource, usage}

ii. Trust is a collaboration of certainty and uncertainty. If the resource is available it is allocated to the customer and the customer performs the action else if not available the

trust of the customer on the resource is minimized.

iii. The degree of the trust can be represented by a real number called trust Value. Trust value represents availability/non-availability.

iv. Customer may have a variation of trust values based on the availability of the resources.

Principle-1: Trust Using Entropy

Thus, by the basic understanding of the trust, we further define the trust value based on usage. If the trust value is calculated based on availability that the resource is allocated for the customer for his action to be performed, then T{customer: resource, usage} denotes that the trust of the customer on the requested resource is based on the availability/non-availability. Then the probability P{customer: resource, availability} will be the availability of the resource to the customer for some action to be performed. Using entropy model as explained by Cover and Thomas (2012) of the Information theory the new trust value, thus defined is as below:

$$T\{customer : resource, usage\} = \begin{cases} 1 - H(p), & for \quad 0.5 < p < 1 \\ H(p) - 1, & for \quad 0 < p < 0.5 \end{cases} \tag{5.1}$$

Where,

H(p)=-plog$_2$(p)-(1-p)log$_2$(1-p) and p=P{customer: resource, usage}. When p=1 the customer is allocated the available resources and the trust value is high. When h=0 the customer is not allocated the resources due to unavailability and the trust is very low.

Example: Let us consider an example. In the first instance, let the probability value be increased from 0.5 to 0.508 and in the second instance, let the probability value be increased from 0.98 to 0.988. The probability value increases by the same amount in both cases, but the trust value increases by 0.00018 in the first case and 0.0177 in the second case. Thus it is understood that the trust value is not a linear function of probability.

Proof: Let us assume that there are three nodes (X,Y,Z) in linear chain as shown in Figure 5.1 and node B observes behavior of node C and makes recommendations to node A as T_{YZ} ={Y:Z, usage}. Node X trusts node Y with T{X: Y, on feedback}= F_{XY}.

Figure 5.1 Linear Chain

Therefore

$$T_{XYZ} = F_{XY} * T_{YZ} \tag{5.2}$$

If node Y has no clue about node Z then T_{YZ}=0 or if node X has no clue about node Y then T_{XY}=0, then the trust between X and Y will be zero i.e., T_{XYZ}=0. From equation (5.2) it is understood that trust increases or decreases with increase or decrease in feedback.

Principle-2: Dynamic Direct Trust Value

Evaluating Trust in dynamic cloud environment is a necessary factor as cloud is dynamic in nature. Here trust in calculated based on the number of successful transactions made so as to take into account the availability of resources for every successful transaction.

Initial Trust i_t is calculated as :

$$i_t = (r_{it} * c_{it}) \tag{5.3}$$

where r_{it} is initial resource trust value

c_{it} is initial customer trust value.

After successful transactions, the new trust value will be

$$D_t = i_t + \sum_{i=0}^{i=n} ti/totalNo.oftransactions \tag{5.4}$$

Where t_i is No. of successful transactions and D_t must always be greater than the initial trust value as i_t is the initial trust required to perform any transaction.

Example: Let us assume that the initial trust of the resource r_n be 0.1 and initial trust of customer c_n be 0.1 as initial trust cannot be zero to enter the system and consider one successful transaction for every 10 transactions done by the customer.

From equation (5.3) and equation (5.4) if

i_t= 0.01 then d_t = 0.11 where t_i=1 and total no. of transactions=10

i_t=0.01 then d_t = 0.16 where t_i=6 and total no. of transactions=10

As the number of successful transactions increase the dynamic trust value also increases. If the No. of successful transactions is zero then the dynamic trust will be equal to initial trust. Thus any increase in successful transactions increases the trust.

Proof: Let us assume a multiple network as shown in Figure 5.2. Here the node P can have trust on node S as T_{PS} in two paths P-Q-S or P-R-S.

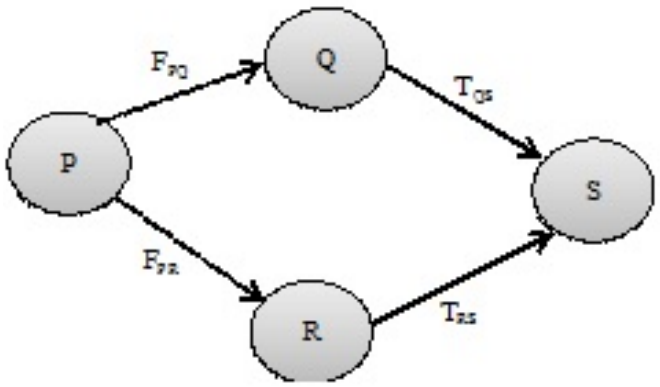

Figure 5.2 Multiple Network

Let F_{PQ}= T{P:Q, on feedback} and F_{PR} = T{P:R, on feedback}. Using the maximal ratio combining theorem we get:

$$T\{P:S,Usage\} = \omega 1(F_{PQ} * T_{QS}) + \omega 2(F_{PR} * T_{RS}) \tag{5.5}$$

where

$$\omega 1 = F_{PQ}/(F_{PQ} + F_{PR}) \tag{5.6}$$

$$\omega 2 = F_{PR}/(F_{PQ} + F_{PR}) \tag{5.7}$$

Here, if any of the paths have trust zero, the end value of the trust is not affected, that is it tends to initial trust. Thus we can conclude that if any transaction fails the dynamic trust value becomes equal to initial trust value.

Principle-3: Trust Value for Migration

The new trust value calculated for the customer on resources is stored in central table which can be retrieved by all Cloud Service providers (CSP).

When a customer wants to migrate to a different service provider the initial trust of the customer with the new CSP is calculated as :

$$M_t = (i_t + D_t)/0.5 \tag{5.8}$$

Where,

0< Mt for availability

Mt < 0 for non-availability

i_t is initial trust by principle-1

D_t is the dynamic trust by Principle-2

0.5 is the minimum trust required by any entity for a successful transaction.

5.2 Result and Analysis

Our Mathematical Model was simulated using MATLAB software for our inference. The simulation results are as explained below:

The Figure 5.3 shows that for every increase in probability value due to Entropy the trust decreases and at a particular point of probability value 0.5 the trust becomes zero and again increases with increase in probability value. This shows that the trust increases with increase in probability.

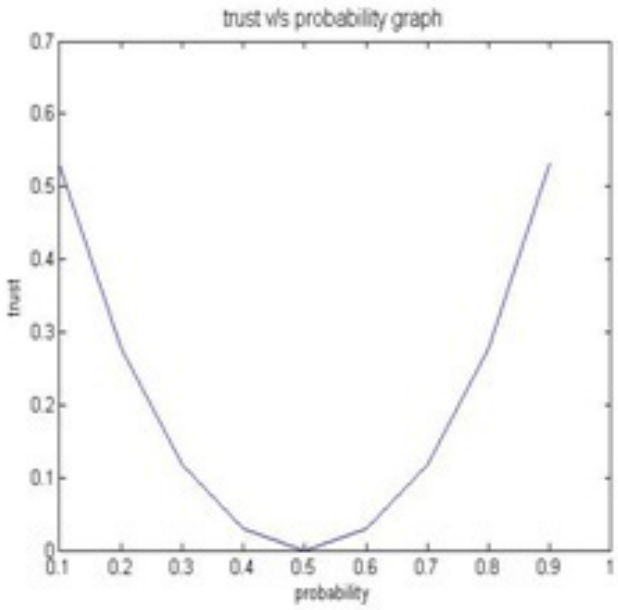

Figure 5.3 1-H(p) values

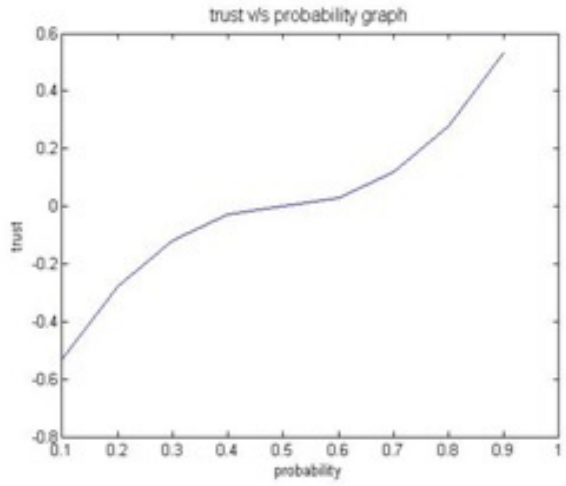

Figure 5.4 H(p)-1 values

Figure 5.4 shows that with at a threshold of probability value 0.5 the trust starts increasing to positive which clearly indicates the availability of resources due to increase in the trust value. Thus the principle-1 shows that with every availability of resources the trust increases.

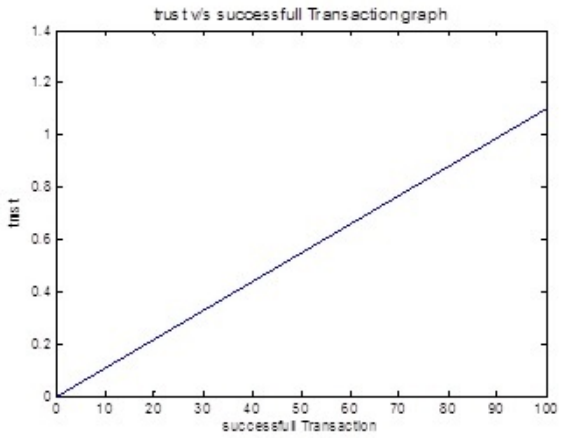

Figure 5.5 Dynamic trust after every successful transaction

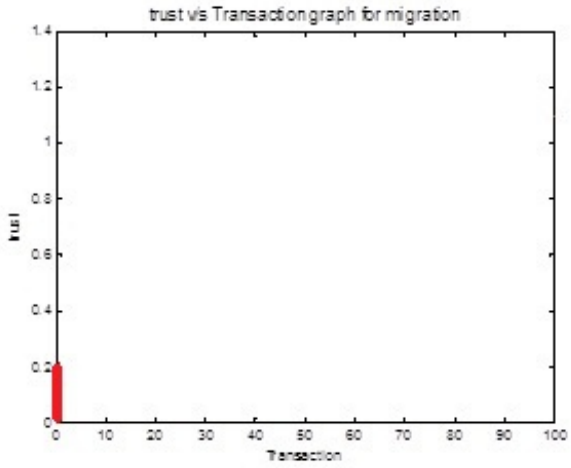

Figure 5.6 Dynamic trust after every unsuccessful transaction

To check the migration trust value we conducted an experiment using oracle database where we had some set of users and some random files were allocated to them. We allowed the users to perform minimum 100 transactions on these files. We considered in our experiment these minimum 100 transactions to check whether the trust value changes due to these transactions.

The Figure 5.5 shows that with every successful transaction the trust increases. This shows the availability of resources for the successful transactions which in turn increase the trust value. The graph shows that trust value increases with every successful transaction; we have considered the upper threshold of trust value as 1. Any increase in trust value

above 1 is considered as trust value 1.

Figure 5.6 depicts that for every unsuccessful transaction the initial trust value drops down to 0.2 but for every successful transaction the trust increases. Our experiments show that the minimal trust required for requesting resources for the transaction to happen is 0.2. So to calculate the dynamic trust again after every unsuccessful transaction the minimum trust required is 0.2 whereas minimum trust required for every successful transaction is 0.5.

5.3 Summary

In this chapter we have described 3 different ways to calculate trust. Trust calculation using entropy helps in building the initial trust of the resources based upon the availability and also user. Upon multiple transactions by the user, the Dynamic trust helps in calculating the new trust of the user and resources in the dynamic cloud environment for additional request of resources. In a federated cloud environment the trust by Migration helps in calculating the new trust of the user when he wants to migrate from one cloud to another. Our experiments to evaluate the trust principles help us to understand the correctness of the developed models.

Chapter 6

EVALUATION OF TRUST MODEL USING OPTIMAL SOLUTION ALGORITHM

In the previous chapter, we have defined different ways to calculate trust in a cloud environment. In this chapter, we bring to you a new modified optimal solution algorithm known as Family Gene Genetic Algorithm which gives an optimal solution when the population size is considerably large.

6.1 Selection of Resources in Cloud

Every optimization problem is NP-Hard problem. An Optimal solution is found for every optimization problem.An optimal solution is a feasible solution which maximizes the objective function. To compute an optimal solution we use multiple problem-solving methods like dynamic programming, meta heuristic, simulated annealing, etc.

Dynamic programming is solving an optimization problem by caching sub problems rather than recomputing. Meta heuristic is a high-level strategy for other heuristics to get a feasible solution. Simulated healing is to find a proper solution to optimization problem by trying random variants of the current solution. In the next successive section-6.2 we speak in detail about a dynamic programming algorithm called Genetic Algorithm.

6.2 Optimization in Selection of Resources

A genetic algorithm (GA) Noraini and Geraghty (2011) is a method for solving both constrained and unconstrained optimization problems based on a natural selection process that

depicts biological evolution.

The algorithm repeatedly modifies a population of individual solutions. At each step, the genetic algorithm randomly selects individuals from the current population and uses them as parents to produce the children for the next generation. Over successive generations, the population evolves toward an optimal solution.

One can apply the genetic algorithm to solve problems that are not well suited for standard optimization algorithms, including problems in which the objective function is discontinuous, nondifferentiable, stochastic, or highly nonlinear.

When GA is used to solve optimization problems, good results are obtained quite quickly. Genetic algorithms are guided random search and one of the most popular optimization techniques among evolutionary algorithms for multi-objective optimization problems. To use a genetic algorithm, it is required to represent the solution of the problem as a genome (or chromosome). The genetic algorithm then creates a population of solutions and applies genetic operators such as mutation and crossover to evolve the solutions in order to find the best one. These operate on a population of potential solutions, applying the principle of survival of the fittest to generate improved estimations to a solution.

The working principle of a standard GA is illustrated in Algorithm 1. The major steps involved are the generation of a population of solutions, finding the objective function and fitness function and the application of genetic operators. These aspects are described with the help of a basic genetic algorithm as below.

The basic principle behind GA$'$s is that they create and maintain a population of individuals represented by chromosomes. Chromosomes are essentially a character string analogous to the chromosomes appearing in DNA. These chromosomes are typically encoded solutions to a problem. The chromosomes then undergo a process of evolution according to rules of selection, reproduction and mutation. Each individual in the environ-

Algorithm 1 Basic Genetic Algorithm

Input: population
Output: Best possible solution
1. Generate random population of n chromosomes/individuals (suitable and possible solutions for the problem)
2. [Fitness] Evaluate the fitness f(x) of each chromosome/individual x in the population
3. [New population] Create a new population by repeating following steps until the New population is complete
4. [selection] select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to get selected).
5. [crossover] With a crossover probability, cross over the parents to form new offspring (children). If no crossover was performed, offspring is the exact copy of parents.
6. [Mutation] With a mutation probability, mutate new offspring at each locus (position in chromosome)
7. [Accepting] Place new offspring in the new population.
8. [Replace] Use new generated population for a further run of the algorithm.
9. [Test] If the end condition is satisfied, stop, and return the best solution in current population.
Go to the step 2 for fitness evaluation.

ment (represented by a chromosome) receives a measure of its fitness in the environment. Reproduction selects individuals with high fitness values in the population, and through crossover and mutation of such individuals, a new population is derived in which individuals may be even better fitted to their environment. The process of crossover involves two chromosomes swapping chunks of data and is analogous to the process of sexual reproduction. Mutation introduces slight changes into a small proportion of the population and is representative of an evolutionary step.

6.3 Family Gene Approach

Genetic Algorithm is one of the best optimizing algorithms currently available for users. This algorithm gives the best optimized solution for a given set of population. But the disadvantage of this algorithm is it fails to provide optimal solution when the population set is large. So to overcome this disadvantage a new enhanced genetic algorithm was proposed, namely: Family Gene Genetic Algorithm by Jianhua et al. (2006). This algorithm provides the best optimal solution even for larger set of population. The Basic algorithm as suggested by the authors is as explained below:

Algorithm 2 Original Family Gene Genetic Algorithm

Input: population,initial optimal solution

Output: Best optimal solution in the given population size

l. Initial basic population of size N is produced randomly and parameters of operators are set. The size of the current optimum families is set as I = 0 and the current number is set as J = 0 ;

2. The self-adjusting genetic operator is performed in basic family;

3. The solution X in accordance with the max fitness value of the basic population in the world space is found;

4. Initial optimum family population size m is produced within the micro search space domain and m is the size of optimum family population. The first search dimension will be searched. And the family size is increased by one, i.e.I = I + l;

5. If J < 1, then the process progresses into next step. Or else it jumps to step 12;

6. If the optimum family numbered J cannot search a larger fitness value after K times generations has been performed, searching for this dimension is finished, and start searching next dimension. If not, FGA will hold on searching this dimension.

7. If the optimum family numbered J finished searching dimensions W times, this family is destroyed and replaced by the family numbered I=1 .

8. If a gene which is found by the optimum family numbered J is better than the best gene is found by basic population in world space, this gene will be sent to the gene warehouse of the population in the world space;

9. If the optimum family numbered J finds a gene better than the parent's optimum gene, it is reproduced in the similar way narrated in step 4 but the X is the new solution with the larger fitness value.

l0. If J < I, the self-adjusting genetic operator is performing in the family numbered I;

l1. J = J +1, the process goes back to step 5;

12. The terminal condition is examined and the process is terminated when the condition is satisfied. When the terminal condition is not satisfied the process is going back to step 2.

To verify the correctness of the proposed mathematical model, we modified and implemented the Family Gene Genetic Algorithm in Cloud Environment.The adapted algorithm is as below:

Our adapted algorithm is a modified version of the existing FGA suggested by the authors Jianhua et al. (2006). In our algorithm, when the final population with best fitness function is derived, our mathematical trust model is incorporated. The trust values are randomly added to the final fitness value and then again the final population is derived again with new fitness function.

To check the correctness of the trust values calculated, the fitness values are assigned to a set of systems in a manual network set up by us. When the system is accessed for compu-

Algorithm 3 Adapted Family Gene Genetic Algorithm

Input: population, Trust value
Output: Best trust value in form of IP address retrieved, System Computation time
1: Initialize P population of n elements.
2: Use a fitness function to evaluate the current solution
3: Use genetic operators (Cross over, Mutation, Selection) to create new generations.
Go to 2 until the population does not pass the fitness criteria
4: Incorporate the Trust Model developed on the new population along with the new fitness function.
5: Find the best population from the newly incorporated population.
6: Using the best fitness access the IP of the network.

tation we found that the trust value assigned for that system matches with the trust values derived from the fitness function assigned.

6.4 Experimental Results

Our trust model was implemented using the Genetic Algorithm(GA) and Adapted Family Gene Genetic Algorithm (FGA) in Aneka Cloud Platform. Our experiment concluded that the selected IP with the best fitness value has the best trust value. The time for completing the GA process for 1000 population size (ps) was 21 seconds 397milliseconds whereas the time taken for complete execution of Family Gene Genetic Algorithm with trust incorporated was 7 seconds 805 milliseconds. Figure 6.1 and Figure 6.2 shows the results of the genetic algorithm and family gene algorithm respectively which tells that Family Gene Genetic Algorithm gives a better optimal solution.

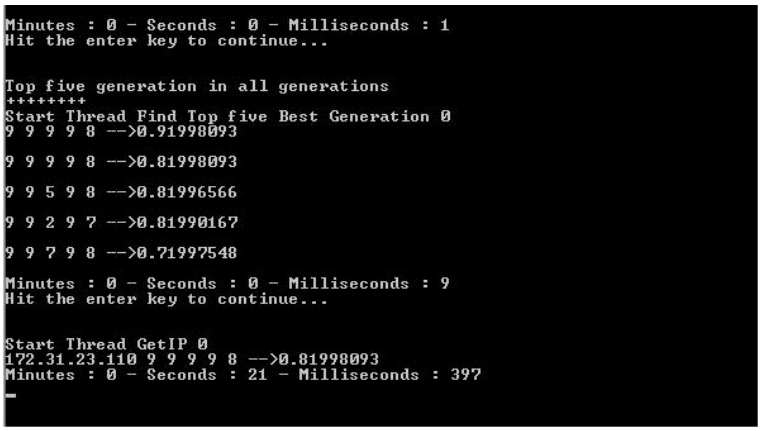

Figure 6.1 Time taken by GA

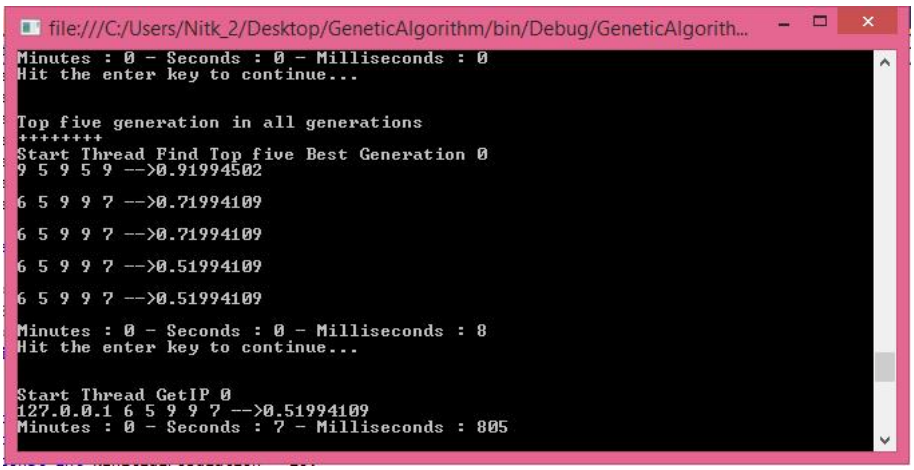

Figure 6.2 Time taken by FGA

From Figure 6.3 and Figure 6.4 it is visible that as the population size increases the computation time taken by FGA decreases. From the experiment, it is evident that Family Gene Algorithm is the best algorithm for an optimal solution when the population size is large.

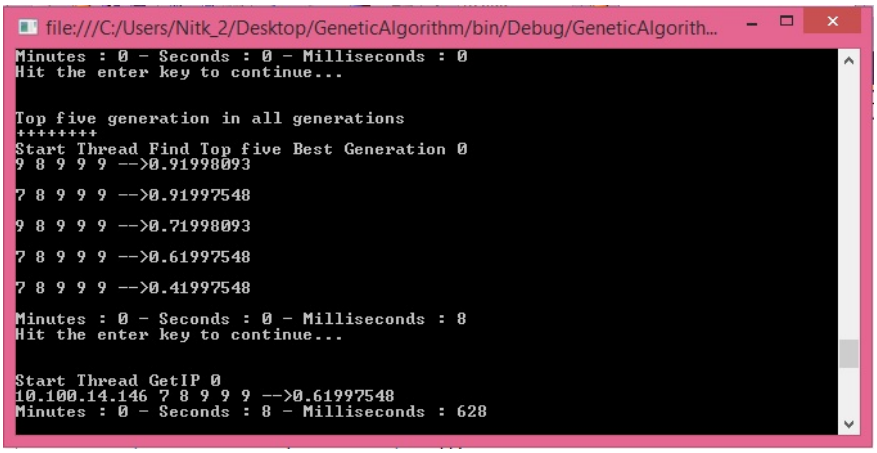

Figure 6.3 Time taken by GA for ps 100000

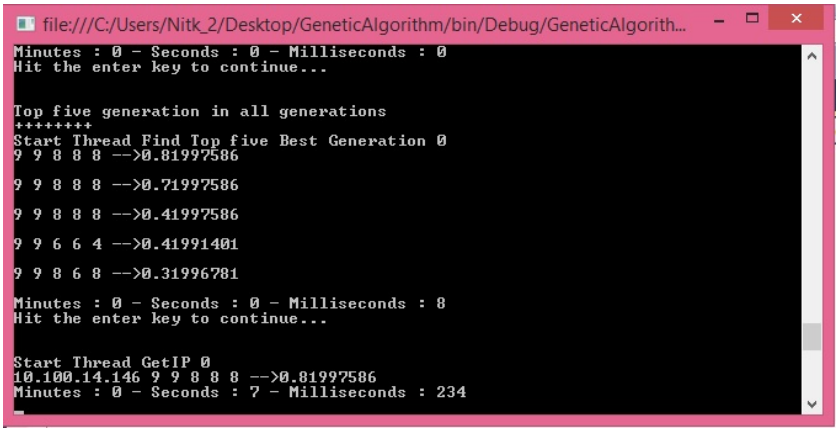

Figure 6.4 Time taken by FGA for ps 10000000

From the experiments conducted, it is concluded that the selected family gene algorithm gives an optimal solution with best fitness function which consists of the best trust values incorporated from the mathematical model.

6.5 Summary

In this chapter, we have discussed about the optimal solution algorithm called Genetic Algorithm. GA gives the best optimal solution for any NP-hard problem. The fitness function of the GA helps in generating new populations as per the requirements. But the basic disadvantage being that this algorithm fails when the population size is large. So we have considered the extension of the GA known as FGA which gives the best optimal solution even for a large population. We have modified the algorithm to check the correctness of our mathematical model. Our experiments conducted using Aneka and Opennebula cloud platforms have shown that if trust is implemented properly in a system, it would reduce the risk of security of the resources to some extent though not completely as trust is the first step of any security.

Chapter 7

END-TO-END MONITORING OF CLOUD RESOURCES USING TRUST

In our previous chapter, we showed how a well developed trust model when implemented using the optimal solution algorithm will yield better results. In this chapter, we would throw light on End-to-End resource monitoring in the cloud. Resources are major entities of any cloud and when these resources are secured with a strong trust they give better security to the system. The two major entities of the cloud are user and the resources of the cloud. A strong trust between these entities will build a secure system. So monitoring of the resources based on their availability and then allocating them to a trusted user is a major task. In the next sections, we speak more about how to handle an End-to-End resource monitoring based on their availability for allocation.

7.1 End-to-End Resource Monitoring in Cloud

End-to-End in cloud environment means the relationship between end-user and the resources. Monitoring the availability of the resources to the user is an important factor in successful transaction completion.End-to-End monitoring implies secure availability of resources to the user in the cloud environment.

End-to-End monitoring implies that the monitoring of computable resources in a cloud to meet the end-user requirement or satisfaction in conjunction with that of the resource provider. The two end points in the cloud ecosystem (user and the provider) are the important entities in supporting the proposed cloud monitoring and this can be achieved by

using the trust between the two.

Resource monitoring in a cloud computing eco system is the meaning of healthy functioning and actual consumption of cloud resources. These are the most vital to keep an eye on in order to enforce the SLA (Service Level Agreements), make sure that the resources scale appropriately, and keep the applications availability always.

Some of the important parameters of the cloud computing, pertaining to resources that need to be addressed include utilization (U), saturation (S), failure rate (F), and availability (A). Utilization refers to the percentage of the resource that is currently being used. Saturation is the amount of work waiting to be completed by the resource. The failure rate is related to the non-functioning of resources. Availability is the percentage of time that the resource has been responding to the user requests. It is customary that in a cloud computing environment, resource providers want to monitor all computing resources. Most important resources in a given cloud computing environment are storage, CPU, and memory. In this chapter we have considered storage as the resource for our study. For a storage disk, utilization would measure the amount of time the device was working, saturation would measure the length of the wait queue to write or read, errors would report any disk problems, and availability would be the percent of time the device has been available to read or write.

7.2 End-to-End Resource Monitoring Model using Trust

The model depicted below consists of the users, service providers and the resources themselves.In our end-to-end working model, cloud customers and cloud service providers who are the custodians of the resources too, are expected to be involved in the monitoring process of the model. Also, it is assumed in this model that, although the entities may be in different domains of ownership, sufficient data will be made available, by both parties involved, in order that end-to-end monitoring can be quantified.

In Figure 7.1 below, there are cloud customers and cloud service providers who are connected via Internet as known to all. The mutual trust verification - monitoring by both

parties on each other, is the essence of this model. This process is illustrated in the sequence diagram shown in Figure 7.2.

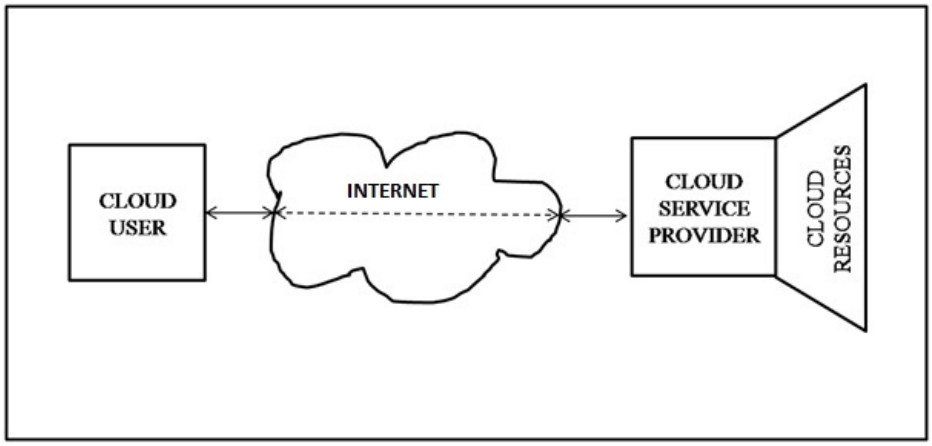

Figure 7.1 End-to-End Trust Model

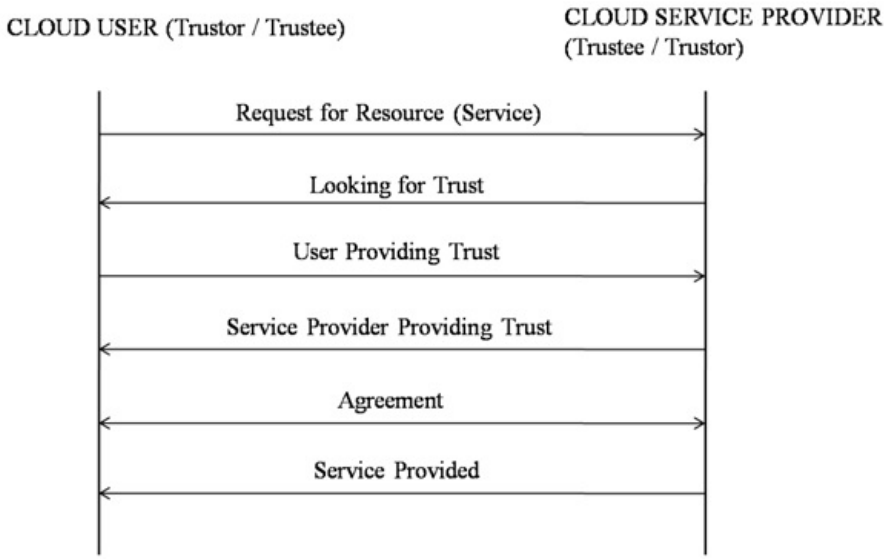

Figure 7.2 Sequence Diagram: Working of End-to-End Trust Model

Main advantages of this end-to-end model are as follows:

(a) each party, i.e., cloud consumer as well as cloud service provider monitor each others trust through a bi-lateral mechanism (Figures 7.1 and 7.2)

(b) cloud consumer maintains his trust as its profile based trust (Figure 7.3), and,

(c) cloud service provider maintains his resource related trust values (Table 7.1 and Table 7.2) that are based on the four parameters as explained before.

Trust in cloud consumers is maintained as per the profile-based trust procedure ex-

plained as per the Figure 7.3, below. Here, we assume that the users - customers have already completed the procedures related to access control or registration in the system such as log-in, in the given cloud environment. The proposed profile-based trust for users in our model is an entity that has some pre-defined privileges to access cloud services. When a user accesses the cloud and provides the these credentials or the privileges , the system in place validates the user credentials and verifies the user profile. Once the user profile is validated the trust is granted to the user for all the services that are requested for.

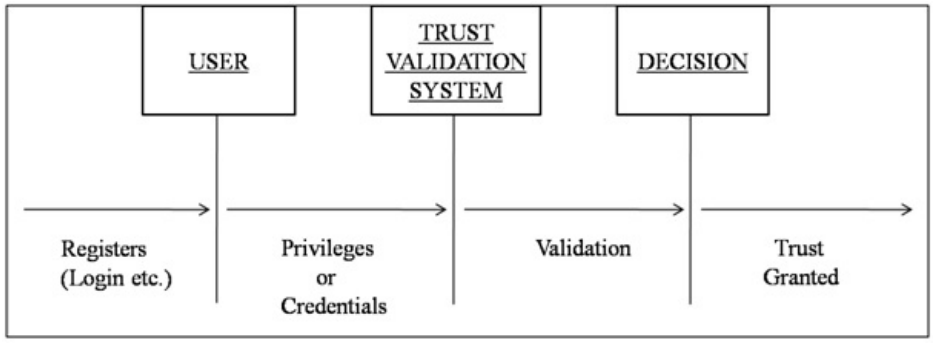

Figure 7.3 Trust for Cloud consumer - user

Trust in cloud service provider's side is maintained by keeping the consolidated value of trust based on the trust weight that are given for the four parameters considered in our work. These parameters, as described earlier, are: utilization, saturation, failure rate, and availability. Cloud consumer should be able to consider whether a trust value related to a parameter of a cloud resource is acceptable or not, based on the base value which we have considered in our work. Table 7.1 represents values that were established to determine the trust for the quantitative evaluation of a parameter.

Table 7.1 Trust Values for Quantitative Evaluation

Trust Value	**Description**	**Decision/**
Zero	No Trust	Not Accepted
0 - 0.4	Low Trust	Not Accepted
0.41 - 0.6	Average Trust	Accept but Verify
0.61 - 0.89	High Trust	Accept
0.9 - 1.0	Very High Trust	Accept

As per the details given in the Table 7.1, cloud consumer trusts cloud service provider

and thereby the resource from trust value T which is greater than or equal to 0.6. The calculation of trust of a cloud resource - service provider is being represented by the following equation.

$$Tv_{(CSP,CU)} = W1 * U + W2 * S + W3 * F + W4 * A \tag{7.1}$$

As noted from the equation, W1 is the weight factor for resource utilization, and it is 35% in our model, W2 is the weight factor for saturation, and it is 15%, W3 is the weight factor for the failure rate of the resource and it is 15%, and, W4 is the weight factor for availability of the resource and it is 35% in our assumption and trust calculation. The trust decisions based on these weight factors are tabulated here in Table 7.2.

Table 7.2 Factors Affecting Trust Decision

Utilization (W1=35)	**Saturation (W2=15)**	**Failure (W3=15)**	**Availability (W4=35)**	**Trust Decision**
High	High	High	High	High
High	Low	Low	High	High
High	High	Low	High	High
High	Low	High	High	High
Low	High	High	Low	Low
Low	Low	High	Low	Low
Low	High	Low	Low	Low
Low	Low	Low	Low	Low
High	High	Low	Low	Low
Low	Low	High	High	Low

Four parameters can have impact on calculation of trust of a cloud provider on cloud consumer about the resources, as shown in Table 7.2. Greater utilization and availability capacity have more weight in the choice of a resource consumer for more reliability because these features are responsible to ensure the integrity and usage of that resource. As the trust value ranges from [0 to 1] and is said to be dynamic, so the provider can have his resource usage increased or decreased depending on the consolidated trust value of these four parameters.

7.3 Simulation and Results

With the assigned weights, we have performed the calculation of the trust of a cloud provider for a cloud consumer. The simulation is started by performing the calculation with the cloud consumer trusting cloud provider, is assigned the value 1 to all metrics. To perform the simulation we used the standard Monte Carlo method for the generation of random values, for four metrics - parameters. Thus, from the first iteration, the values of each of these metrics are assigned randomly varying between 0 and 1, as shown in Table 7.3.

Table 7.3 Simulation and Trust Decision

Iteration	**Utilization**	**Saturation**	**Failure**	**Availability**	**Trust Decision**
1	1	1	1	1	Trust
2	0.82	0.43	0.51	0.76	Trust
3	0.06	0.11	0.30	0.89	Not Trust
4	0.68	0.01	0.50	0.32	Not Trust
5	0.70	0.34	0.61	0.74	Not Trust
6	0.25	0.67	0.32	0.54	Not Trust
7	0.76	0.98	0.64	0.69	Not Trust
8	0.77	0.24	0.87	0.87	Not Trust
9	0.98	0.42	0.33	0.89	Trust
10	0.56	0.31	0.25	0.71	Trust

Observing Table 7.3, we can see that the values of the metrics in each simulation directly influence the decision to trust or not. This procedure enables the cloud consumer to make a decision based on the trust to invoke and monitor the services of the provider and resources. In the return, the cloud provider is given the trust of the cloud consumer through his profile based approach. This end-to-end trust model has the benefits for both the parties for mutually trusting each other in the given cloud environment.

From the simulation results, we found that the trust decision varies based on the availability, failure, saturation and utilization of the resources. We further from the above Figure 7.4 found that, it is very evident that as the availability of the resources increase the trust value also increases. Using Principle-1 of entropy we know that trust increases with increase in availability of resources and it decreases when the resources requested by user are not available for the computation. Hence monitoring of resources in terms of availability increases trust of the resources if they are readily available for any computation at

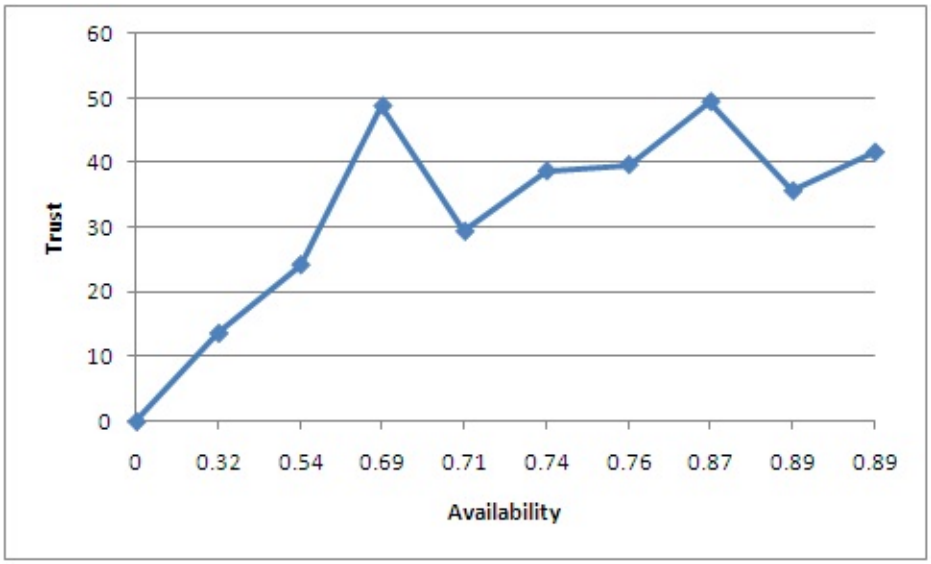

Figure 7.4 Availability v/s Trust

the request of the user. A ready availability of resource also indicates a strong trust path between the user and the resource. This strong trust path increases the trust of the user as well as the resource in the cloud environment.

7.4 Summary

In this chapter, we proposed a trust model, known as End-to-End Trust Model to monitor resources in terms of availability, that are based on four important parameters which govern the reliability of the resource. This model is to ensure that the trust values being exchanged between consumers and providers are of great importance, due to the fact that mutual trust values are being used for the resource utilization. In this model, the trust value of a given resource is obtained from a set of four important parameters for monitoring resource in terms of availability in any related operations. Cloud providers and thereby resources with greater trust values are subsequently chosen by the cloud consumers. So the mathematical model proposed as Principle-1 in Chapter 5 is proven that there is increase in trust due to the availability of resources.

Chapter 8

PERFORMANCE ANALYSIS WITH PERCEIVED FACTORS

In our previous chapters, we have seen how trust can be calculated in different ways and how the implementation of the trust using the optimal solution algorithm gives a better results than others. In this chapter, we would analyze the performance of the developed model implemented in a cloud system using the perceived factors like performance.

8.1 Perceived Factors

Perception is the process by which an individual selects, organizes, and interprets information inputs to create a meaningful picture the world. In technology, perception is the process of getting the work done with accuracy. In cloud computing environment perception means the ease of use of the services with utmost security. Multiple factors attribute to the perception Lin and Chen (2012) like:

Relative Advantage: is defined as the degree to which a cloud service is perceived as an improvement on the system it is intended to replace.

Ease of Use: is defined as the degree to which an individual believes that using a cloud service would be free of effort.

Compatibility: as the degree to which a cloud service is perceived as being consistent with the existing values, needs, and past experiences of the potential adopters.

Trialability: is defined as the degree to which a cloud service may be experimented with before adoption.

Security: is defined as the extent to which a user believes that using cloud services will be

risk-free. Security has become the most important factor influencing an enterprise adoption of internet-related services since risk perception has been increased in the context of threats associated with applications.

Performance: is defined as the speed at which the system works.

To understand the user perception of trust, we need to understand what trust means and the way it is established and whether trust perception changes based on different application domains. Trust has been defined in different ways as explained by Costante et al. (2011): as expectation, as vulnerable to the actions of others, as probability, and as a risk. Studies on trust perception aim to understand the mechanisms adopted by humans to trust other humans, machines or e-services.

8.1.1 Perceived Factors Influencing Cloud Environment

Cloud is a emerging technology in current scenario. It brings with it major benefits like low cost, easy to adapt,interoperability, etc. However, it faces a risk in terms of unintended economic and security impacts. Perception of using the cloud technology due to its ease of adoption is extensively increasing in the current market. Some of the perceived factors affecting the adoption of cloud are:

Risk: Though the cloud technology is a ease-at-use concept the risk of security of the data with the service providers is unknown. This causes a major concern for all the organizations to migrate completely to the cloud.

Privacy: The authentication and security of the data when it reaches the hands of the service provider is at his discreet. The security measures taken by the service provider are not known to the user, and so the user data might be at risk.

Transparency: In the case of IaaS service model of cloud, the security measures taken by the service providers for the security of data is very little known or almost unknown.

Relative Affordability: As the concept of cloud provides pay-as-Use the resources are readily available to the user at his door step. But the privacy of his data is at risk.

Performance: As the users computations are done using the resources of the service provider using internet medium the performance of the system might vary due to many reasons like high internet speed, in-house cloud platform, etc.

Technology Readiness: The resources are easily available through the internet at the door

step of the user, he is willingly using the technology for his computations.

Environmental Uncertainty: The data centers where the user data is placed is not known by the user. Retrieving the data from the stored location can be a problem due to multiple factors like land governance rules, natural calamity, etc.

In spite of these factors that pose risks and security concerns, major organizations want to migrate to cloud environment rather than manage resources at their premises.This leads to low cost of maintenance, security and adaptability.

8.2 System Performance as Perceived Factor

Performance measurement is a topic which is often discussed but rarely defined. Literally, it is the process of quantifying action, where measurement is the process of quantification and action leads to performance.

The level of performance a business attains is a function of the efficiency and effectiveness of the actions it undertakes, and thus:

Performance measurement as explaned by Lin and Chen (2012)can be defined as the process of quantifying the efficiency and effectiveness of action.

A performance measure can be defined as a metric used to quantify the efficiency and/or effectiveness of an action.

In Cloud Computing performance is measured as the ease of use of the services of the service provider with utmost security.

In our experiments we have considered Compute time and CPU time as performance metrics.

CPU Time: is the amount of time for which a central processing unit (CPU) was used for processing instructions of a computer program or operating system.

Compute Time or Computational Time: Computational time is the length of time required to perform a computational process.

8.3 Experiment and Results

Aneka is a PaaS based cloud environment.It offers platform as a service to the customers. Aneka has a client server architecture. We have developed the code of our adapted FGA

using this Aneka platform. We have considered 3 nodes of Aneka platform to conduct our experiment.

Opene Nebula is an open source cloud platform which provides Infrastructure as service to its customer. We installed this open source cloud in Linux environment. To create Virtual Machines in Opennebula we used a virtualisation software called VMware. Using VMware we created 3 virtual machines in opennebula environment to conduct our experiment.

Using our adapted FGA algorithm, we found that whenever the population size of the algorithm increases the performance of the system only increases.

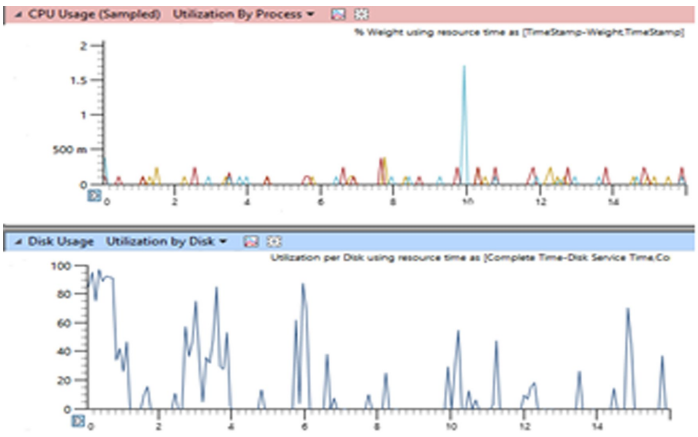

Figure 8.1 Performance Analysis Before Implementation

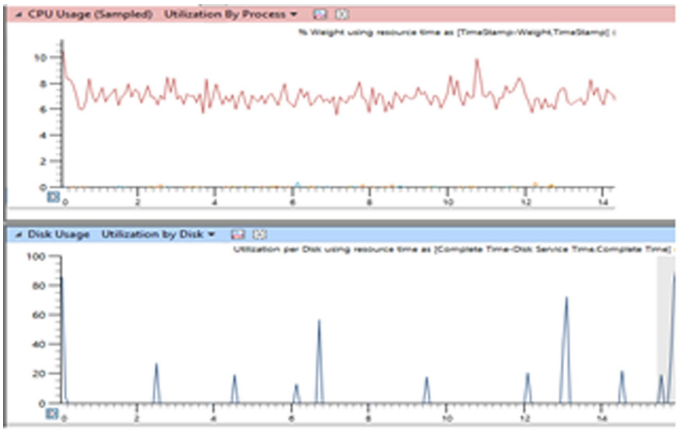

Figure 8.2 Performance Analysis After Implementation

In Figure 8.1 Performance Analysis in terms of CPU (Central Processing Unit) usage and Storage usage. In the CPU usage graph, the X-axis consists of the time-stamp with weight and Y-axis consists of the time-stamp and the processes running in the system,

Aneka Manager, Aneka UI (User Interface). In the storage graph, the X-axis Consists of Disk Service Time and Y-Axis Consists of Complete time. It is clearly evident from the figure that when the system is in idle state the basic usage of CPU and Storage is minimal. In Figure 8.2 Performance Analysis of the system after Implementing Proposed Trust model represents the performance analysis of the proposed model with regard to CPU usage and Storage. From the performance analysis, it is clearly evident that the system CPU and Storage are well utilized optimally.

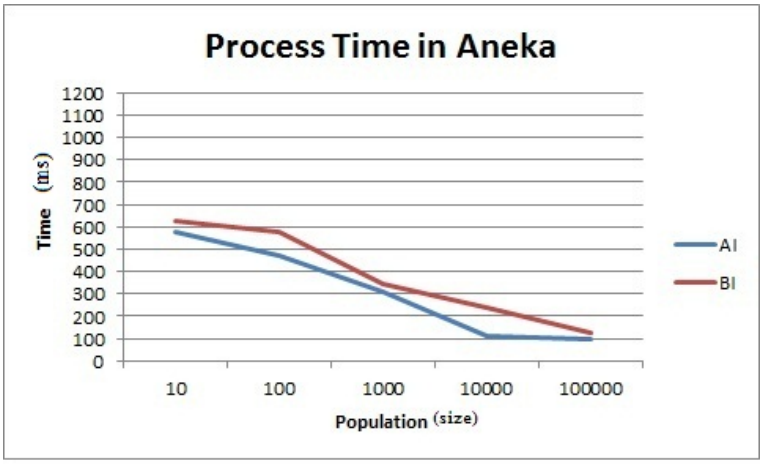

Figure 8.3 Process time in Aneka

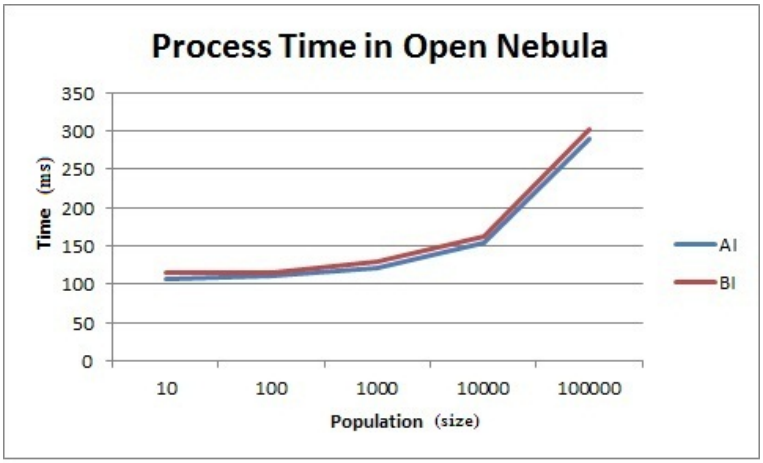

Figure 8.4 Process time in Opennebula

We implemented our adapted FGA in both the cloud environments, without incorporating the trust values and also by incorporating the calculated trust values got from the trust model proposed. Figure 8.3 and Figure 8.4 represent the process time in Aneka and Opennebula platforms as Before Implementation of Trust (BI) and After Implementation

of Trust (AI).We found that process time of the system in Aneka and Opennebula before implementation of trust model is slightly higher than the process time of the system after the implementation of the trust model.

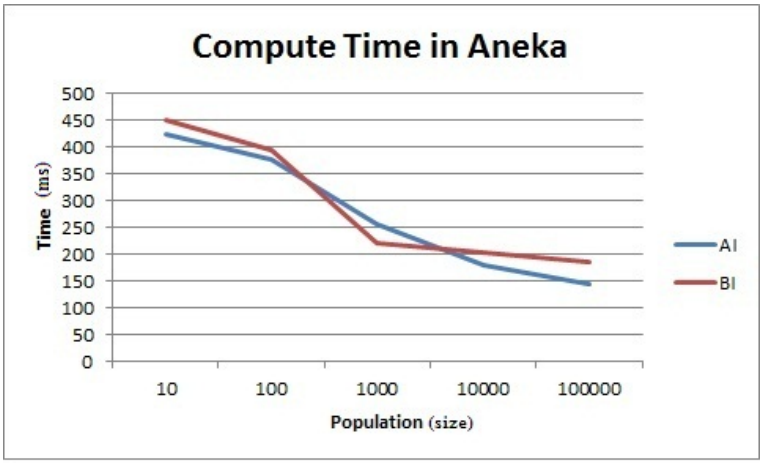

Figure 8.5 Compute time in Aneka

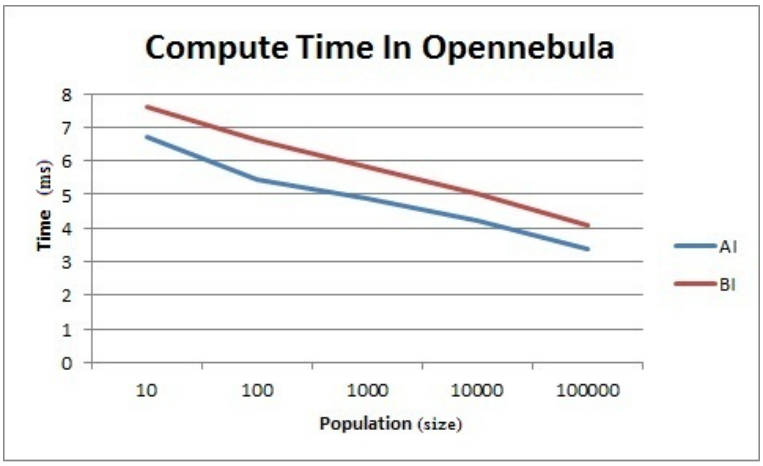

Figure 8.6 Compute time in Opennebula

Figure 8.5 and Figure 8.6 represent the compute time in Aneka and Opennebula as Before Implementation of Trust(BI) and After Implementation of Trust(AI). We found that compute time of the system in Aneka is although the same before and after implementation but yields better results after implementation of the trust model when population size increases. The compute time of the system in Opennebula before implementation of trust model is slightly higher than the compute time of the system after the implementation of the trust model thus clarifying that the proposed model yields better results with system performance.

8.4 Summary

We have implemented our model in Aneka and Opennebula cloud platform. We have created a network of 3 machines on which Aneka and Opennebula are installed. Our model is implemented on these 3 machines simultaneously with other homogeneous tasks running in the background. We considered performance factors like compute time and process time to check the performance of the system with our model running in back ground. We found that the performance of the system in-fact does not deteriorate rather only improves due the trust model implemented in the system. We also found from our experiments that our proposed model was platform independent as we implemented the model in the two different cloud platforms namely: Aneka and Opennebula.

Chapter 9

CONCLUSION & FUTURE WORK

9.1 CONCLUSION

In recent years, cloud computing has become a vibrant and rapidly expanding area of research and development. In today's competitive environment, the service dynamism, elasticity, and the choices offered by a highly scalable cloud computing technology are too attractive for enterprises to ignore. These opportunities, however, don′t come without challenges. Because CSP controls the data, enterprises are concerned about different challenges on data confidentiality, privacy, integrity, and availability for CCs. Today, the problem of trusting cloud computing is a paramount concern for most enterprises in such a way that trust is widely regarded as one of the top obstacles to the adoption and growth of cloud computing. In order to evaluate trust management systems, trust models have been developed. However, each developed trust model evaluates a limited number of assessment criteria and it is hard for enterprises to use these trust models in their decision-making process.

To enhance the security of any network there is a need for strong trust between the entities in that network as trust is the first step towards security. A strong trust model is needed to signify the importance of trust. Our Conceptual framework shows the way to create a strong trust path between the two entities of cloud viz: user and resources.

From the literature survey it is well known that trust of resources based on their availability in never considered towards security of the resources. So the mathematical model proposed enables to evaluate the trust of the user as well as the resource in the cloud. Based on the trust value the service provider can decide to allocate the requested resources to the

user. As cloud is dynamic environment static trust value would not add for the trust of the resource or the user. So dynamic trust calculation is proposed to evaluate new trust value due to the scalability in cloud. In a federated cloud environment traditional SLA based trust proposes many problems as the need for common conscience is very much necessary. So to make migration a bit more easy trust model for migration is proposed which helps the user as well as the service provider in evaluating the trust of the user/resource during migration.

As cloud is a scalable environment the number of VM′s at a particular time is unpredictable. The best optimal solution algorithm is used to evaluate the trust of the VM′. The optimal solution algorithm FGA evaluates the trust values calculated and validates the mathematical model. The implementation of the model using FGA in various cloud platforms emphasizes the matter that the proposed trust model is platform independent.The end-end trust model proposed enhances the security of the entities based on the four parameters, as there would be a strong trust path built between the entities due to the monitoring of the resources based on their availability. The perceived factors like performance in terms of CPU time and Compute time, ease of use have proved that the proposed trust model when implemented using the FGA gives an utmost performance and is platform independent.

The proposed trust model was implemented in two different cloud environments to check for the platform dependency and we found that our model is platform independent as we implemented in two different service models of cloud namely: Aneka and Opennebula.

A strong security of the system in terms of trust will always enhance the overall security of the system as a strong trust path built between the entities does not allow the breach of conduct of the entities. Thus the proposed model enhances the security of the resources and user in cloud computing environment.

Future Work

Our future work includes implementing the proposed trust model for migration in a federated cloud environment, which would help in migrating from one cloud platform to another with minimum effort. Our end-end trust model was simulated using the trust

values based on only four parameters namely: Saturation, Utilization, Failure Rate And Availability. We also would like to extend the proposed model for further more parameters of the cloud like scalability, security, integration and cost. We would also like to experiment the proposed model in real time cloud environment to check its effectiveness and platform dependency.

Bibliography

Aneka architecture. http://www.manjrasoft.com/aneka-architecture.html. Accessed: 2013-09-30.

Open nebula technology. http://opennebula.org/about/technology. Accessed: 2013-09-30.

Alhamad, M., Dillon, T., and Chang, E. (2010). Sla-based trust model for cloud computing. In *Network-Based Information Systems (NBiS), 2010 13th International Conference on*, 321–324. Ieee.

Anisetti, M., Ardagna, C. A., and Damiani, E. (2014). A certification-based trust model for autonomic cloud computing systems. In *Cloud and Autonomic Computing (ICCAC), 2014 International Conference on*, 212–219. IEEE.

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., et al. (2010). A view of cloud computing. *Communications of the ACM*, 53(4), 50–58.

Bennani, N., Boukadi, K., and Ghedira-Guegan, C. (2014). A trust management solution in the context of hybrid clouds. In *WETICE Conference (WETICE), 2014 IEEE 23rd International*, 339–344. IEEE.

Blaze, M., Feigenbaum, J., and Lacy, J. (1996). Decentralized trust management. In *Security and Privacy, 1996. Proceedings., 1996 IEEE Symposium on*, 164–173. IEEE.

Chakraborty, S. and Roy, K. (2012). An sla-based framework for estimating trustworthiness of a cloud. In *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, 937–942. IEEE.

Cheng, Y., Li, X.-Y., and Ling, M.-Q. (2012). A trusted cloud service platform architecture. In *Information Science and Applications (ICISA), 2012 International Conference on*, 1–6. IEEE.

Costante, E., Den Hartog, J., and Petkovic, M. (2011). On-line trust perception: What really matters. In *Socio-Technical Aspects in Security and Trust (STAST), 2011 1st Workshop on*, 52–59. IEEE.

Cover, T. M. and Thomas, J. A. (2012). *Elements of information theory*. John Wiley & Sons.

Cover, T. M., Thomas, J. A., et al. (1991). Elements of information theory.

Cummings, L. L. and Bromiley, P. (1996). The organizational trust inventory (oti). *Trust in organizations: Frontiers of theory and research*, 302(330), 39–52.

Divakarla, U. and Chandrasekaran, K. (2016). Secure allocation of resources in cloud using trust. *International Journal of Computer Network and Information Security*, 8(1), 43.

Fan, W. and Perros, H. (2013). A reliability-based trust management mechanism for cloud services. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2013 12th IEEE International Conference on*, 1581–1586. IEEE.

Filali, F. Z. and Yagoubi, B. (2015). Global trust: A trust model for cloud service selection. *International Journal of Computer Network and Information Security*, 7(5), 41.

Firdhous, M., Ghazali, O., and Hassan, S. (2011a). A trust computing mechanism for cloud computing. In *Kaleidoscope 2011: The Fully Networked Human?-Innovations for Future Networks and Services (K-2011), Proceedings of ITU*, 1–7. IEEE.

Firdhous, M., Ghazali, O., and Hassan, S. (2011b). A trust computing mechanism for cloud computing with multilevel thresholding. In *Industrial and Information Systems (ICIIS), 2011 6th IEEE International Conference on*, 457–461. IEEE.

Firdhous, M., Ghazali, O., and Hassan, S. (2012a). Trust management in cloud computing: a critical review. *arXiv preprint arXiv:1211.3979*.

Firdhous, M., Hassan, S., and Ghazali, O. (2012b). Hysteresis-based robust trust computing mechanism for cloud computing. In *TENCON 2012-2012 IEEE Region 10 Conference*, 1–6. IEEE.

Fu, J., Wang, C., Yu, Z., Wang, J., and Sun, J.-G. (2010). A watermark-aware trusted running environment for software clouds. In *ChinaGrid Conference (ChinaGrid), 2010 Fifth Annual*, 144–151. IEEE.

Gabarro, J. J. (1978). The development of trust, influence, and expectations. *Interpersonal behavior: Communication and understanding in relationships*, 290, 303.

Gambetta, D. et al. (1988). Can we trust trust. *Trust: Making and breaking cooperative relations*, 13, 213–237.

Grandison, T. and Sloman, M. (2000). A survey of trust in internet applications. *IEEE Communications Surveys & Tutorials*, 3(4), 2–16.

Habib, S. M., Ries, S., and Muhlhauser, M. (2011). Towards a trust management system for cloud computing. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on*, 933–939. IEEE.

Haq, I. U., Brandic, I., and Schikuta, E. (2010). Sla validation in layered cloud infrastructures. In *International Workshop on Grid Economics and Business Models*, 153–164. Springer.

Hermerschmidt, L., Perez, A. N., and Rumpe, B. (2014). A model-based software development kit for the sensorcloud platform. In *Trusted Cloud Computing*, 125–140. Springer.

Holmes, J. G. (1991). Trust and the appraisal process in close relationships.

Hu, J., Wu, Q., and Zhou, B. (2008). Ttem: An effective trust-based topology evolution mechanism for p2p networks. *Journal of Communications*, 3(7), 3–10.

Huang, J. and Nicol, D. M. (2013). Trust mechanisms for cloud computing. *Journal of Cloud Computing: Advances, Systems and Applications*, 2(1), 9.

Jianhua, L., Xiangqian, D., Sun'an, W., and Qing, Y. (2006). Family genetic algorithms based on gene exchange and its application. *Journal of Systems Engineering and Electronics*, 17(4), 864–869.

Kanwal, A., Masood, R., Ghazia, U. E., Shibli, M. A., and Abbasi, A. G. (2013). Assessment criteria for trust models in cloud computing. In *Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing*, 254–261. IEEE.

Khan, K. M. and Malluhi, Q. (2010). Establishing trust in cloud computing. *IT professional*, 12(5), 20–27.

Ko, R. K., Jagadpramana, P., Mowbray, M., Pearson, S., Kirchberg, M., Liang, Q., and Lee, B. S. (2011). Trustcloud: A framework for accountability and trust in cloud computing. In *Services (SERVICES), 2011 IEEE World Congress on*, 584–588. IEEE.

Li, W., Ping, L., and Pan, X. (2010a). Use trust management module to achieve effective security mechanisms in cloud environment. In *Electronics and Information Engineering (ICEIE), 2010 International Conference On*, 1, 1–14. IEEE.

Li, X.-Y., Zhou, L.-T., Shi, Y., and Guo, Y. (2010b). A trusted computing environment model in cloud architecture. In *Machine Learning and Cybernetics (ICMLC), 2010 International Conference on*, 6, 2843–2848. IEEE.

Lin, A. and Chen, N.-C. (2012). Cloud computing as an innovation: Percepetion, attitude, and adoption. *International Journal of Information Management*, 32(6), 533–540.

Mansour, M., Fahmy, A., and Hashem, M. (2012). Maintaining location privacy and anonymity for vehicleâĂŹs drivers in vanet. *International Journal of Emerging Technology and Advanced Engineering*, 2(11), 8.

Manuel, P. (2015). A trust model of cloud computing based on quality of service. *Annals of Operations Research*, 233(1), 281–292.

Manuel, P. D., Selvi, S. T., and Abd-El Barr, M. I. (2009). Trust management system for grid and cloud resources. In *Advanced Computing, 2009. ICAC 2009. First International Conference on*, 176–181. IEEE.

Mell, P., Grance, T., et al. (2011). The nist definition of cloud computing.

Mohammadnia, H. and Shakeri, H. (2014). Hitcloud: Novel hierarchical model for trust management in cloud computing. In *Technology, Communication and Knowledge (ICTCK), 2014 International Congress on*, 1–8. IEEE.

Muchahari, M. K. and Sinha, S. K. (2012). A new trust management architecture for cloud computing environment. In *Cloud and Services Computing (ISCOS), 2012 International Symposium on*, 136–140. IEEE.

Noor, T. H., Sheng, Q. Z., Zeadally, S., and Yu, J. (2013). Trust management of services in cloud environments: Obstacles and solutions. *ACM Computing Surveys (CSUR)*, 46(1), 12.

Noraini, M. R. and Geraghty, J. (2011). Genetic algorithm performance with different selection strategies in solving tsp.

Pavlidis, M., Mouratidis, H., Kalloniatis, C., Islam, S., and Gritzalis, S. (2013). Trustworthy selection of cloud providers based on security and privacy requirements: Justifying trust assumptions. In *International Conference on Trust, Privacy and Security in Digital Business*, 185–198. Springer.

Pearson, S. (2009). Taking account of privacy when designing cloud computing services. In *Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing*, 44–52. IEEE Computer Society.

Pearson, S. and Benameur, A. (2010). Privacy, security and trust issues arising from cloud computing. In *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*, 693–702. IEEE.

Ranchal, R., Bhargava, B., Othmane, L. B., Lilien, L., Kim, A., Kang, M., and Linderman, M. (2010). Protection of identity information in cloud computing without trusted third party. In *Reliable Distributed Systems, 2010 29th IEEE Symposium on*, 368–372. IEEE.

Sato, H., Kanai, A., and Tanimoto, S. (2010). A cloud trust model in a security aware cloud. In *Applications and the Internet (SAINT), 2010 10th IEEE/IPSJ International Symposium on*, 121–124. IEEE.

Shen, Z., Li, L., Yan, F., and Wu, X. (2010). Cloud computing system based on trusted computing platform. In *Intelligent Computation Technology and Automation (ICICTA), 2010 International Conference on*, 1, 942–945. IEEE.

Shen, Z. and Tong, Q. (2010). The security of cloud computing system enabled by trusted computing technology. In *Signal Processing Systems (ICSPS), 2010 2nd International Conference on*, 2, V2–11. IEEE.

Song, S., Hwang, K., Zhou, R., and Kwok, Y.-K. (2005). Trusted p2p transactions with fuzzy reputation aggregation. *IEEE Internet computing*, 9(6), 24–34.

Subashini, S. and Kavitha, V. (2011). A survey on security issues in service delivery models of cloud computing. *Journal of network and computer applications*, 34(1), 1–11.

Sun, X., Chang, G., and Li, F. (2011). A trust management model to enhance security of cloud computing environments. In *Networking and Distributed Computing (ICNDC), 2011 Second International Conference on*, 244–248. IEEE.

Supriya, M. et al. (2012). Estimating trust value for cloud service providers using fuzzy logic.

Takabi, H., Joshi, J. B., and Ahn, G.-J. (2010). Securecloud: Towards a comprehensive security framework for cloud computing environments. In *Computer Software and Applications Conference Workshops (COMPSACW), 2010 IEEE 34th Annual*, 393–398. IEEE.

Uikey, C. and Bhilare, D. (2013). A broker based trust model for cloud computing environment. *International Journal of Emerging Technology and Advanced Engineering*, 3(11), 247–252.

Vivekananth, P. (2010). A behavior based trust model for grid security. *International Journal of Computer Applications*, 5, 1–3.

Wang, H., Zhao, G., Chen, Q., and Tang, Y. (2013). Trust management for iaas with group signature. In *Emerging Intelligent Data and Web Technologies (EIDWT), 2013 Fourth International Conference on*, 422–430. IEEE.

Wang, S., Zhang, L., Ma, N., and Wang, S. (2008). An evaluation approach of subjective trust based on cloud model. In *Computer Science and Software Engineering, 2008 International Conference on*, 3, 1062–1068. IEEE.

Wang, S.-X., Zhang, L., Wang, S., and Qiu, X. (2010a). A cloud-based trust model for evaluating quality of web services. *Journal of Computer Science and Technology*, 25(6), 1130–1142.

Wang, T., Ye, B., Li, Y., and Yang, Y. (2010b). Family gene based cloud trust model. In *Educational and Network Technology (ICENT), 2010 International Conference on*, 540–544. IEEE.

Wang, T., Ye, B., Li, Y., and Zhu, L. (2010c). Study on enhancing performance of cloud trust model with family gene technology. In *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on*, 9, 122–126. IEEE.

Xia, H., Jia, Z., Ju, L., Li, X., and Zhu, Y. (2011). A subjective trust management model with multiple decision factors for manet based on ahp and fuzzy logic rules. In *Green Computing and Communications (GreenCom), 2011 IEEE/ACM International Conference on*, 124–130. IEEE.

Xiaolin, G., Bing, X., Yinan, L., and Depei, Q. (2004). Study on the behavior-based trust model in grid security system. In *Services Computing, 2004.(SCC 2004). Proceedings. 2004 IEEE International Conference on*, 506–509. IEEE.

Xiong, L. and Liu, L. (2003). A reputation-based trust model for peer-to-peer e-commerce communities. In *E-Commerce, 2003. CEC 2003. IEEE International Conference on*, 275–284. IEEE.

Xu, F. and Guo, Y. (2009). A domain-based trust model in peer-to-peer environment.

Yang, Z., Qiao, L., Liu, C., Yang, C., and Wan, G. (2010). A collaborative trust model of firewall-through based on cloud computing. In *Computer Supported Cooperative Work in Design (CSCWD), 2010 14th International Conference on*, 329–334. IEEE.

Yu, F., Zhang, H., Yan, F., and Gao, S. (2006). An improved global trust value computing method in p2p system. *Autonomic and trusted computing*, 258–267.

Zhao, G., Wang, H., Rong, C., and Tang, Y. (2013). Resource pool oriented trust management for cloud infrastructure. In *Availability, Reliability and Security (ARES), 2013 Eighth International Conference on*, 268–273. IEEE.

LIST OF PUBLICATIONS/ CONFERENCE PAPERS

Journal Publications

[1] Usha Divakarla., K. Chandrasekaran (2015). *"Secure Allocation of Resources in Cloud Using Trust."* I. J. Computer Network and Information Security 2016, 1, 43-52.

[2] Usha Divakarla., K. Chandrasekaran (2016). *"Enhanced Trust Path Between Two Entities in Cloud Computing Environment."* International Journal of Cloud Applications and Computing.Volume 6, Issue 3,July-September 2016.

Conference Proceedings

[1] Usha Divakarla., K. Chandrasekaran (2014). *"Trust Models in Cloud: A Survey on Pros and Cons."* New Trends in Networking, Computing, E-learning, Systems Sciences, and Engineering Lecture Notes in Electrical Engineering Volume 312, 2015, pp 335-341.(*Springer*)

[2] Usha Divakarla.,K. Chandrasekaran (2013). *"Trust: A Psychological and Technical Aspect."* In the Proceedings of International conference on IMPact Of E-Technology On US.IC-IMPETUS 2013.

[3] Usha Divakarla.,K. Chandrasekaran (2016). *" Trusted path between two entities in Cloud."* In Cloud System and Big Data Engineering (Confluence), 2016 6th International Conference (pp. 157-162). IEEE.

[4] Usha Divakarla.,K. Chandrasekaran (2016). *"A novel approach for evaluating trust of resources in cloud environment."* In Region 10 Conference (TENCON), 2016 IEEE (pp. 459-463).

[5] Usha Divakarla.,K. Chandrasekaran (2017). *"End-to-End Monitoring of Cloud Resources Using Trust."* In the proceedings of 2nd International Conference on Computer Communication and Computational Science - IC4S 2017(Springer).

[6] Usha Divakarla.,K. Chandrasekaran (2017). *"Trust Aware Secure Service-Composition in Cloud Environment."* In the proceedings of 6th International Engineering Symposium - IES 2017.

BIO-DATA

Name : Usha D

Email Id : ushachavali@gmail.com

Date of Birth : June 3, 1976

Address : D/o. D Satyanarayana,

H.No 359,

KHB Colony,Kallahalli

Vinobanagar,Shimoga

Educational Qualifications:

Degree	Year of Passing	University
B.E.	2000	Bangalore University, Bangalore.
M.Tech	2006	Maharshi Dayanand University, Haryana.