In the Name of Allah
The Most Merciful,
The Beneficial

ACKNOWLEDGMENTS

Innumerable words of praise and thanks to Allah, the Almighty, and the Creator of the universe for carving the path for me and always helping me out in the best possible way. Without His Will and mercy, I would not be able to accomplish this milestone. I am speechless to my kith & kin, expressively my mom & dad for their immense love, moral support, encouragement and prayers throughout and lay the foundation of every achievement during my career and life. I would like to mention all my supportive brothers explicitly Omer, Ali, Hamza & Hassan and my adorable sister Humaira for boosting my moral and always being there for me.

I am deeply obliged to my kind supervisor, Brig. Dr. Muhammad Younus Javed, for always supporting my ideas and listening to me with endurance. His valuable guidance in the light of his diverse knowledge and experience and precious words of inspiration formed the backbone of my educational life which sharpened my skills and would prove to be fruitful in the future too INSHALLAH. I would like to thank all commendable faculty members, who really have always been a source of inspiration.

I gratefully acknowledge the courage, strength and guidance provided by Lt cmdr Nazir Malik which made me stand on my feet for pursuing this project. Their valuable connotations and comments were a great boost for the progress of this dissertation.

I am highly thankful to the administration of the department especially lab administrators for their every possible help and assistance throughout my entire research work.

In the end, I would like to express my extreme gratitude to all my loyal friends for their cooperation, motivation and advice. Last but not the least I am grateful to every single entity of this universe that proved successful for the project development.

DEDICATION

To Allah, Al Rehman Al Rahim

To Our Prophet Muhammad SAW who brought us from darkness to sunlight
&
To my Parents, the foremost source of my being

ABSTRACT

With the increasing amount of the handheld devices and respective pervasive services, the issues of security have alarmingly risen. One can wonder about the pervasive services to be free of every security flaw. The paper suggests a mechanism for a trust based ad-hoc authentication with a distributed mechanism and authorization to pervasive services. The openness of a truly pervasive environment has been encountered with its interaction history and recommendation from trustworthy sources. Proposed distributed security architecture is realized with a web based framework managing the user registration of the university at a novel level. The architecture is designed to provide secure autonomous services for an assumed pervasive based scenario. The nature of pervasiveness is fulfilled by allowing the novel user to proceed in the system but with the minimum trust initially which changes with the passage of time, dynamically depending on the actions which the user performs. Certain actions on the specified services are granted to the user on the basis of trust level and user category. Delegation access control module exists with respect to the time period.

Table of Contents

Chapter 1 1

INTRODUCTION 1

1.1 Introduction 1

1.2 Challenges Of Pervasive Computing 4

1.3 Security issues of Pervasive Computing 6

1.4 Trust as a security paradigm: 7

1.5 Problem statement 9

1.6 Objective 9

1.7 Thesis Organization 10

1.8 Conclusion 10

Chapter 2 11

LITERATURE REVIEW 11

2.1 Introduction 11

2.2 Keynote, Policy Maker, Referee 11

2.2.1 PolicyMaker [1996] developed by Matt Blaze 11

2.2.2 The KeyNote 12

2.2.3 Advantages & Disadvantages: 12

2.2.4 REFEREE 12

2.3 Trust Based Delegation System 13

2.4 Distributed Trust management [DTM] 14

2.5 TrustAC: Trust-Based Access Control for Pervasive Devices 15

2.6 Trust Distributed Trust Management System Architecture 16

2.7 Middleware Architecture for Mobile Respective Pervasive Environments 17

2.8 An Adaptive Lightweight Trust Reliant Secure Resource Discovery 18

2.9 Evaluating Trust-based Access Control for Social Interaction 19

2.10 Conclusion: 20

Chapter 3 21

PROPOSED MODEL DESIGN 21

3.1 Introduction 21

3.2 Research Methodology 21

3.2.1 Stages of Constructive Research Methodology 21

3.3 Design Specifications of the Proposed Architecture 23

3.4 Description of Proposed Architecture 24

3.4.1 Trust Computation Module (TCM) 24

3.4.2 Trust Category Computation Module (TCAM) 26

3.4.3 Access Control Manager (ACM) 26

3.4.4 Delegation Control Module 26
3.5 Class Diagram 27
3.5.1 Access control of the University system 28
3.5.2 Delegation module 29
3.6 Trust properties and access flow control 29
3.6.1 Introduction 29
3.6.2 Trust Computation Module (TCM) Flow Control 30
3.6.3 Trust Category Assignment Module (TCAM) Flow Control 30
3.6.4 Update Trust value Module 31
3.6.5 Access Control Manager (ACM) Flow Control 32
3.6.6 Delegation Control Manager (DCM) Flow Control 33
3.7 PROPOSED ALGORITHM 34
Chapter 4 38
FRAMEWORK DEVELOPMENT OF DISTIBUTED TRUST BASED ACCESS CONTROL IN ASSUMED SCENARIO OF PERVASIVE COMPUTING 38
4.1 Introduction 38
4.2 System Specifications 39
4.2.1 Ad hoc User 39
4.2.2 A Trusted User i.e. Administrator 43
4.3 Trust Value Calculation 45
4.3.1 Introduction 45
4.3.2 Recommendations....................47
4.3.3 Observations 49
4.3.4 Trust Value Calculation....................51
Chapter 5 57
SYSTEM ANALYSIS AND RESULTS 57
5.1 Introduction 57
5.2 Weight (W1) of Observation against Trustworthiness 57
5.3 Weight (W2) Observation against Trustworthiness 58
5.4 Comparison of Weight Changing behaviour Against Trustworthiness....................59
5.5 Analysis of Trusted User Actions against Trust Values 61
5.6 Analysis of Ad-Hoc User Actions against Trust Values....................61
5.7 Average of the Observations against Trust Value 62
5.8 Average of the Recommendation against Trust Values....................63
5.9 Direct & Indirect Interactions against Trust Value 64
5.10 Direct Trust i.e. Observations Against Indirect Trust i.e. Recommendation 65
5.11 Evaluation 68
5.12 Conclusion:....................70
Chapter 6 71

CONCLUSION & FUTURE WORK........71

6.1 CONCLUSION........71

6.2 FUTURE WORK........72

REFERENCES........73

Table of Figures

Figure 1-1 Smart devices of pervasive environment 1

Figure 1-2 Pervasive computing environment. Mobile devices connected via wired and wireless links to one another and to the global networking infrastructure providing information and integrated services. 3

Figure 1-3 Ubiquitous Service Model 4

Figure 1-4 Ubiquitous environment involving authentication mechanisms 5

Figure 1-5 Biometric Authentication 6

Figure 1-6 Digital certificate 6

Figure 1-7 unique challenges and requirements for security insurance in a pervasive computing environment 6

Figure 1-8 Taxonomy of computer systems research problems in pervasive computing 7

Figure 1-9 Importance of trust 9

Figure 2-1 PolicyMaker Input and Output of a query 12

Figure 2-2 Trust Chain 13

Figure 2-5 Trust Management System Architecture 17

Figure 2-6 Network Topology in the test scenario 18

Figure 2-7 Conceptual diagram of SSRD Model 19

Figure 2-8 SUKI Architecture 20

Figure 3-1 Flowchart of the constructive research methodology 22

Figure 3-3 Concept of Delegation using trust 27

Figure 3-4 Class Diagram of the access control for the University system 28

Figure 3-5 Class Diagram of Delegation Control 29

Figure 3-6 Flow chart of the trust based access control 31

Figure 3-7 Positive and negative thresholds for trust 32

Figure 3-9 Algorithm designed for the access control and delegation of services 37

Figure 4.1 Overview of the Development 39

Figure 4-2 Ad-Hoc User registration 40

Figure 4-3 The User Type as Student 40

Figure 4-4 Unable to do the changes and can only view the courses 42

Figure 4-5 A useful and trustworthy message broadcasted for recommendation 42

Figure 4-6 A User of type guest 43

Figure 4-7 The Trustworthy User Administrator Home Page 44

Figure 4-8 List of users recommendation request been received by a highly trustworthy person 48

Figure 4-9 The log containing details of the actions performed by the user 48

Figure 4-10 The Trust Values of Users after Observations and Recommendations 49

Figure 4-11 Log Containing all the Actions performed by every user who joins the sytem 50

Figure 4-12 Trust Values of User after Interactions 51

Figure 4-13 The Trust Values of the Actions with Details are Recorded in the Database 51
Figure 4-14 The Set of Delegate Rights 53
Figure 4-15 The New Delegation 53
Figure 4-16 The Privileges of the dean delegated by the admin specified with details 54
Figure 4-17 Search shows the acknowledgement of delegations 54
Figure 4-18 Arrival of an Ad-hoc user as Guest or Dean 55
Figure 4-19 Documents on the menu bar which are delegated 55
Figure 4-20 Dean has the Access to do Recommendation on Delegation of this right 56
Figure 5-1 Weight (W1) Observation against Trustworthiness 58
Figure 5-2 Weight (W2) of Recommendation against Trustworthiness 59
Figure 5-3 Weight Changing behaviour Against Trustworthiness 60
Figure 5-4 Bar Graph Showing Comparisons of Observations and Recommendation Weights with Trust Value 60
Figure 5-5 Analysis of Trusted User Actions against Trust Values 61
Figure 5-6 Ad-Hoc User's number of actions Vs Trust Value 62
Figure 5-7 Average of the Observations against Trust Value 63
Figure 5-8 Average of Recommendation against Trust Values 64
Figure 5-10 Direct Trust i.e. Observations against Indirect Trust i.e. Recommendation 66
Figure 5-11 Indirect Trust against Direct Trust 67
Figure 5-12 Trust Evolution 1 68
Figure 5-13 Trust Evolution 2 68
Figure 5-14 Comparison of Trustworthiness of Ad-Hoc and Trusted User 69

List of Table

TABLE 3-1 SET OF TRUST DEGREES……………………………………………………………………………………42
TABLE 4-1 ASSUMED SCENARIO OF CASE I W.R.T W1…………………………………………………………………67
TABLE 4-2 ASSUMED SCENARIO OF CASE II W.R.T W2………………………………………………………………..68

Chapter 1

INTRODUCTION

1.1 Introduction

The word pervasiveness means ever present or always occurring. The ubiquitous or pervasive or sentient or ambient intelligence or percom are the one and the same terms used in the same context [21]. The tomorrow's world will bring the ease in the real world which involves the convergence of the technologies and enhancing the comfort of the routine life towards a better way of life [1].

Ubiquitous computing was introduced by Weiser as he introduced the new world of ubiquitous [2]. The pervasive computing environment immensively involves human-interactions and computations. It requires the systems to be embedded invisible in the environment so that the user can only concentrate on the task, being unaware of the underlying technology [25]. The basic objective is to bring the technology in the human world for the ease in daily routine work. Even when the user is on move, the services remain pervasively and invisibly available involving seamless interactions among portable and networked processing devices, distributed at all scales in the everyday life. Decentralized & distributed, openness & dynamism are some of the characteristics of the pervasive computing. Figure 1-1 briefs a pervasive computing scenario.

Figure 1-1 Smart devices of pervasive environment

The future technologies expected to prevail in the pervasive computing environment are wearable computers, smart homes and smart buildings. Among the countless tools expected to support them are: application-specific integrated circuitry (ASIC); speech recognition; perceptive interfaces; smart matter; flexible transistors; gesture recognition; system on a chip (SoC); reconfigurable processors; field programmable logic gates (FPLG); and microelectromechanical systems (MEMS) [28].

Pervasive computing is specified by the four factors stated as [3]

- **Effective Use of Smart Spaces**

  Smart spaces defines the boundary of an open place to combine opponent entities together until now E.g. The automatic regulation of the cooling, heating effects in a room based on an inhabitant's electronic profile [3].

- **Invisibility**

  Invisibility aims to fulfill user expectations with minimum user surprises. Weiser expressed invisibility as, total unawareness of the user from the technology [31]. E.g. the user is automatically logged into the system on sensing, identifying and authenticating the authorized individual [3].

- **Localized Scalability**

  Localized scalability increases the complexity by increasingly large number of users and the interactions between the human's and the personal computers [3].

- **Masking Uneven Conditioning**

  It states the complete invisibility though impossible to achieve entirely, but reduced variability can be strived for. This large dynamic range of smartness can be very high to a user, detracting for making pervasive computing technology invisible [3].

Ubiquitous Information:

The information being shared, manipulated, distributed, consumed and archived is the power of ubiquitous access which forms the basis of the praiseworthy success and inexpensive utilization of web across the world.

There are two areas of interest here.

- ➢ First, what application areas are enabled by ubiquitous information access? What impact, for example, does it have on education or health care or collaborative research or in short in real life?

- Second, how to distinguish different applications? One possibility is to separate them on access privileges to the information, ranging from personal to global access.
  - Personal access for individuals taking notes for their own use.
  - Within a work group trusted access for colleagues.
  - Within a family unit, or global access intentioned for public consumption.

These are illustrated in figure 1-2.

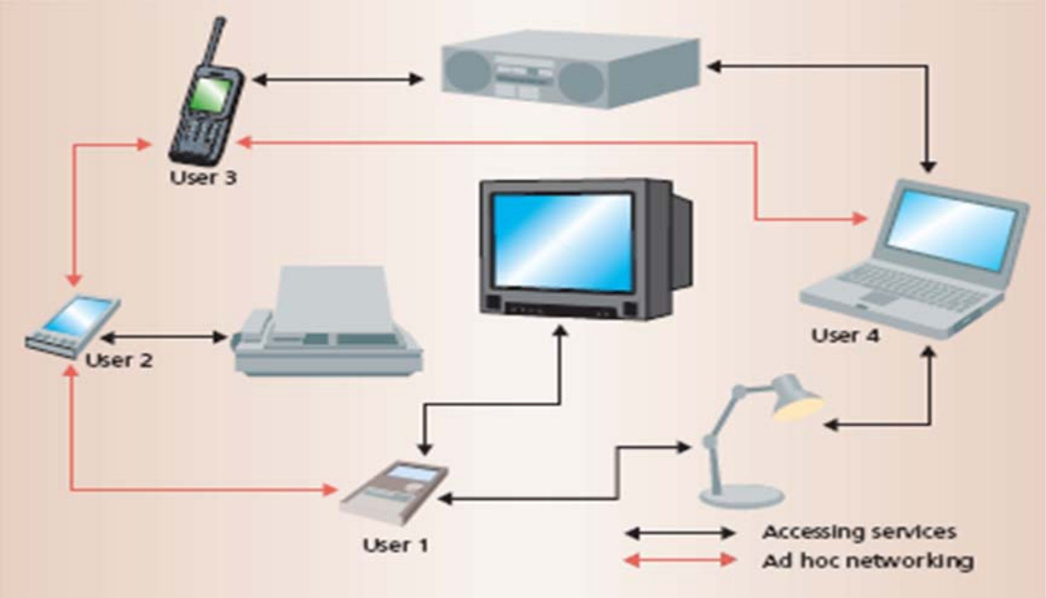


Figure 1-2 Pervasive computing environment. Mobile devices connected via wired and wireless links to one another and to the global networking infrastructure providing information and integrated services.

Ubiquitous Service

A software component which performs computation or activity on behalf of a system entity is called ubiquitous service. This entity can be the user or device or any other service. Services are usually well-defined in their functionality along with their defined inputs and outputs [4].

Five goals of ubiquity, with regard to a service:

- **Availability (**User needs, User preferences or status)
- **Transparency** (focus on task not on tool; hides Underlying technology)
- **Seamlessness** (everlasting service session, user recognition)
- **Awareness** (system awareness of its user i.e. context and his/her task which enhances the user's decisions)

- **Trustworthiness** As depicted in the figure 1-3:

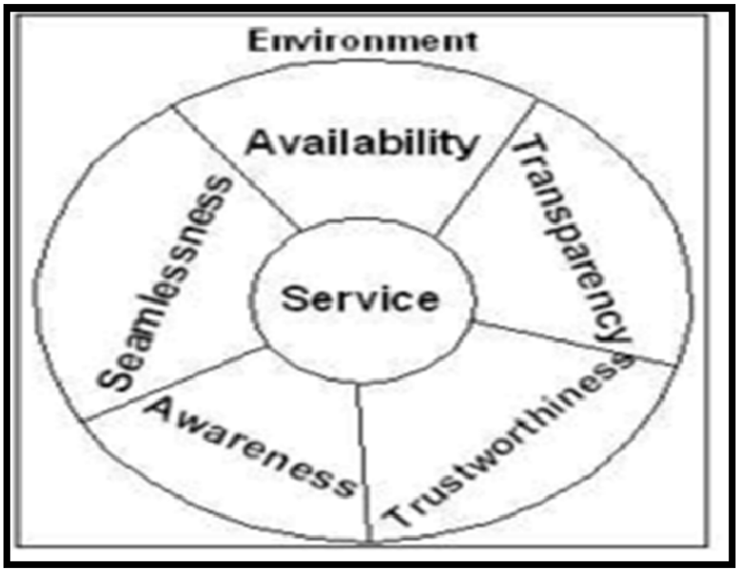


Figure 1-3 Ubiquitous Service Model

1.2 Challenges Of Pervasive Computing

Research in pervasive computing has focused on developing applications that enable users to collect, communicate, save, organize, and reuse information. Pervasive services are provided through countless invisible devices embedded in the user environment that might work relatively or personal. This continuous information collection exposes personal activities, habits, preferences, loath and associations. Privacy and security of this information has not been given enough consideration. In addition pervasive computing applications rely heavily on mobile and wireless communications that brings up new privacy issues. Privacy sensitive information is available to pervasive service providers continuously, thus making it difficult to protect it.

Due the characteristics described in section 1.1 the pervasive environment is exposed to a number of open issues. Some challenges faced by Ubicomp Environment are

- Authentication
- Transparency and Unobtrusiveness
- Multilevel Security
- Context-Awareness
- Flexibility and Customizability
- Interoperability

- Extended Boundaries
- Scalability
- Provide different levels of security services based on system policy, services, context information and available resources.
- Insufficient Privacy Response
- Changing Environment i.e. dynamic behaviour
- Private Information Retrieval
- Avoiding Privacy Violation for Resource Sharing

Passwords, Biometric, Digital Certificate and Pseudonyms are the four authentication mechanisms as shown in figure 1-4. New authentication mechanisms and devices keep on evolving to address these issues. Transparency and unobtrusiveness requires biometric authentication and context awareness.

Figure 1-4 Ubiquitous environment involving authentication mechanisms

The above issues cannot be dealt with the limitations of traditional security solutions like [5]:

- Client-Server Architecture
- Centralized and Static Access Control Policies.
- User Interactions (Login, logout, file permissions etc)
- Context In-Sensitive
- Non-Adaptable

Therefore some authentication methods for the pervasive computing are [5]

- Passwords
- Biometric as shown in figure 1-5

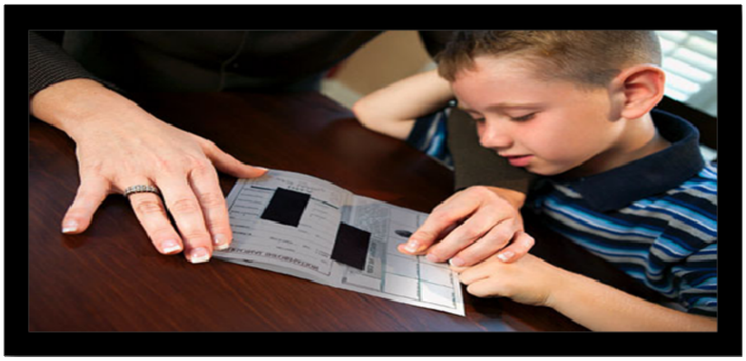


Figure 1-5 Biometric Authentication

- Digital Certificate as displayed in figure 1-6
- Pseudonyms

Figure 1-6 Digital certificate

1.3 Security issues of Pervasive Computing

The security issues of the ubiquitous environment can be well interpreted by the figure 1-7

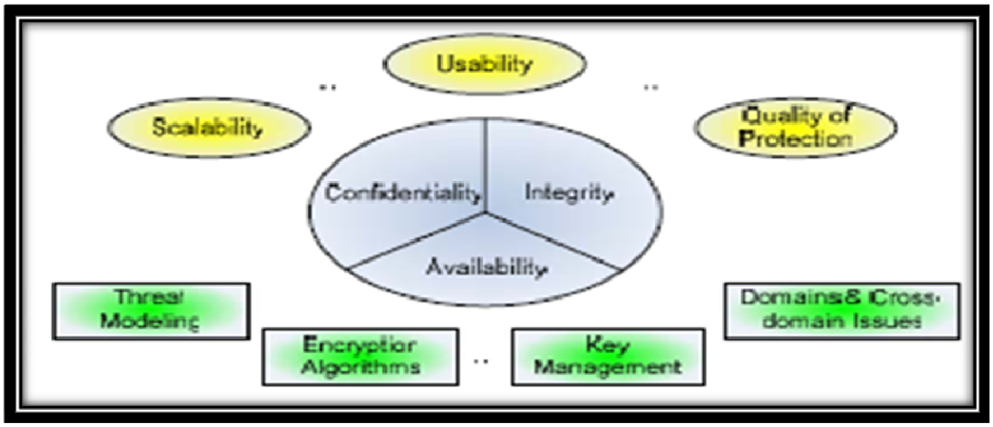


Figure 1-7 unique challenges and requirements for security insurance in a pervasive computing environment

Moving from left to right in this figure 1-8 new problems arises which are tracked down [3]. The problems evolves as the evolution goes from simple distributed computing via mobile computing towards pervasive computing and making the existing problem more complex [3]. It is more intricate and complex to design and develop a pervasive computing system than a distributed system [3]. Thus the increase in complexity is multiplicative rather than additive as seen from modulation of figure 1-8. Although research evolution has been described in this figure with respect to time but in some scenarios a few research aspects of pervasive computing began relatively early. For example, work on smart spaces began in 1990 and proceeded somewhat independent of the processing's in mobile computing [3]. Figure 1-2 shows a sight into the ubiquitous computing environment.

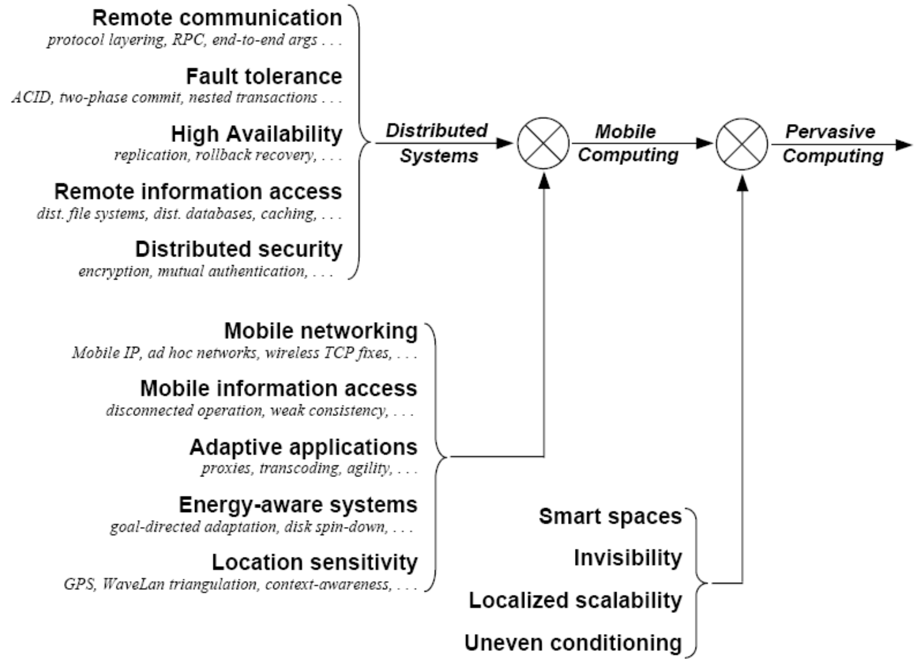

Remote communication
protocol layering, RPC, end-to-end args . . .
Fault tolerance
ACID, two-phase commit, nested transactions . . .
High Availability
replication, rollback recovery, . . .
Remote information access
dist. file systems, dist. databases, caching, . . .
Distributed security
encryption, mutual authentication, . . .
Distributed Systems
Mobile Computing
Pervasive Computing
Mobile networking
Mobile IP, ad hoc networks, wireless TCP fixes, . . .
Mobile information access
disconnected operation, weak consistency, . . .
Adaptive applications
proxies, transcoding, agility, . . .
Energy-aware systems
goal-directed adaptation, disk spin-down, . . .
Location sensitivity
GPS, WaveLan triangulation, context-awareness, . . .
Smart spaces
Invisibility
Localized scalability
Uneven conditioning

Figure 1-8 Taxonomy of computer systems research problems in pervasive computing

1.4 Trust as a security paradigm:

Trust is stated by Sociologist Diego Gambetta as:

> *"Trust is a particular level of the subjective prob. with which an agent will perform a particular action, before [we] can monitor such action and in a context in which it affects [our] own action"*

With the rapid growth of global digital computing and networking technologies, trust becomes an important aspect. The idea applied is to transform the human trust mechanism to the digital world [5]. Digital trust mechanisms must work well in a dynamic, interactive environment [5]. Computational trust applies the human notion of trust into the digital world.

The important issue with reference to the access control of the ubiquitous information is to whom the access should be granted. There are four kinds of access control:

- **Role-based Access Control (RBAC)**

  It can not enforce fine-grained access control in a context-aware environment where the roles are not predefined and the access rights and constraints are dynamic.

- **Context-based Access Control (CBAC)**

  A scalability problem arises in which a vast amount of resources needed to be monitored and adjusted permission regularly.

- **Policy-based Access Control (PBAC)**

  Policies are not dynamic and cannot accommodate the changing security requirements of a context aware system with time.

- **Trust-based Access Control (TBAC)**

  TBAC is not scalable and flexible for making decision if role is not integrated as well. This is because there are a huge amount of entities in the ubiquitous environment and such entities are usually different from others. Trust basis is being explored largely by the researchers overall. In our everyday life, actions of the human being are dependent on the trust as it is established between the entities. With the passage of time the level of trust in a relationship varies, as it continues to evolve being well interpreted by the figure 1-9.

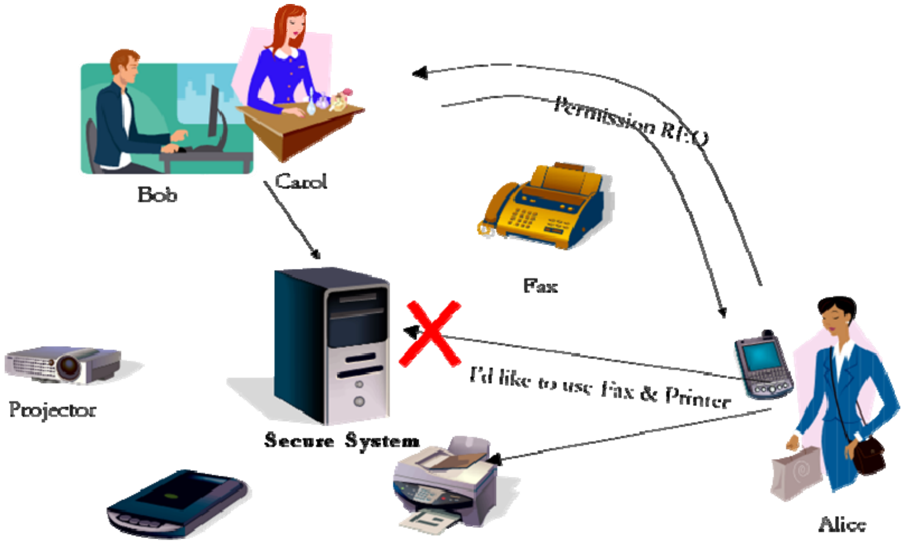


Figure 1-9 Importance of trust

1.5 Problem statement

Create a distributed trust based access control architecture for a pervasive environment which

- Requires an ad-hoc authentication of users on the basis of trust categories along with access control and delegations.
- Grant user access rights with respect to their relationship and context.
- Accommodate the unknown user requests to access information/services while maintaining the privacy and the security concerns of the users.
- Determine the access level of the user before authorization of actions through dynamic trust calculation.

Design a web based framework for the user registration system of the university running academic programs under the semester system while automating the user tasks, thus reducing the need for a centralized and manual administration.

1.6 Objective

The objective of the thesis is to have an idea of the research ongoing towards the new evolving technology and the tomorrow's world of ubiquitous. The pervasive world aims to be the next generation where user will be at ease to fulfill its tasks and overall provides convenience to carry out the everyday tasks. The challenges faced in this world are still partially unexplored and intricate. The trust mechanism is introduced in the distributed system to develop a flexible approach in order to fulfill the requirements of the dynamic and ad-hoc pervasive environment. The architecture is devise for the trust based access control which is developed into a web based framework for the user management system of the university.

The distributed trust based access control architecture is proposed for a pervasive environment which works for the ad-hoc authentication of users, assigning trust categories and maintaining access control and delegations so as to avail a service. Objective is to automate the user tasks and reduce the need for a centralized and manual administration. The evolution of this architecture resulted after studying the established trust based access control mechanisms with their pros and cons. After development, the aim is to perform the thorough analysis of the developed prototype on the basis of observation, recommendation and previous trust values which determines the trust level for the user.

An automated distributed trust model is devised for the user access to the service to do an action on the basis of dynamic trust calculations with respect to the interactions, recommendations and time-based delegation which forms the basis for the authentication and authorization of the ad-hoc user.

1.7 Thesis Organization

Overall, the thesis division involves chapter 2 of literature review and related work. Chapter III provides the design specifications of the proposed architecture and related system designing. Chapter 4 discusses implementation of the architecture, various procedures and the screen shots of web based framework whereas chapter 5 illustrates the analysis of the framework and comparison with the related models done and chapter 6 summarizes the research work and states the future work.

1.8 Conclusion

The brief introduction to the world of pervasive computing environment is compiled in this chapter. The issues, challenges and the various mechanisms to provide controlled access to the information maintaining anonymity and privacy of the user are shortly described. The problem domain and the objective to achieve are enclosed finally.

Chapter 2

LITERATURE REVIEW

2.1 Introduction

The pervasive computing system requires security architecture on the basis of trust rather than just user authentication and access control [8]. However, no solid model was given [8]. Trust in the computer science dictionary has no accepted specific definition and classify trust as a discrete degree, a number or a combination of both [18]. Trust is an indefinable concept and its lack of definition opens the trust models to individual interpretations and incompatible protocol implementations [19]. Trust in keynote management system was introduced for the management of security related to policies, credentials and trust relationships by a public key infrastructure [6, 7].E.g. In authenticating the identity credential of a previously unknown user which lacks any access control information. Trust provides the basis for securing ubicomp systems, but has no consideration in majority work. A little work has been done for the security of the ubiquitous system. Developed mechanism of security does not provide enough security services which mainly focus the authentication and the access control role of the user. Traditional authentication mechanisms are inapplicable for ubiquitous systems because it requires predefined users, e.g. username/password, public/secrete key, etc whereas Ubicomp has un-predefined entities. Ubicomp include various and numerous devices with different authentication methods.

The most prominent trust management systems for access control are:

2.2 Keynote, Policy Maker, Referee

2.2.1 PolicyMaker [1996] developed by Matt Blaze

[14] PolicyMaker unites public keys to predicates. The keys are assigned the actions which should be performed describe for the predicate which makes it best for the anonymity. The PolicyMaker system can be thought of as a form of database that the application queries for answers of the questions of the type: “Can this action be performed by the key according to the defined local policy”. PolicyMaker aimed to provide a framework for a wider range of applications. It uses an Action Environment which is passed from the application to the KeyNote system. The Action Environment consists of:

- The security policy
- A list of credentials
- The request

Once passed through Action Environment, KeyNote returns an application type specific string, which in the easiest case could be "Action authorized". It takes a decision whether action could be successful because it does not imposes the policies upon the system but do gives recommendation to the applications that ask for it [14].

2.2.2 The KeyNote

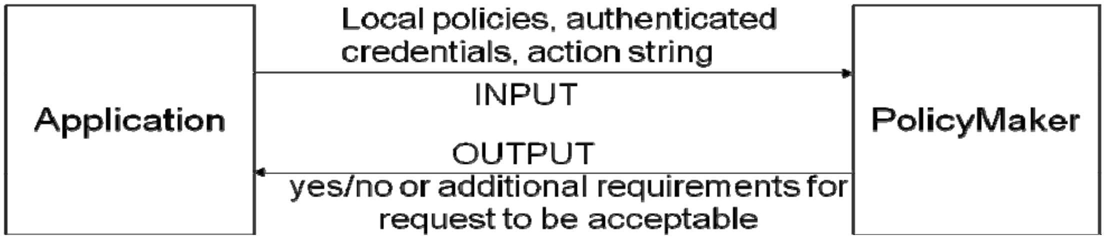

Application
Local policies, authenticated credentials, action string
INPUT
OUTPUT
yes/no or additional requirements for request to be acceptable
PolicyMaker

Figure 2-3 PolicyMaker Input and Output of a query

[7] KeyNote [1998] trust management system is a direct inheritor of the PolicyMaker system. It is also developed by Matt Blaze and shares the same idea as PolicyMaker. One of the main differences between PolicyMaker and Keynote is that KeyNote has a comparatively simpler syntax and semantics designed to have public-key infrastructure applications. PolicyMaker structure achieves for a wider range of applications.

2.2.3 Advantages & Disadvantages:

Behaves like a database query engine to the application as shown in figure 2-1. Do not directly enforce policy; they only provide recommendation to the applications that asks for it [15]. The pros & cons are well predicted as they are the initial systems.

2.2.4 REFEREE

[16] The REFEREE (Rule-Controlled Environment for Evaluation of Rules and Everything Else) system is from 1997. The system differs from previous two in the way that it helps in making access decisions concerning web sites. It functions as an

engine that can be queried for recommendations. The answer to a query can be of any three forms true, false or unknown. Unknown means that the system was not able to make a decision about whether the requested action could be recommended using the policy that was in question. In such a case the calling application needs to determine which action should be taken. All trust decisions are controlled on the basis of policy [16].

2.3 Trust Based Delegation System

(Trust Based Delegation System for managing access control) [10] Architecture of trust based delegation system is based on the trust relationship that exists in the native world and mapping it to the digital world. In this system a token is generated from a trust source & it propagates through trusted users until it is handed over to the desired service. Token contains explicit information about the access to the resource. A user can pass a copy of token to another user on which he/she trusts & before forwarding the token further in the chain, the sending user can alter the access rights which are based on the degree of trustworthiness of User A on to User B as illustrated in figure 2-2. Tokens can be copied and restricted in a chain. Tokens contain a digital secret which is known only to services and trust source.

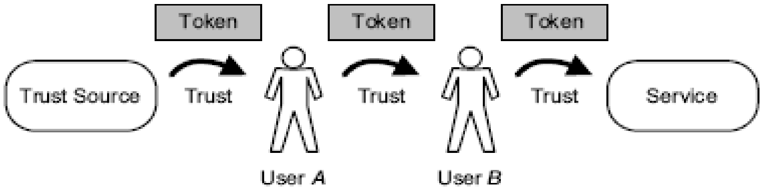


Figure 2-4 Trust Chain

Service invoking requires delivering tokens to the service which decides autonomously whether service should be granted or not by deciphering the token until secret key is visible on the basis of last permission. Token contains electronic information about access privileges & permissions which are coupled with constraint like time, situation etc [10].

Advantages

- Secure transmission of token as whole trust chain is secured by cryptographic measures whose integrity is checked before providing service.

- Supports high usability, user convenience, avoids malicious users (detected by checking permission update list in service against the revocation token which limits the permission).
- Supports anonymity of users
- Supports dynamism.

Drawbacks

- Do not provide strict restrictions on delegations of access rights.
- Lacks stronger context awareness.

2.4 Distributed Trust management [DTM]

[11] A Framework for Distributed Trust Management demonstrated that each group of agent is id protected by a special Security Agent (SA) which is trustworthy. An agent can be a process or a user or a device. Each SA can access services/resources within that group. Delegation is kept with SA which validates the authorization of requester & determines whether the delegation is according to their policies. SA forwards the request to the resource held responsible for the service else if request is not valid according to SA then an error message is generated and sent to the requesting agent. SA is ignored if the resource holder is powerful enough to investigate the authenticity of the user so the requester can be entertained directly. Delegation can occur only if the agent has the right to delegate. Requester if not allowed to access the resource has been delegated the access right by:

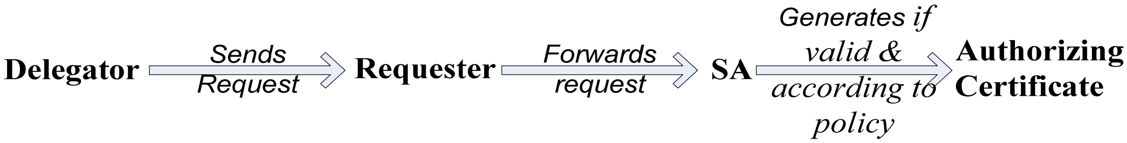


Figure 2-5 Summarizes the flow of access control in DTM

Authorizing Certificate decides whether to grant access to the resource if authenticated and conforms according to the policy else deny the request as shown in figure 2-3[11].

Advantages:

- Overall constraints exist on access rights, delegation & redelegations which depend on security policy, access rights of the agents and rights of the agent in the delegation chain.
- Delegations can be forwarded to groups.

Drawback:

A security agent should return a list of rules that it used for taking the decision, in case of failure of the authorization process. This allows the requester to figure out where its credentials failed and afterwards makes it able to correct the faults [11].

2.5 TrustAC: Trust-Based Access Control for Pervasive Devices

[12] Since, in open dynamic environments it is very difficult to depend on central server for all the processing's and computations. So, in this scheme the access control for a ubiquitous environment is based on a pervasive trust management model from which trust degrees are dynamically obtained. It's a drift from a central controller to distributed trust access control. Thus an access control scheme is discovered based on trust called TrustAC. Here, access control decisions are based on trust because in these scenarios (open and dynamic environments) the users are peers, there are no roles and preconfigured access control lists (ACLs), or previously deployed infrastructures. Likewise, the trust is subjective and changeable, each user stores his/her own trust values, without depending on third parties to guarantee user security; therefore, at anytime we know the user's trustworthiness to grant or deny him/her the access to our resources.

Each user is autonomous to establish his/her trust thresholds for granting access. This is supported by the underlying trust management model, PTM, which offers us trust values in a dynamic and automatic way, minimizing the user intervention.

On the other hand, the use of numeric values allows establishing categories to the access of the resources, instead of the individual basis. The trust values can be used in several domains, by allowing the interoperability among them unlike roles or explicit permissions. TrustAC takes into account the environment conditions (context) to define the access control policies, being very important for mobile users. TrustAC specification includes a Reference Model defining sets of basic TrustAC elements and relationships among them, and a Functional Model defining the features required by the system as described in figure 2-4. So, TrustAC simplifies the access control management for large number of users because it allocates permissions to trust degrees rather than individuals, and there are fewer trust degrees than users being its great advantage.

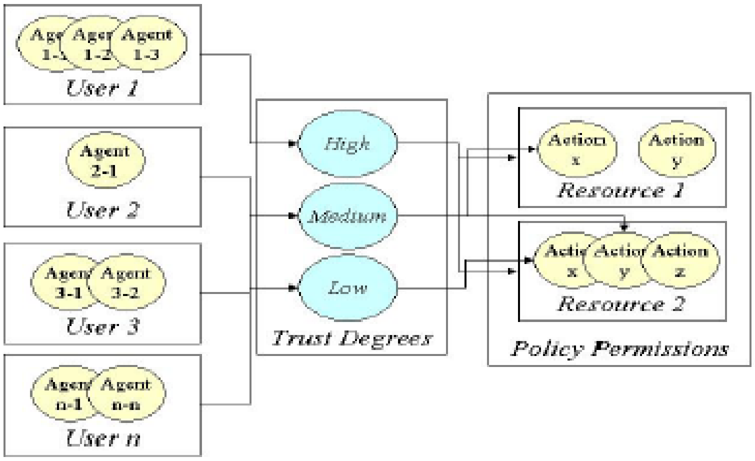


Figure 2-4 TrustAC Reference Model

Limitations

- ✓ TrustAC simplifies the access control management for a large scale because it allocates permissions to trust degrees, rather than users.
- ✓ Unreliable because no involvement of recommendations and malicious devices.

2.6 Trust Distributed Trust Management System Architecture

[13] Proposed scheme of trust management possesses some pervasive specifications stated as flexibility, simplicity and distributed operations, updatable trust values. The architecture consists of:

Trust value computation

Each pervasive device maintains a trust value in the same environment. Each device also maintains trust values of the other devices in the network which are active. Each trust value maintained on the device may differ from the others for the same device. Communication may start only if the trust value (TV) is above a certain threshold shown in figure 2-5. If an unknown device wants to communicate, it must be assessed trust worthy. The assessment can be made in two ways:

- Direct Observation by a function of $O_A(B)$
- Indirect computation.

If enough trust information cannot be gathered by the direct observation then combination of above & recommendations from trust worthy devices in the same environment i.e. $R_A(B)$.

Update & maintain trust value

Trust values are updated (deleted / updated) for reliable & safe communication. Updation occurs on the basis on Previous trust values & current behavior as $C_A(B)$ keeping in sight the context of the situation. Purging Module deletes the irrelevant trust values which are of no more interest to the system thus avoiding the overhead of maintaining trust values in the system.

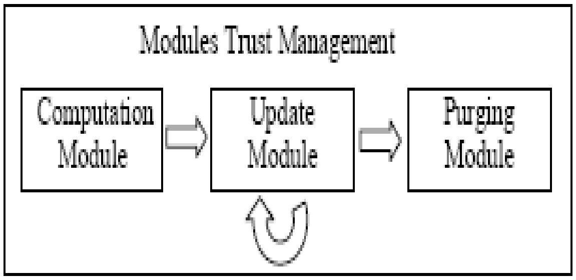


Figure 2-5 Trust Management System Architecture

Advantages

- Use of recommendation always makes the communication reliable.
- Trust values are dependent on the weights which are assigned according to application environment.
- Malicious devices recommendations are avoided.
- Trust values are dynamically calculated & updated to avoid overheads.

Drawbacks

- This scheme is applicable to very specific scenarios because no roles or trust degrees are defined. The device after proving to be trustworthy should have full access to all the services which should not be the case.

2.7 Middleware Architecture for Mobile Respective Pervasive Environments

[17] A modular architecture based on service oriented approach fabricates a flexible composition of systems with a heterogeneous character both at client- and server-side. The wired and wireless networks are combined to provide mobility of users and client devices. The problem of dynamic IP-addresses or re-authentication of users are

regulated by the additional task of establishing layer which forms a virtual path in order to maintain stability in a connection during handing over of the technology. The user is unaware of these changes as all the user task is only to accept/decline the change. The distributed modular approach consists of three basics layers which include access, service and data layer.

USB token being more powerful hardware is currently being developed by the embedded processors and low-power memory. In the first test scenario shown in Figure 2-6 where a primary use case is secure communication of wireless client devices on a centralized infrastructure for eLearning applications. The goal is to hide the underlying complexities of the architecture and enabling user control over security transactions. Finally, this leads to a Pervasive University, where services based on individuality and location-awareness can be securely, accessed anytime and anywhere on campus.

Advantages:

- The system modularity allows a flexible composition of the required functionality to be based on web services like a powerful middleware.
- A well-promising, available solution from the common security mechanisms is CryptoToken, which allows a single-sign-on using the USB interface.

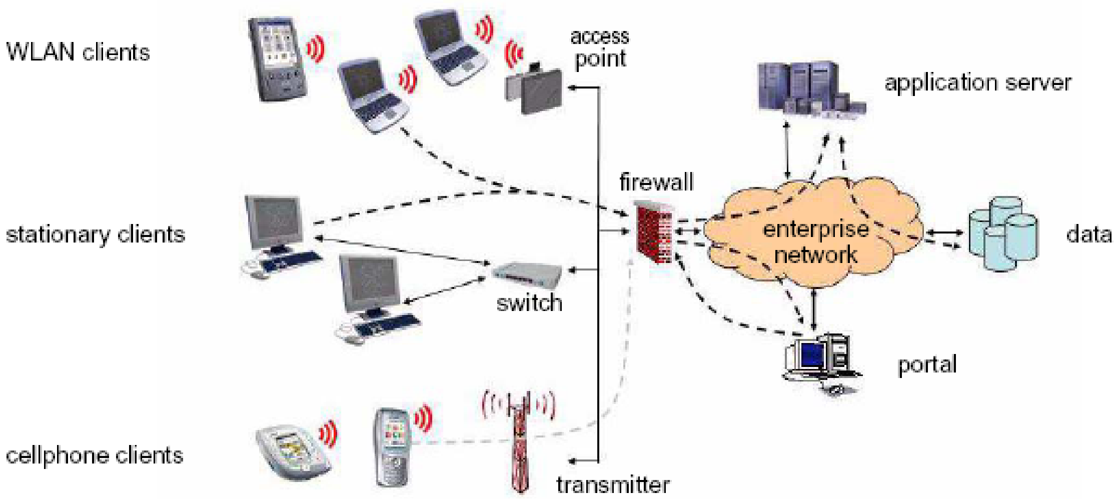


Figure 2-6 Network Topology in the test scenario

2.8 An Adaptive Lightweight Trust Reliant Secure Resource Discovery

This architecture establishes a mechanism for securely discovering resources for the devices in the pervasive computing environment. Service-oriented adaptive security mechanism named SSRD (Simple and secure resource discovery) trust model is also proposed. This model has discovery model which is lightweight and mobile devices are capable to handle the computations by themselves in figure 2-7. Keeping in mind the performance should not degrade, the mechanism allows discovering resources but securely. A range of trust values are defined in which initially each device is assigned a supposedly trust value. This value changes with the changing behavior and the context of the devices. A device ‘A’ trusts another device ‘B’ depending on some service which has some trust value like T(As,B,z) where z is between (0.0 <= z <= 1.0). So the security levels are defined for each service. The trust model is dynamic and accepts changes. It allows any ad-hoc device to gain service but also checking malicious devices to deny access to services based on the mechanism defined in the trust model.

An application prototype is implemented for the proposed scheme using test beds of dell pocket PC. The battery powerful consumption for SSRD is nominal as the power is calculated before and after running the prototype. So, it is designed to work in everyday circumstances without compromising device performance. This prototype is extended for implementation on smart phones.

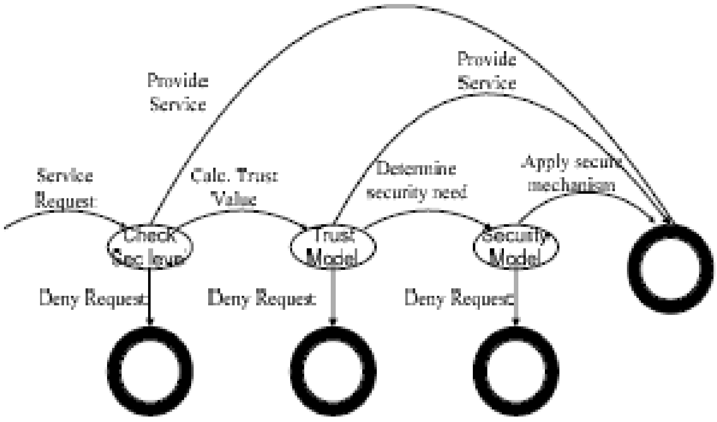


Figure 2-7 Conceptual diagram of SSRD Model

2.9 Evaluating Trust-based Access Control for Social Interaction

[27] The world of ubiquitous involves the interactions of the users between them and their surroundings immensively. This paper aims to diagnose evaluation of TBAC models in ubiquitous computing environments. This framework compares and evaluates several trust based access control models by simulating a number of well

known security attacks including the Sybil ones in peer to peer systems. In ubiquitous environment the advertisement messages are delivered according to user choice, fondness or shopping lists in more personal ways. Trust based access control (TBAC) requires access to the resources based on the trust values owned by entities. Trust evolves depending on the user behavior and the surroundings. There are many evaluation mechanisms but there is no standard to evaluate a trust based access model. Usually they are evaluated against some scenarios executed in experiments to verify that it goes well and behaves accordingly in the set situation. The evaluation of models in different experiments makes it impossible to compare the alternative models. A novel based scenario evaluation framework is proposed which automatically compares the strengths and weaknesses of the model under testing in defined scenarios and perimeters. SUKI Architecture is shown in figure 2-8

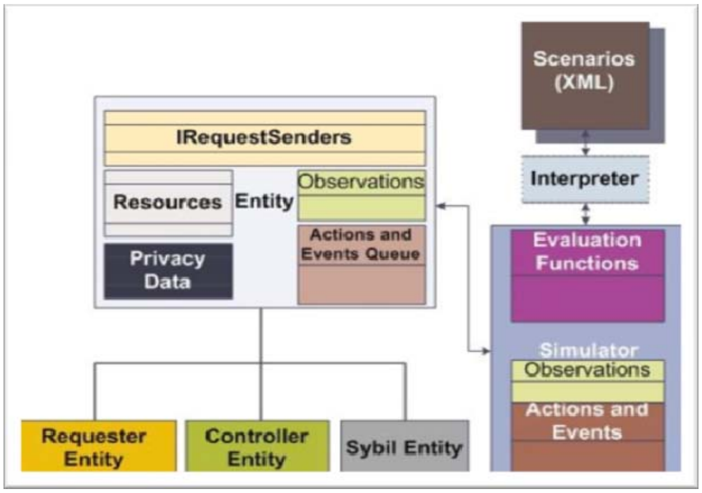


Figure 2-8 SUKI Architecture

2.10 Conclusion:

The schemes relative to the trust based access control, delegation control and the trust categories for the pervasive computing have been studied. The advantages and the drawbacks for the schemes are also listed. The techniques specify that it is a diverse field which has still many unexplored dimensions.

Chapter 3

PROPOSED MODEL DESIGN

3.1 Introduction

This chapter introduces the basis for the research which includes the layout of the steps involved in the scientific research, detailed description of the problem thorough flowcharts, sequence diagram and class diagram, the system architecture/design and algorithm of the proposed system and thus narrowing down the widened area of the research domain into actual problem domain. Each minute detail is included to explain the functioning of the system. Resources or documents or services are the interchangeable terms used in the same context.

3.2 Research Methodology

Research can be defined as the search for knowledge or any systematic investigation to establish facts and reasons for the question in focus. Scientific research is one way of the research process.

Research is scientific or critical exploration aimed at determining and understanding facts [22]. Scientific research relies on the application of the scientific method. The objective of the research process is to produce new knowledge, which may take the three main forms.

- Constructive research develops solution to a problem.
- Exploratory research structures and identifies new problems.
- Empirical research tests the feasibility of a solution using empirical evidence.

  This research is carried out using constructive approach because it aims at producing novel solutions to practically relevant problems. It solves the managerial problem through the construction of models, diagrams, plans, and organizations.

3.2.1 Stages of Constructive Research Methodology

Constructive research methodology which is used for my thesis involves assessing the construct being developed analytically against some predefined criteria or performing some benchmark tests with the prototype. Construct can be a new theory proposed, algorithm, model, software, or a framework [22].

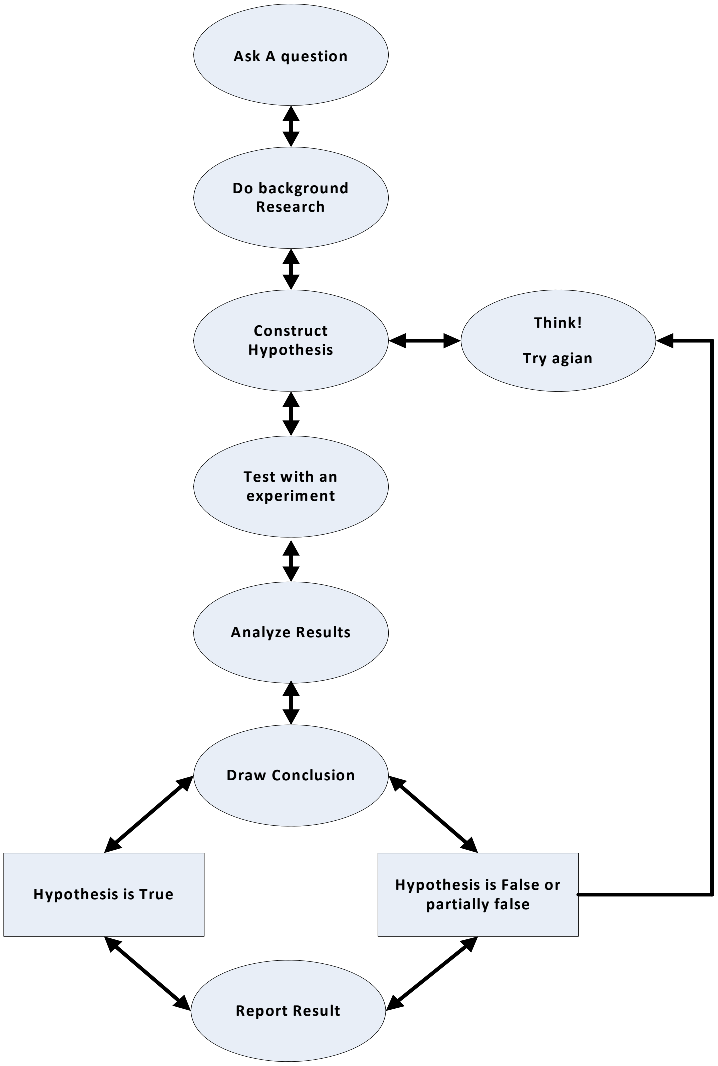


Figure 3-6 Flowchart of the constructive research methodology

i. *First stage*

It involves identifying a problem and understanding its various aspects. The problem should be practically relevent. To identify and then understand the problem, a survey of literature has been carried out. A number of relevent research papers and research journal have been studied to find the existing security problems and challenges faced by the ubiquitous environment as illustrated in figure 3-1 [20]. Trust based access control in pervasive computing has been on rise in the field of research nowadays regarding security of the system and immense work is needed to be done on it towards the privacy and security control of the ubiquitous envirnment. Trust which calls reliability trust and decision trust respectively will be used in this study. As the name suggest, reliability trust can be interpreted as the reliability of something or somebody, by Gambetta (1988) [30]. Decision trust is to take a decision on the basis of reliability trust.

ii. *Second stage*

A discovery of novel idea which is practically implementable is the next step. In this research work, a hybrid approach is proposed based on previous works of Trust-Based Access Control [Trust AC] of pervasive devices which allocated a group of user to the resources, A distributed trust management scheme [DTM] allows the computation of trust metric from observations and recommendations through mathematical formulas and a framework for Distributed Trust Management [FTDM] in which concept and rules for the delegations has been solicited. A hybrid scheme is constructed consisting of the above defined mechanisms to overcome the existing problems of unsafe and unsecure authentication and authorization to the resources discovered in the previous stage.

iii. *Third stage*

This step include implementation of the proposed approach and the comparitive analysis of the defined schemes. Finally the novelty and theoritical contribution is evaluated by the performance evaluation. The web based prototype is evaluated by comparing the trust levels of the good and bad actions in chapter 5.

3.3 Design Specifications of the Proposed Architecture

Introduction

The proposed model has the following properties which were considered while designing the architecture. These properties relate to the properties of the ubiquitous environment and are possessed by the proposed architecture.

A. Dynamism

The trust values and the trust categories changes with respect to the changing behavior of the entities and the rules of the system. It also requires updating the degree of trustworthiness of the entities periodically in a session.

B. Category-Based Services

The security mechanism provides services with respect to the category of the entities. The system is designed to secure services while concealing the privacy of the user. The user's actual actions are evaluated in their domains regarding the access of the resource.

C. Trust-Specific Authorization

The architecture uses trust mechanism as it exists in the real world. The access to the services and actions are permitted based on the trust information available in the network. Trust encounters the uncertain nature of the pervasive computing. In real world trust owns some properties such as always changing. It is based on policies defined in the organization, is context dependent and provides a foundation to build good-will and bad-will of a person. These properties make it ideal to be used as a security mechanism in the ubiquitous world. All these properties of trust are being applied in the development of our proposed architecture.

D. Resource-Constraints

The proposed trust based access control mechanism is designed to be simple while considering the limitations of the memory requirements and other constraints of the mobile devices. So its purpose is to avoid the data redundancy.

E. Distributed actions

The actions of the entities are distributed in order to avoid workload on the central computing server because the authentication and trust calculations are made on the basis of the existing observations and other trustworthy devices recommendations.

3.4 Description of Proposed Architecture

The proposed architecture is a hybrid model aiming for the more secure and safer access control to services in a pervasive computing environment. Basically it consists of four modules illustrated in figure 3-2. Modules are

- **Trust Computation Module (TCM)**
- **Trust Category Computation Module (TCAM)**
- **Access Control Manager (ACM)**
- **Delegation Control Manager (DCM)**

3.4.1 Trust Computation Module (TCM)

TCM allows autonomous trust computation without user involvement on the basis of following

- Node’s interaction history produced by the interaction with the server which is called direct communication.
- By a method of recommendation called indirect communication. In the proposed scheme each entity moves forward in the network with the defined trust values.

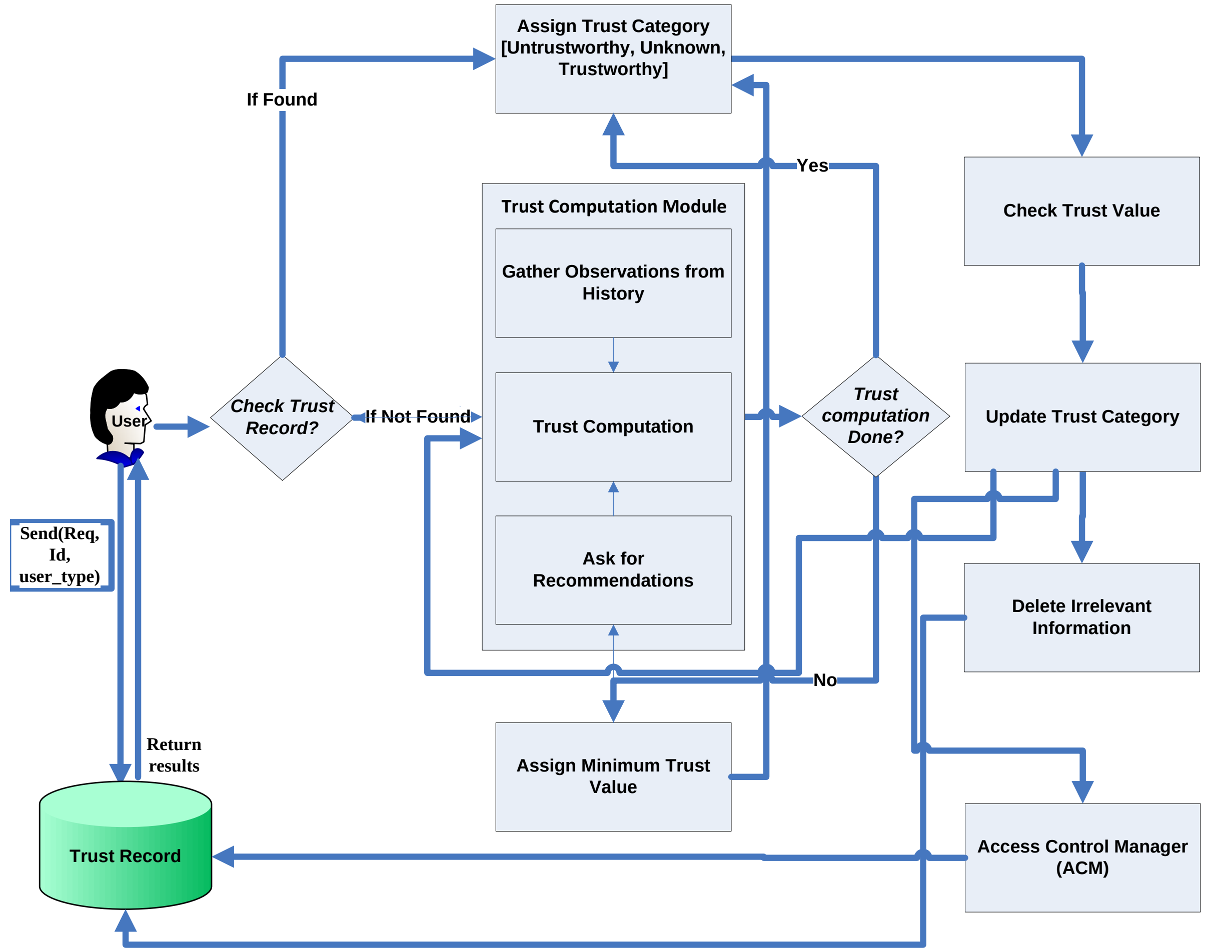


Figure 3-2 Architecture of the trust based access control

Each device keeps the trust value of all other active devices which have performed some interaction in the network. The formulas used in this scheme are derived from the distributed trust management scheme [13]. These formulas are applied on the observation and recommendation value to get a single trusted value. The recommendation request is broadcasted to a list of only trustworthy entities. The motivation behind having a trust model is to avoid providing services to malicious devices but at the same time want to avoid the denial of requests from the legitimate devices. The ad-hoc user whose trust cannot be computed by TCM or found in trust record is categorized as ‘unknown’ which is then allowed to register its detail in order to access the service but with minimum trust value and privileges.

Conventional authentication and access control methods require much user interaction in the form of logins, logouts, and file permissions which are done manually [9]. These manual interactions go against the vision of autonomous ubiquitous computing [9].

3.4.2 Trust Category Computation Module (TCAM)

Trust values calculated, are processed to compute the trust categories. Instead of allowing trust values to access the service, trust categories are allocated for each user trust value. So, in this way single trust category could be assigned to multiple entities which are further proceeded to query access policies to perform an activity or access a service. Overall, this strategy benefits the system by allocating a category to a group of requests [6].

3.4.3 Access Control Manager (ACM)

ACM defines a procedure to validate each action whether entities shall be granted access or not, in order to perform the actions they requested. The user actions are checked against a run time procedure called for the validation of the action. Once authentication has been validated, the access control process checks that if the entities are authorized to perform the action and manages an access control log for the users. Access control log contains result for access control action, delegations and authentication details associated with them.

3.4.4 Delegation Control Module

Pervasive computing provide user with the services anytime anywhere without any priori registration thus minimizing the administration overhead [25]. Thus the concept of delegation came which requires delegating all or subset of rights to the user or a group on which the delegator trusts with respect to the time constraint as best depicted by figure 3-3. These are the recipient which can use the services on behalf of the delegator. The actions to these services by the recipients are maintained in the log file and the services remain available till the time out occurs.

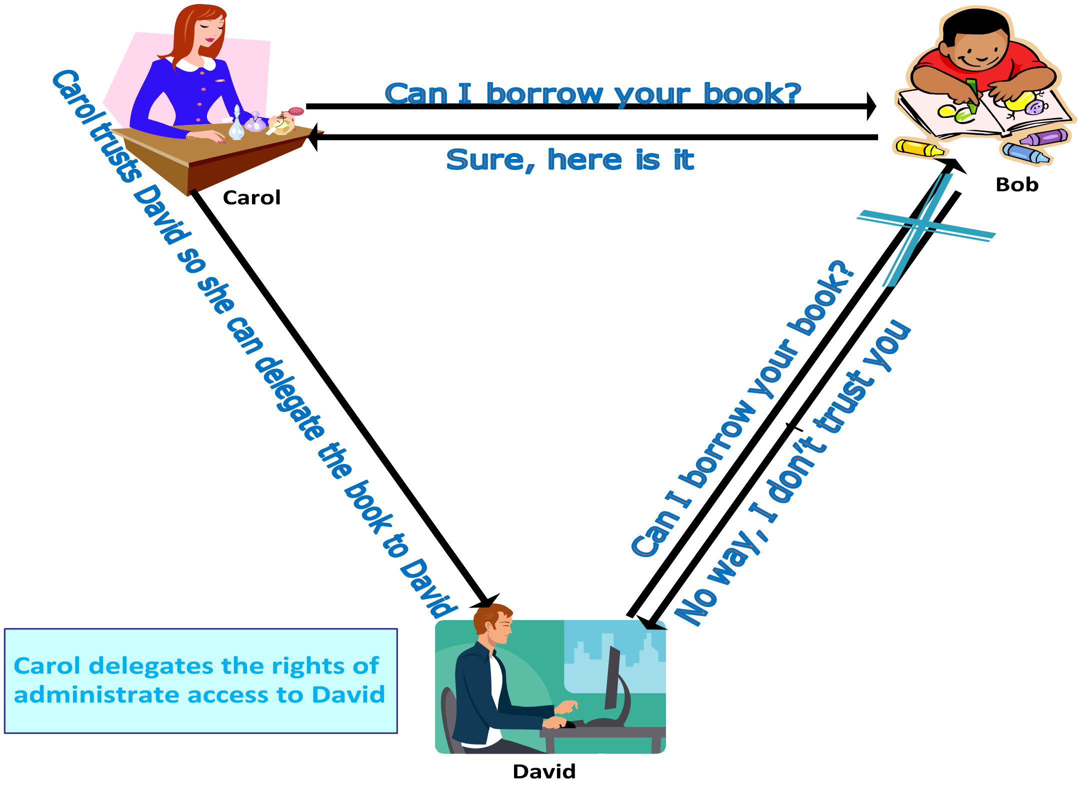

Can I borrow your book?
Sure, here is it
Carol
Bob
Carol trusts David so she can delegate the book to David
Can I borrow your book?
No way, I don't trust you
Carol delegates the rights of administrate access to David
David

Figure 3-3 Concept of Delegation using trust

3.5 Class Diagram

The class diagram of the the designed framework is shown in figure 3-4 and 3-5 comprising of:

- Access control of the university system
- Delegation module

3.5.1 Access control of the University system

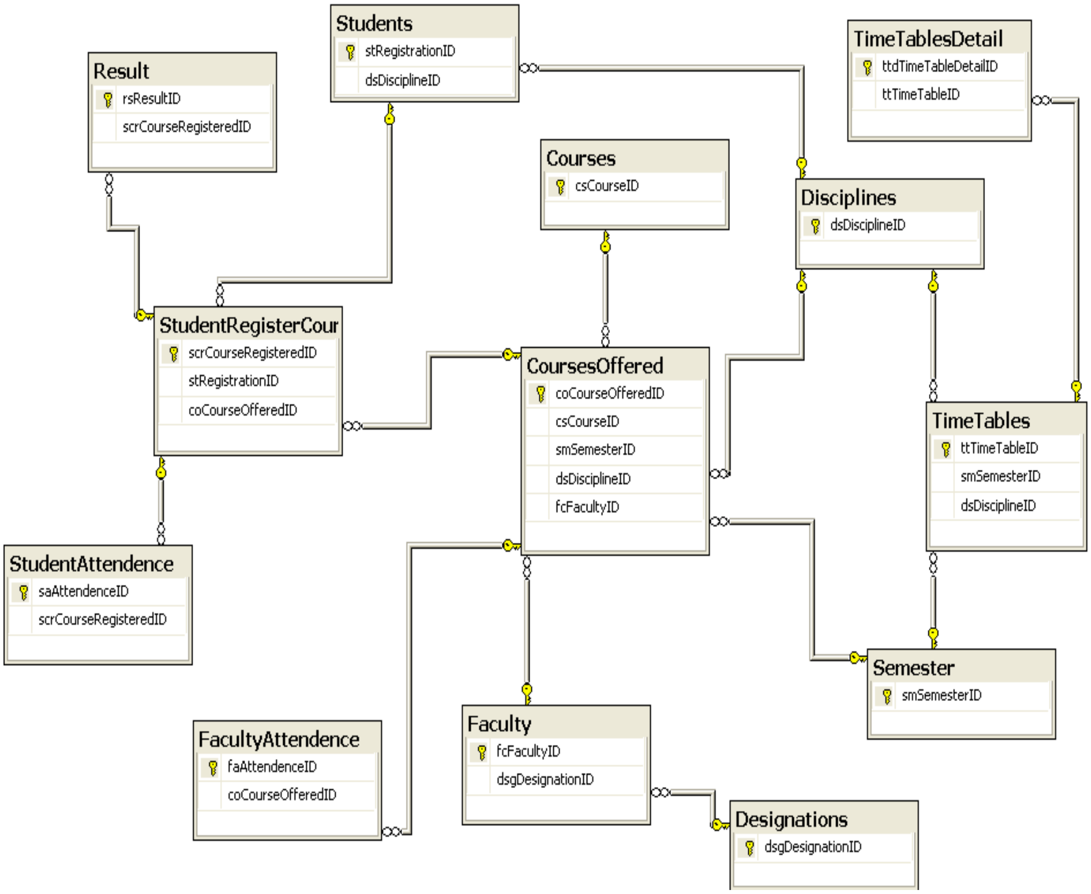


Figure 3-4 Class Diagram of the access control for the University system

3.5.2 *Delegation module*

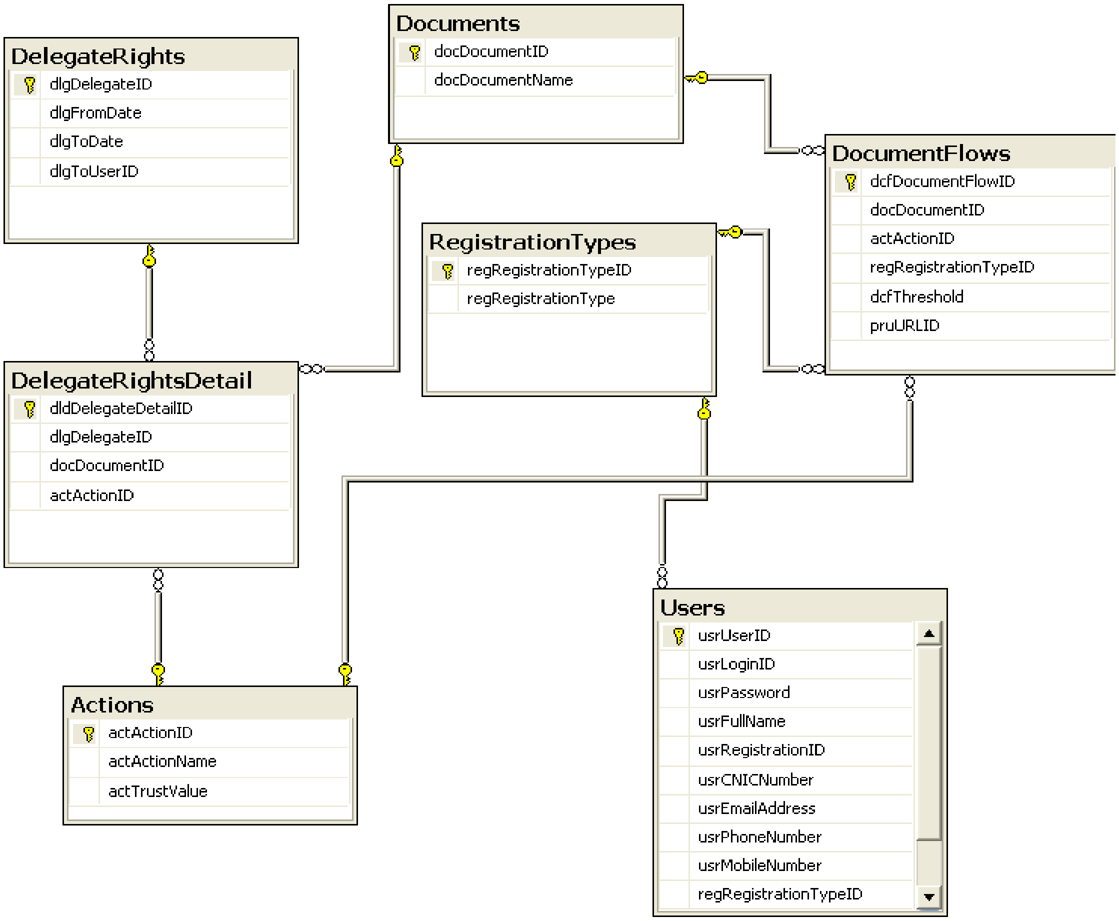


Figure 3-5 Class Diagram of Delegation Control

3.6 Trust properties and access flow control

3.6.1 *Introduction*

This section explains the flow control in Figure 3-2 and step by step proceeding of the access request. Initially all the devices in the network have been set an initial trust value. The user request is forwarded to the server by attaching the required information i.e id, request, and user_type and user_details. The query is checked against the trust record. In case no trust record is found then trust of the entity is computed by a method described below. The proposed model is drawn in figure 3-6 supports the following properties of social trust:

i. Trust is dependent on the behavior and the nature of the entity.

ii. Scales negative and positive attitude of an agent's trustworthiness, on a defined range of values

iii. Trust is based on historic interactions.

iv. Entities are able to exchange reputational information through recommendations, thus assisting in the authentication of the user.

v. Trust is based on evaluating all the recommendations taken into account by the trustworthy recommenders.

vi. Trust is subjective with reference to the different observers carrying varying opinion about the same entity's trustworthiness.

vii. Trust is dynamic and ever changing – future interactions increases or decreases the level of trust in another entity.

viii. Only Interpersonal Trust is supported [12].

3.6.2 Trust Computation Module (TCM) Flow Control

The Trust Computation Module (TCM) in Figure 3-2 interferes when no trust information is present in the trust record. It allows gathering the information from observing past interactions. Inadequate trust information by observation results in broadcasting recommendation request to only the trustworthy devices so as to avoid the incorrect information by the malicious devices. Together the recommendation value with inadequate knowledge of observation generates a trust value which goes to the TCM in Figure 3-2. A device once recognized is allocated login name and password. After requesting recommendation, the user observations are recorded with the access rights assigned. The trust calculations are taken from paper reference as [13]. Next time the authorized login is excused to perform authentication but is required to go through the procedure of update trust module and access control process in TCAM described below.

3.6.3 Trust Category Assignment Module (TCAM) Flow Control

The allocated or assigned trust value is checked in the Threshold Observant Area (TOA) after processing ends from TCM. When no trust information can be computed or is found not enough, then the entity is allowed to access or provide the service with minimum trust value in order to fulfill the criteria of pervasive computing and to avoid taking security risks on the system. The trust value is kept under the TOA which is the area between trust and mistrust in order to decide a trust category against the defined threshold value describe in table 3-1. The system awareness to user activities is increased in TOA which is able to increase or decrease the trust degree of the user. Set of trust categories are defined in table 3-1.

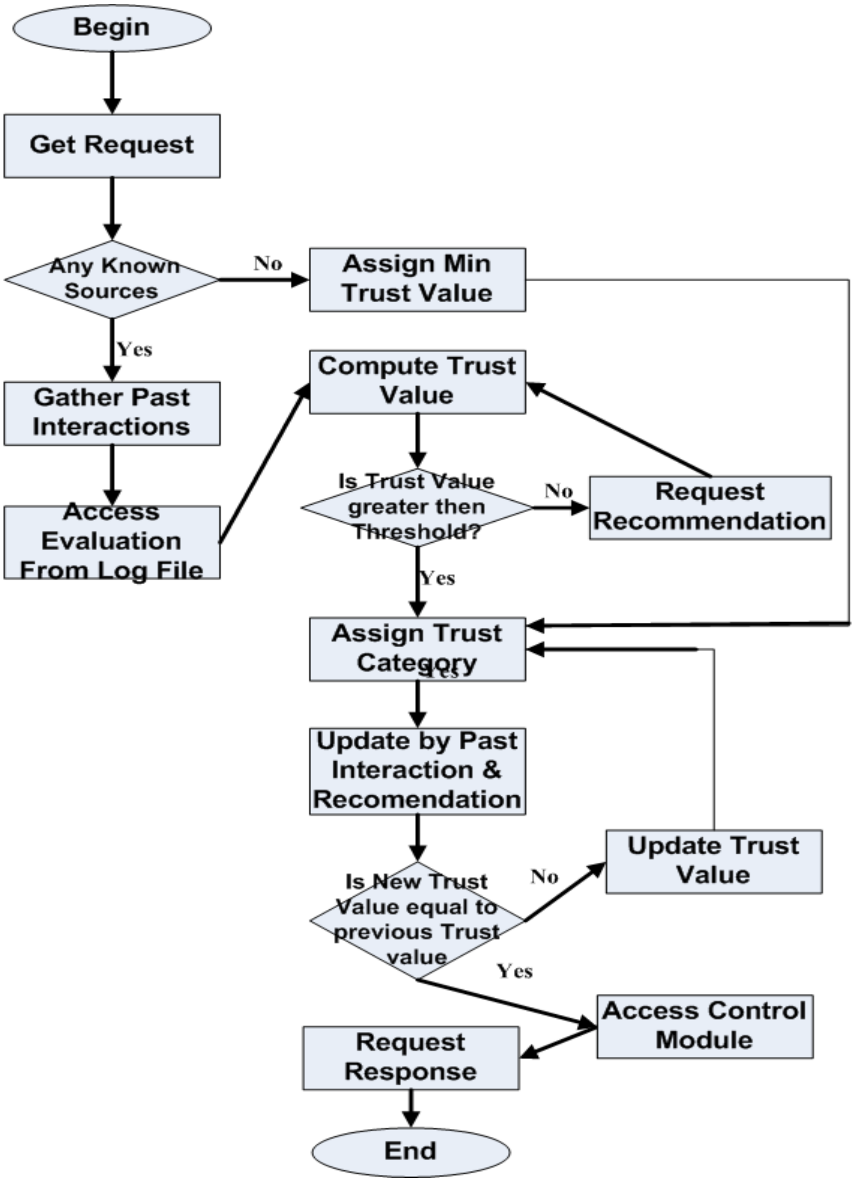
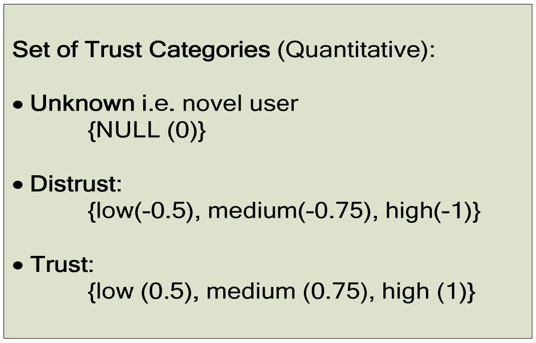


Figure 3-6 Flow chart of the trust based access control

3.6.4 Update Trust value Module

The trust values of the devices need to be monitored periodically. If, for instance any communication session has expired then prevent further interactions from that entity until user is re-authenticated. A time and date is recorded as each user logins. Trust update module checks for any new recommendation or interaction and compare the present trust value to the previous one. If there is a change in the trust value between the old and the new one update the trust category. Update trust value undergoes through TCM as labeled in Figure. 3-2 and computes the new updated values.

Table 3-1 Set of trust degrees



Set of Trust Categories (Quantitative):

- **Unknown** i.e. novel user
  {NULL (0)}

- **Distrust**:
  {low(-0.5), medium(-0.75), high(-1)}

- **Trust**:
  {low (0.5), medium (0.75), high (1)}

There are three levels of trustworthiness, first one is when the user is completely trusted and becomes a trustworthy user. Second case is when the user behaves negatively to the extent that he/she is distrusted. Third scenario arises when the user trustworthiness is uncertain, it lies in the midst of the trustworthiness and untrustworthiness and its level varies with respect to the current activities leading towards the either way of trust extremity [27].

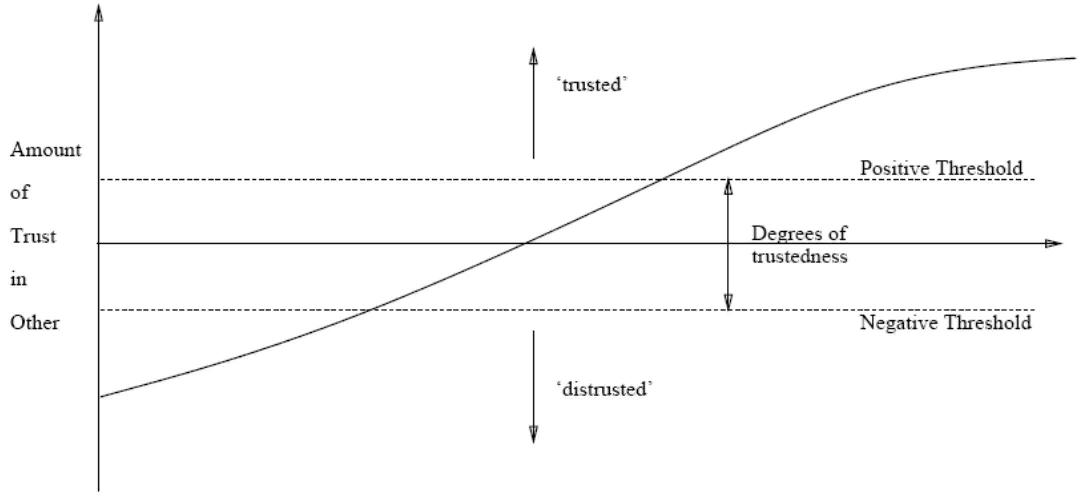


Figure 3-7 Positive and negative thresholds for trust

3.6.5 *Access Control Manager (ACM) Flow Control*

After updating the trust category, it queries ACM containing the access rules to the services. ACM calls a validation action call to check the user authorization to the service and is evaluated as valid/invalid access. A valid action goes to increase the trustworthiness of the user and an invalid action denies the action and records it in its log file decreasing the trustworthiness. It maintains a log for the user interactions performed duration this session. The trust value history is maintained which is accessible to only the trustworthy user.

3.6.6 Delegation Control Manager (DCM) Flow Control

DCM controls the delegation to a user by a trusted person for the defined period of time. The delegation details contain delegator name, delegate request, time period, trust category and delegatee_id which are maintained as delegation history. The delegator is able to delegate for a secure delegation [11]. It allows delegating all or a subset of rights to the receiving entity. The decision of the delegation can take any form from the defined set: {permit, deny, inapplicable}. Permit is to allow the receiver to utilize the delegated rights, the delegation is denied when the time is expired for the delegation and delegation is inapplicable when the delegation grant is deleted. DCM sets this status information for the user in the trust record.

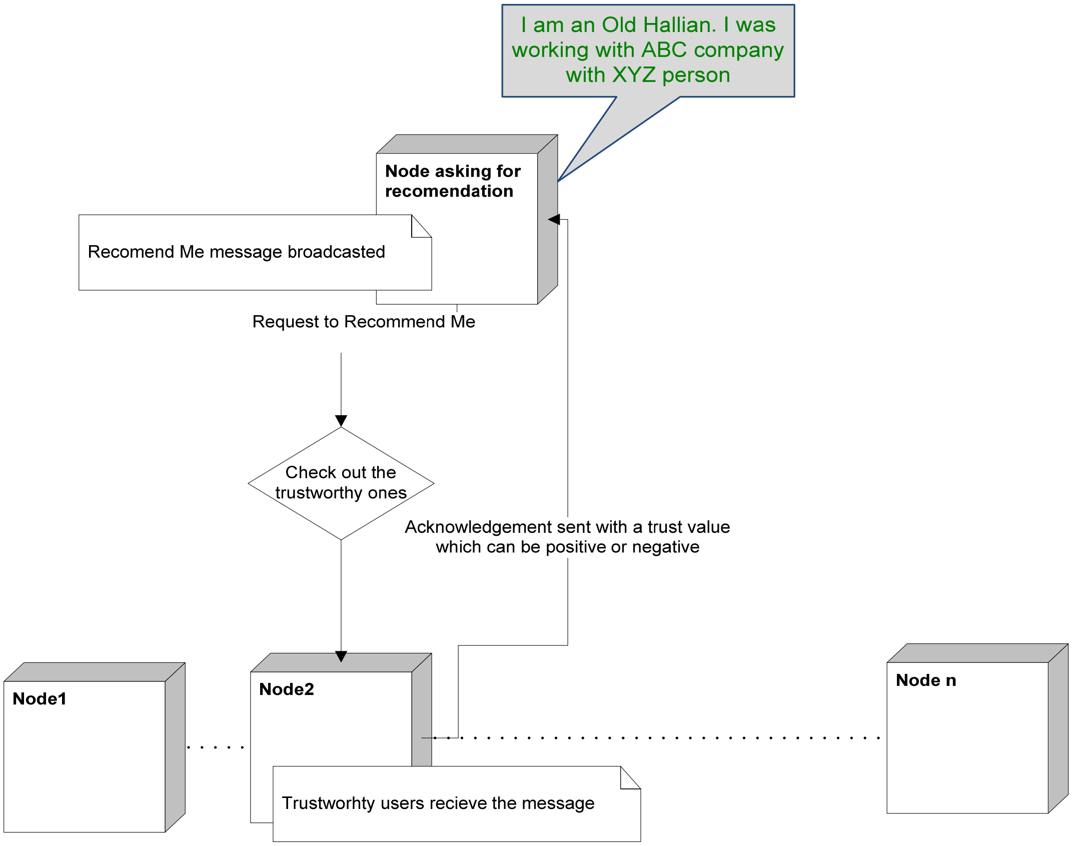

I am an Old Hallian. I was working with ABC company with XYZ person
Node asking for recomendation
Recomend Me message broadcasted
Request to Recommend Me
Check out the trustworthy ones
Acknowledgement sent with a trust value which can be positive or negative
Node1
Node2
Node n
Trustworhty users recieve the message

Figure 3-8 Recommendation Request and Response

3.7 PROPOSED ALGORITHM

The trust database contains record on the per-identity basis and each identity is assigned a trust value. Figure 8 is the designed algorithm in which the symbols and sets used are defined below

- Acc_right={save(s), delete(d),view(v)} where Acc_right is a set of action.
- Set A={(User (U1), User trust value T(U1), User observation value Obs(U1), User action Act(U1), constant_value1, constant_value2, w1 and w2 are values defined between -1 and 1 where w1 and w2 are the controlling factors, User trust level Tr_level(U1), User observations Obs(U1),User average observation value Obsr(U1), User recommendations received Rec(U1), Average recommendation value Recc(U1), old trust value T'(U1), Time period 'n', Waiting trusted user T(U2) with reference to (WRT), recommendation request Rec_request, Trust of User as Guest T(Guest)}

- As delete is the most important action and it is accessible only if the user is fully trusted. But frequent action of delete would decrease the trustworthiness of the user. So delete is a highly observant action.

```
Begin
1.     If T(U1) found in trust record then
              Goto 2;
       Else {
       (i)    Act(U1)←U1 performing Acc_right
              If (action==valid)
                     Obs(U1)←  Act(U1) + constant_value1
              Else
                     Obs(U1)←  Act(U1)- constant_value2
                     Update Obs(U1) by,
                     AVG(Obs(U1))=∑Obs(U1)/n
                     Obsr(U1)←  AVG(Obs(U1))
                     Direct trust computation ends; Goto (iii);

       (ii)   Rec(U1)← Broadcast Rec_request
              If( Rec(U1)!= Null)
              AVG(Rec(U1))=∑Rec(U1)/n
              Recc(U1)←  AVG(Rec(U1))
              Goto (iii)

       (iii)  Compute T(U1) ← ISNULL (w1*Obsr(U1)+ w2*Recc(U1))
              Direct and Indirect trust computation ends; Goto 2.

              Else If (Obsr(U1) & Recc(U1)==NULL)
              Assign min(T(U1)) and register U1;
              U1 allowed broadcasting Rec_request;
         Goto 2; }

2.     Assign the Tr_level(U1) to T(U1) by computing the level of trust & distrust [high,
       Medium(TOA i.e. Threshold observant area for unknowns), low]

              If (T(U1) > Threshold_value)
              U1 Trusted high;
              If(U1 is trusted)
              {
              U1→  gives delegation to a standby T(U2) WRT context
              If(Within Time'n')
              {
                     T(U2) can perform privileged Acc_right on the services
              }
              Else T(Guest)←  T(U2); T(U2) is categorized as guest
              }
                     Tr_level(U1) assigned;

              ElseIf (T(U1) == Threshold_value)
                     U1 Unknown;
                     Tr_level(U1) assigned;

              ElseIf (T(U1) ←Threshold_value)
                     U1 Distrusted;
                     Tr_level(U1) assigned;
```

```
3.      Monitoring the behavior of U1
        Check access for the U1 on Acc_right;
        ⇨(T (U1)) by 1 of (i) and (ii))
        if(T(U1) != T’(U1)))then
                Update T(U1) by 1 of (iii);
End of Algorithm;
```

Figure 3-9 Algorithm designed for the access control and delegation of services

Chapter 4

FRAMEWORK DEVELOPMENT OF DISTIBUTED TRUST BASED ACCESS CONTROL IN ASSUMED SCENARIO OF PERVASIVE COMPUTING

4.1 Introduction

In this chapter the idea of the proposed distributed trust based access control (DTBAC) is developed into a web based framework to show the flow control of the interactions in an assumed scenario of the pervasive computing. The framework is a generic skeleton which can be modified into an application. The delegation control module is embedded along with assigning the privileges of the documents to the users, arrange in some defined category. The services or documents are the terms used interchangeably. DTBAC is a computational framework presenting the ad hoc solution which can be systematically generalized, rationalized and extended. DTBAC uses a concept derived from similar work problem, and showed that it expresses the problem in a more generic way. Figure 4-1 shows the overview of the approach. The three principles guide the design of our system architecture and make it feasible for application developers to program for change, resulting in applications which are adaptable. However, to make the vision to become a reality, developers must build applications that autonomously adapt to a highly dynamic computing environment [26].

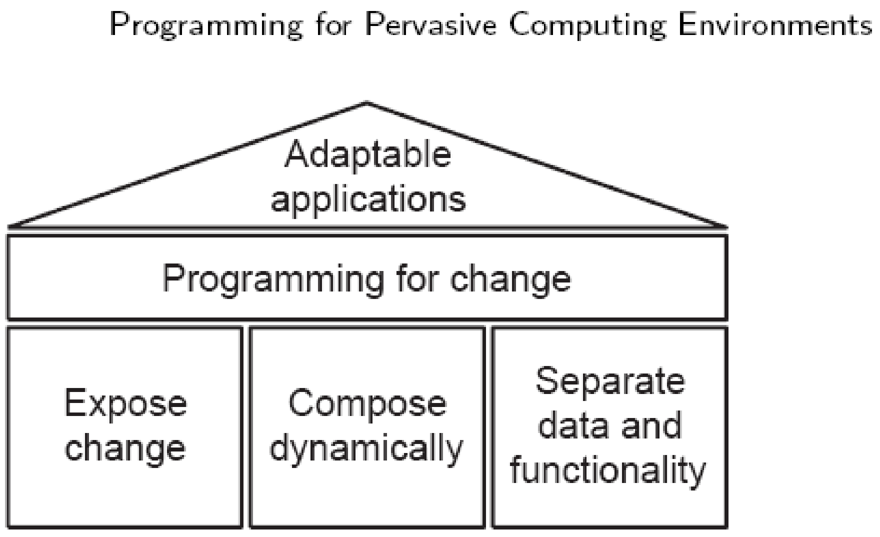

Programming for Pervasive Computing Environments
Adaptable applications
Programming for change
Expose change
Compose dynamically
Separate data and functionality

Figure 0 Overview of the Development

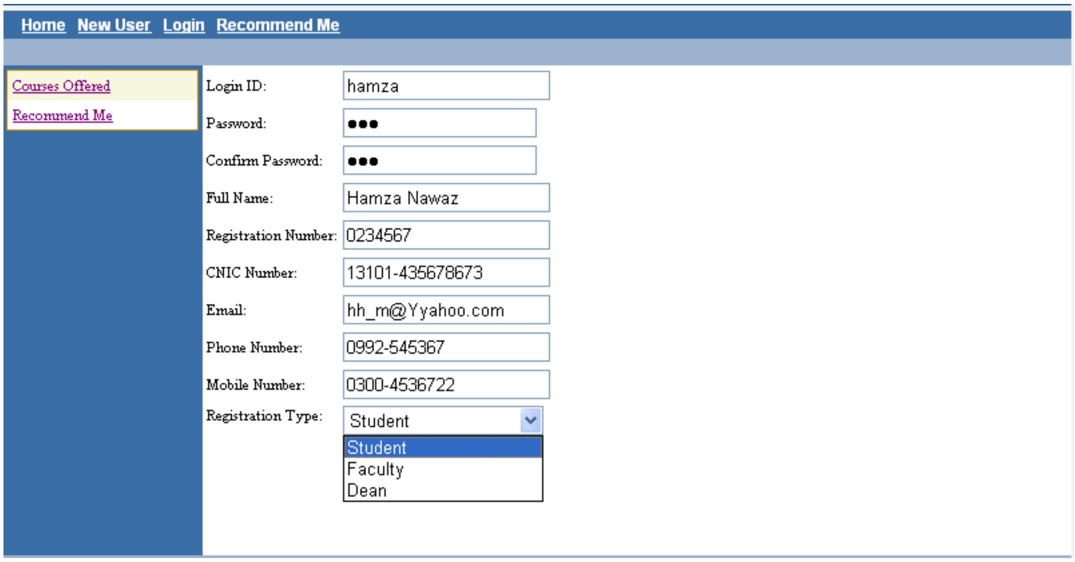
4.2 System Specifications

The purpose of this access control as describe before is to allow the ad-hoc user to enter in the system and then according to the behavior of the user, set the level of its trust. The framework allows the ad hoc user to access the service by pinning down his/her personal information and credentials in a form together with specifying the type of user. The request is processed and the user is allocated a login name and password.

Each user is allowed to move in its own category. The privileges gain or loss to the documents varies depending on the level of trust. For example initially if a new faculty is created then is allowed to view the courses but is not allowed to add or delete a new course or the courses offered. After logging successfully in the system if a user does not perform any activity for a time frame specified then the account session becomes inactive and the user has to login again.

4.2.1 Ad hoc User

A screen shot of an ad-hoc user registration is given in figure 4-2. The category of the user describes the home screen page. For each category home screen appears different.



Login ID: hamza

Password: ●●●

Confirm Password: ●●●

Full Name: Hamza Nawaz

Registration Number: 0234567

CNIC Number: 13101-435678673

Email: hh_m@Yyahoo.com

Phone Number: 0992-545367

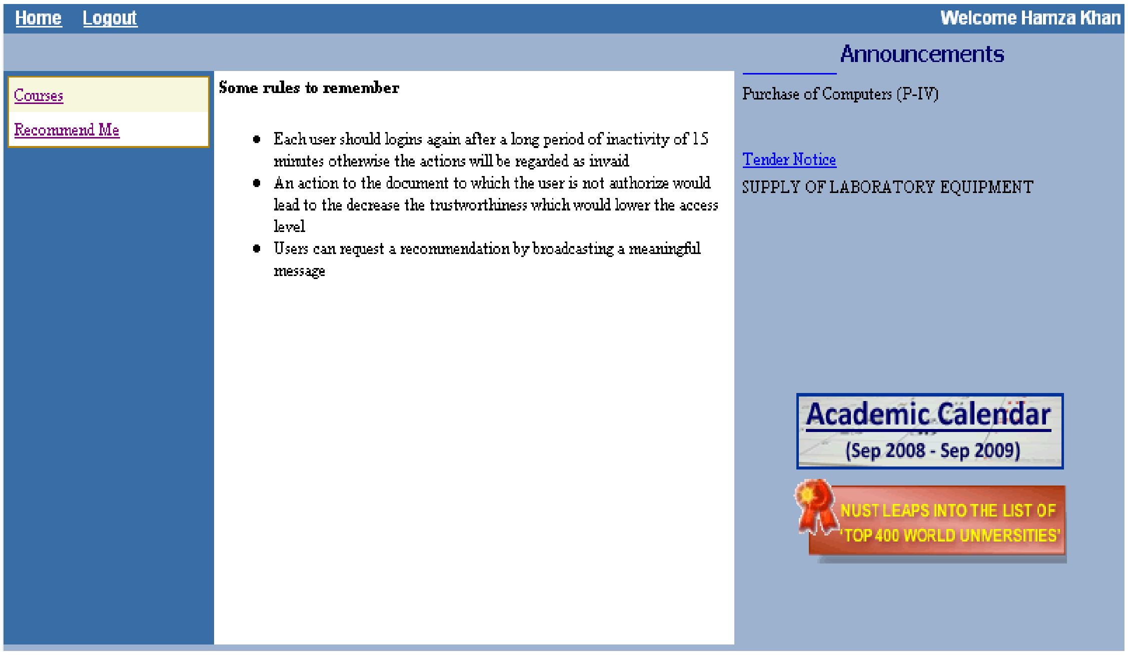
Mobile Number: 0300-4536722

Registration Type: Student

Student
Faculty
Dean

Figure 4-2 Ad-Hoc User registration

The ad-hoc user of type student Hamza Nawaz has the following home screen as given in figure 4-3 after successfully making the registration.



Some rules to remember

- Each user should logins again after a long period of inactivity of 15 minutes otherwise the actions will be regarded as invaid
- An action to the document to which the user is not authorize would lead to the decrease the trustworthiness which would lower the access level
- Users can request a recommendation by broadcasting a meaningful message

Purchase of Computers (P-IV)

Tender Notice

SUPPLY OF LABORATORY EQUIPMENT

Figure 4-3 The User Type as Student

The new user has the access to the menu as shown in figure 4-3 initially after signing in to the allocated account. The basic menu is visible on the category basis but to have more rights and increase level of access then the user has to request for recommendation. The procedure defined for creating login is called from the database server as given below

```
PROCEDURE AuthenticateUser()
(
            @usrLoginID       VarChar
      ,     @usrPassword      VarChar
)
BEGIN
      SELECT            usrUserID
                  ,     regRegistrationTypeID
                  ,     CurrentTrustValue
            FROM  Users
WHERE usrLoginID        =     @usrLoginID AND
usrPassword =@usrPassword AS varbinary(8000))
END
```

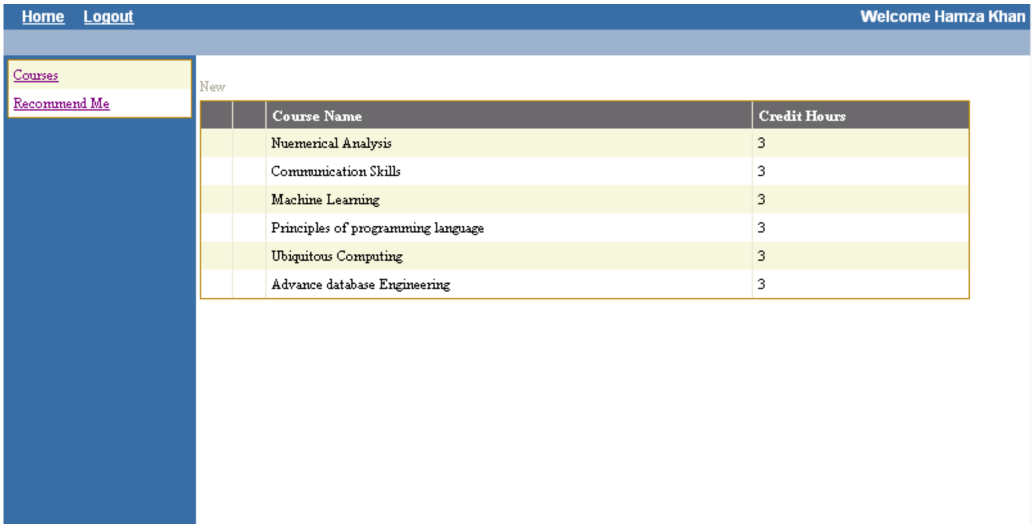
Class for the authentcation of user

```
public class clsAuthenticate : DBAccess
    {
        #region Members
        #region Properties
      public bool AuthenticateUser()
        {
SPCall("[SPI_CalculateTrustValue]",   "@usrLoginID",   m_usrLoginID,
"@usrPassword", m_usrPassword);

System.Data.DataTable    dt    =    SPCall("[SPI_AuthenticateUser]",
"@usrLoginID", m_usrLoginID, "@usrPassword", m_usrPassword);

if (dt != null && dt.Rows.Count > 0)
            {
m_usrUserID=         Int64.Parse(dt.Rows[0]["usrUserID"].ToString());
m_regRegistrationTypeID=Int64.Parse(dt.Rows[0]["regRegistrationTypeID
"].ToString());
m_CurrentTrustValue=double.Parse(dt.Rows[0]["CurrentTrustValue"].ToSt
ring());
AuditUserID    =  Int64.Parse(dt.Rows[0]["usrUserID"].ToString());
                AutidDetail    =  "Login: User authenticated";
                SaveAudit();
                return true;
            }
            return false;
            }
```

On positive recommendation these users can become more trustworthy and have the increase level of access to the documents in the menu. It can be best explained by figure 4-4 as by default the user can only view the courses but the privilege to add, delete and search are unavailable as the new option is disabled. The details of the recommendations are solicited in section 4.3



		Course Name	Credit Hours
		Nuemerical Analysis	3
		Communication Skills	3
		Machine Learning	3
		Principles of programming language	3
		Ubiquitous Computing	3
		Advance database Engineering	3

Figure 4-4 Unable to do the changes and can only view the courses

The user student which is new can broadcast a message which would help him in its recognition as represented in figure 4-5.

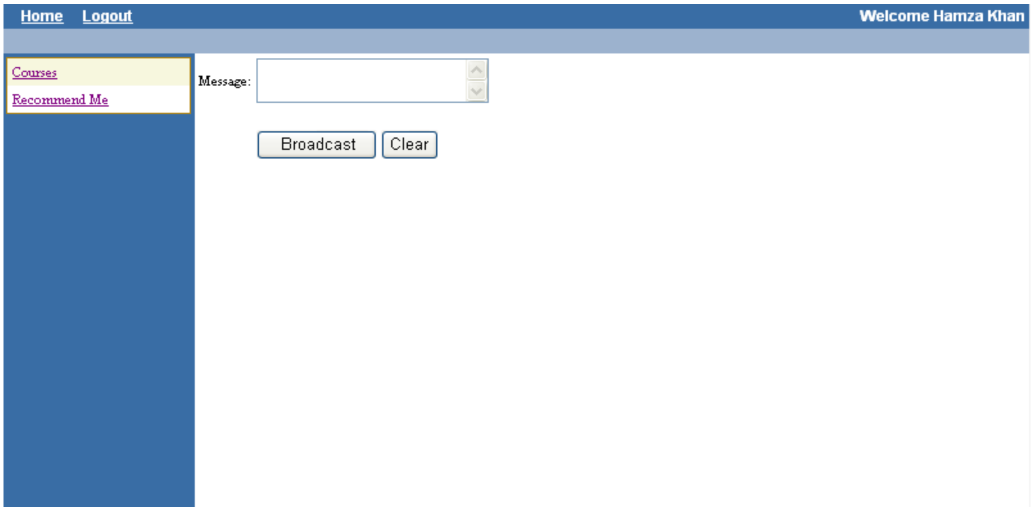
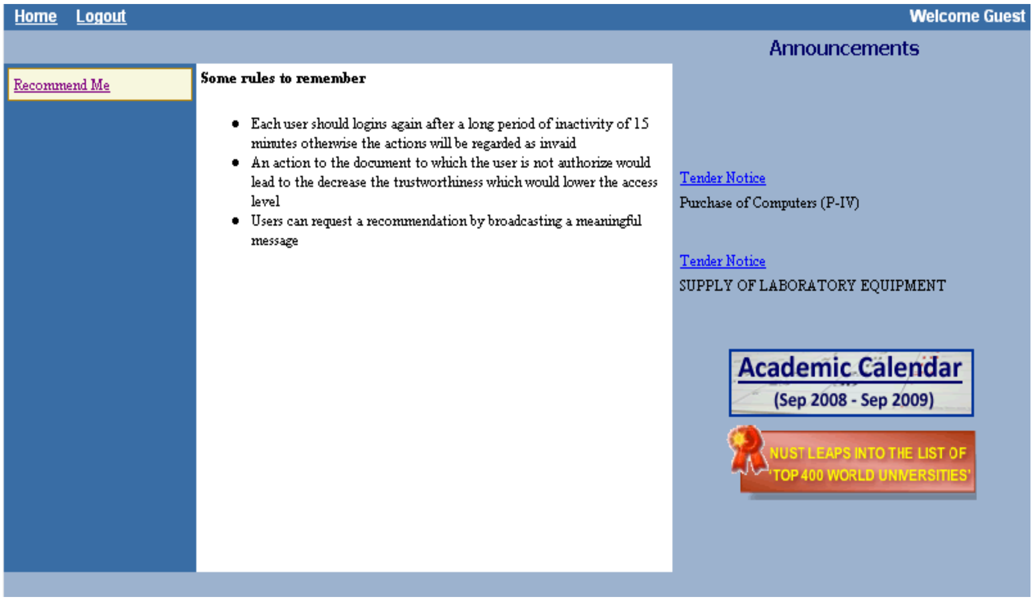


Figure 4-5 A useful and trustworthy message broadcasted for recommendation

The user registered in the system with the type guest has the main screen shown in figure 4-6.



Some rules to remember

- Each user should logins again after a long period of inactivity of 15 minutes otherwise the actions will be regarded as invaid
- An action to the document to which the user is not authorize would lead to the decrease the trustworthiness which would lower the access level
- Users can request a recommendation by broadcasting a meaningful message

Tender Notice
Purchase of Computers (P-IV)

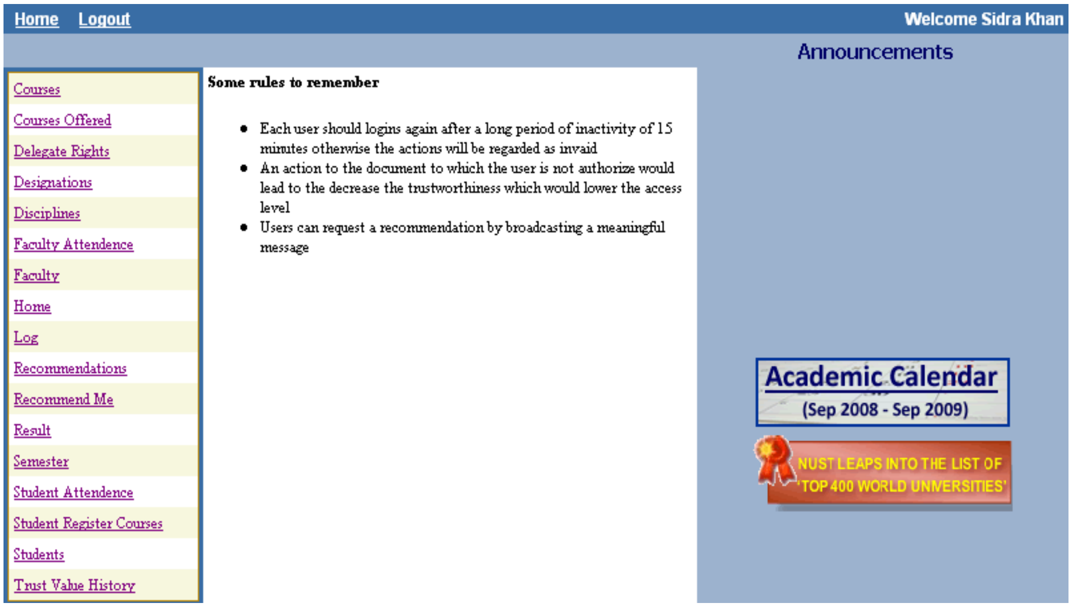
Tender Notice
SUPPLY OF LABORATORY EQUIPMENT

Figure 4-6 A User of type guest

A guest is able to only ask for recommendation. The guest is not able to access any service. A guest is a person who is not in direct interaction with the environment and is an outsider for example a guest wants to view the courses or the courses offered in the specific discipline of the university.

4.2.2 A Trusted User i.e. Administrator

The trustworthy person is the person who has the full access rights and has the highest trust value so that the menu containing all the documents or services are made appeared to him. The user in this case is administrator. The screenshot of the trusted user is shown in figure 4-7:



Some rules to remember

- Each user should logins again after a long period of inactivity of 15 minutes otherwise the actions will be regarded as invaid
- An action to the document to which the user is not authorize would lead to the decrease the trustworthiness which would lower the access level
- Users can request a recommendation by broadcasting a meaningful message

Figure 4-7 The Trustworthy User Administrator Home Page

The list of all the documents is shown to the administrator. These documents are shown to the user on the basis of high trust value which is possessed by the administrator.

The admin can give:

1. Recommendations:

Recommendation is a concept which occurs for the user whose trust record is unavailable. Trust requires an authentication from the trustworthy user. The request is broadcasted to the list of the trustworthy users. Initially the only well reputed user is the admin and other users on the basis of their actions and behavior with the passage of time will built their good-will and can become a trusted user. So the recipient trustworthy user will receive the broadcasted request and will respond with the discreet numeric trust value which will be added to its previous trust value after being multiplied with a weighing factor.

2. Delegate rights

Delegation has the concept of keeping oneself's rights and also allows permitting subset of these rights or all rights to the delegetee. But delegation exists to the persons whose credentials are known and are assigned for the limited period of time.

3. Access to the history of actions

The series of actions performed by the users are maintained in the log. History is accessible to only the trusted user. The actions along with their details like action description, date time and delegatee's name are logged and are only accessible to administrator.

4.3 Trust Value Calculation

4.3.1 Introduction

A user who is under observation on novel basis has the option of becoming a trustworthy user by broadcasting a recommendation message in which any useful meaning can be conveyed to a list of recommenders. A trust value calculation occurs on these recommended values and from interactions which the user performs in its domain as defined in the procedure of CalculateTrustvalue procedure below. These values are updated as any new recommendation or observation occurs. Every action is authorized as a valid or invalid action. Valid is when it remained successful and invalid is when action failed to performed. The details of both recommendation and observation are explained in the subsections of this module.

```
PROCEDURE CalculateTrustValue
(
                @usrLoginID  VarChar
                @usrPasswordVarChar
)
BEGIN
        DECLARE @UserID as bigint
        DECLARE @W1       as Float
        DECLARE @W2       as Float

        SELECT        @UserID = [usrUserID]
                FROM [Users]
                WHERE        usrLoginID= @usrLoginID
                                AND    usrPassword=(@usrPassword
        SELECT         @W1=0.8
                      ,@W2=0.2
        DECLARE @ObservationTrustValue as Float
        DECLARE @RecommendedTrustValue as Float

        SELECT   @ObservationTrustValue =AVG([uatTrustValue])
                FROM [UserAuditTrail]
        SELECT
        @RecommendedTrustValue=AVG([rcdRecommendTrustValue])
                FROM [RecommendedUsers]
```

Same goes for observation calculations:
Average of observations = AVG(Obs),sum of all the observation values is taken as ∑Obs(U1), number of recommendations is taken as (n):

$$AVG(Obs(U1)) = \frac{\sum Obs(U1)}{n}$$

and
The mathematical formula used for recommendation value is:
Sum of all the recommendation values are taken as ∑Rec(U1), number of recommendations is taken as (n):

$$AVG(Rec(U1)) = \frac{\sum Rec(U1)}{n}$$

As there are more than one recommendation, so the average of all the recommended trust values are taken and same is the case with observation value as the average of all the observation trust value are taken.
Weights (W) are multiplying or the controlling factor. As weight1 (W1) is assigned to observation and weight2 (W2) is assigned to recommendation. W2 is set to a lower value than W1 in order to give more importance and emphasis to the decision of the observation than recommendation. Vise versa the values of the weights can also be changed and are kept dynamic so that according to the context of the application the weights can be adjusted.
The weights are set to nuemeric value as **W1=0.8 and W2 =0.2.**
A single trusted value for the user is calculated by the following formula which is set as the user current trust value after performing the updates.

User_Current_Trust_Value= W1*Avg(Obs) + W2*Avg(Rec)

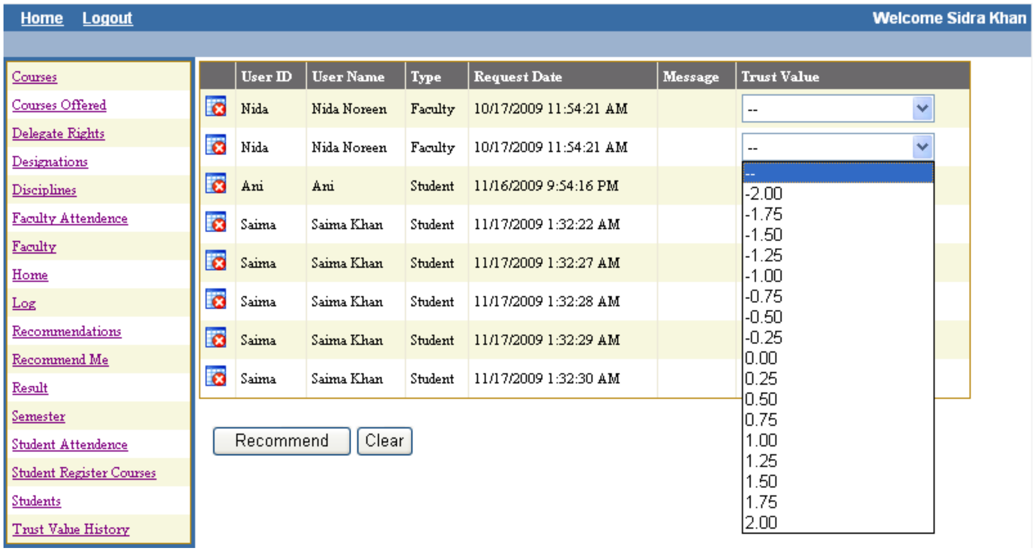
This current trust value is mapped to the user trust value field. So in this way this updation occurs everytime if the value of the trust value changes by the source of observation and recommendation on the next login of the user. The actions made by the user are accountable for the user authorizations in the next sequences of login. The description of the submodules which include recommendation and observation is described below:

4.3.2 Recommendations

The recommendation is based on the perception of a human being about a person based on the communication, collaboration and interactions. For instance, a recommendation can be based on a log history containing all the activities which describe the Trustee experience [24].Recommendation is a vital concept in the trust based access control as it assists the user in his/her easy access to the documents. Recommend Me is a service which is in access of every user. It facilitates the request message to be broadcasted on the channel in a meaningful way.

The message broadcasted is received by a list of users along with the details of recommendation whose trust level is equal to or greater than threshold defined as for each action like shown in the figure 4-8. The procedure of user checking against a defined threshold value is called and validated for the user rights which are given in the module authorization.

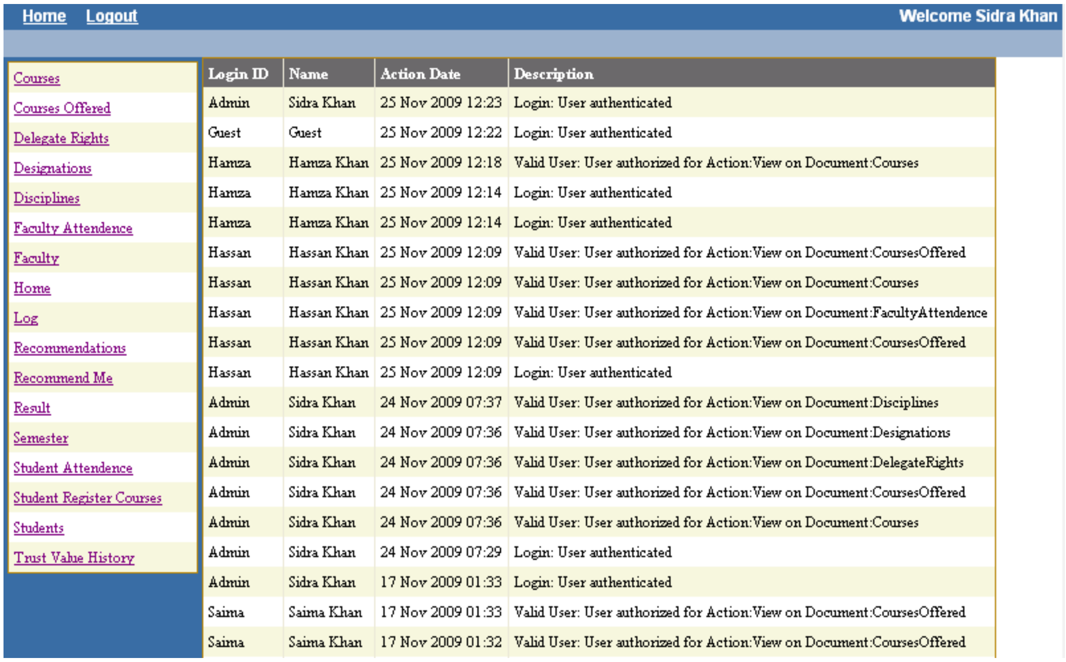
The recommenders recommend a trust value depending on the context of the person in the field by clicking the recommend button. The details of the request can be viewed by the time and date, user id, user type and most important the request message as illustrated in figure 4-8. The details of these users can also be seen form the log. Dean is giving recommendation and can also ask for recommendation. The admin can give recommendations. Besides this, administrator can also delegate the right of recommendations. As admin is also broadcasting message to Recommend Me then dean is the person who can receive this message.



	User ID	User Name	Type	Request Date	Message	Trust Value
☒	Nida	Nida Noreen	Faculty	10/17/2009 11:54:21 AM		--
☒	Nida	Nida Noreen	Faculty	10/17/2009 11:54:21 AM		--
☒	Ani	Ani	Student	11/16/2009 9:54:16 PM		
☒	Saima	Saima Khan	Student	11/17/2009 1:32:22 AM		
☒	Saima	Saima Khan	Student	11/17/2009 1:32:27 AM		
☒	Saima	Saima Khan	Student	11/17/2009 1:32:28 AM		
☒	Saima	Saima Khan	Student	11/17/2009 1:32:29 AM		
☒	Saima	Saima Khan	Student	11/17/2009 1:32:30 AM		

Figure 4-8 List of users recommendation request been received by a highly trustworthy person

The log can be seen from the figure 4-9. The actions done by the user can be observed from the log file.



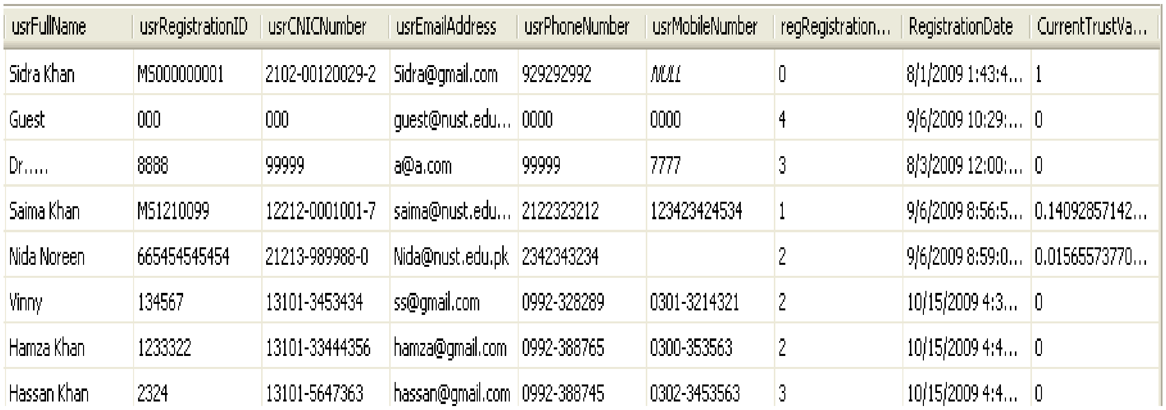
Login ID	Name	Action Date	Description
Admin	Sidra Khan	25 Nov 2009 12:23	Login: User authenticated
Guest	Guest	25 Nov 2009 12:22	Login: User authenticated
Hamza	Hamza Khan	25 Nov 2009 12:18	Valid User: User authorized for Action:View on Document:Courses
Hamza	Hamza Khan	25 Nov 2009 12:14	Login: User authenticated
Hamza	Hamza Khan	25 Nov 2009 12:14	Login: User authenticated
Hassan	Hassan Khan	25 Nov 2009 12:09	Valid User: User authorized for Action:View on Document:CoursesOffered
Hassan	Hassan Khan	25 Nov 2009 12:09	Valid User: User authorized for Action:View on Document:Courses
Hassan	Hassan Khan	25 Nov 2009 12:09	Valid User: User authorized for Action:View on Document:FacultyAttendence
Hassan	Hassan Khan	25 Nov 2009 12:09	Valid User: User authorized for Action:View on Document:CoursesOffered
Hassan	Hassan Khan	25 Nov 2009 12:09	Login: User authenticated
Admin	Sidra Khan	24 Nov 2009 07:37	Valid User: User authorized for Action:View on Document:Disciplines
Admin	Sidra Khan	24 Nov 2009 07:36	Valid User: User authorized for Action:View on Document:Designations
Admin	Sidra Khan	24 Nov 2009 07:36	Valid User: User authorized for Action:View on Document:DelegateRights
Admin	Sidra Khan	24 Nov 2009 07:36	Valid User: User authorized for Action:View on Document:CoursesOffered
Admin	Sidra Khan	24 Nov 2009 07:36	Valid User: User authorized for Action:View on Document:Courses
Admin	Sidra Khan	24 Nov 2009 07:29	Login: User authenticated
Admin	Sidra Khan	17 Nov 2009 01:33	Login: User authenticated
Saima	Saima Khan	17 Nov 2009 01:33	Valid User: User authorized for Action:View on Document:CoursesOffered
Saima	Saima Khan	17 Nov 2009 01:32	Valid User: User authorized for Action:View on Document:CoursesOffered

Figure 4-9 The log containing details of the actions performed by the user

Point is that user's bad actions are kept under observation. A bad action leads to the decrease of the user trust value ultimately leading to decreased access to services where as a positive recommendation increases the trust value.

4.3.3 Observations

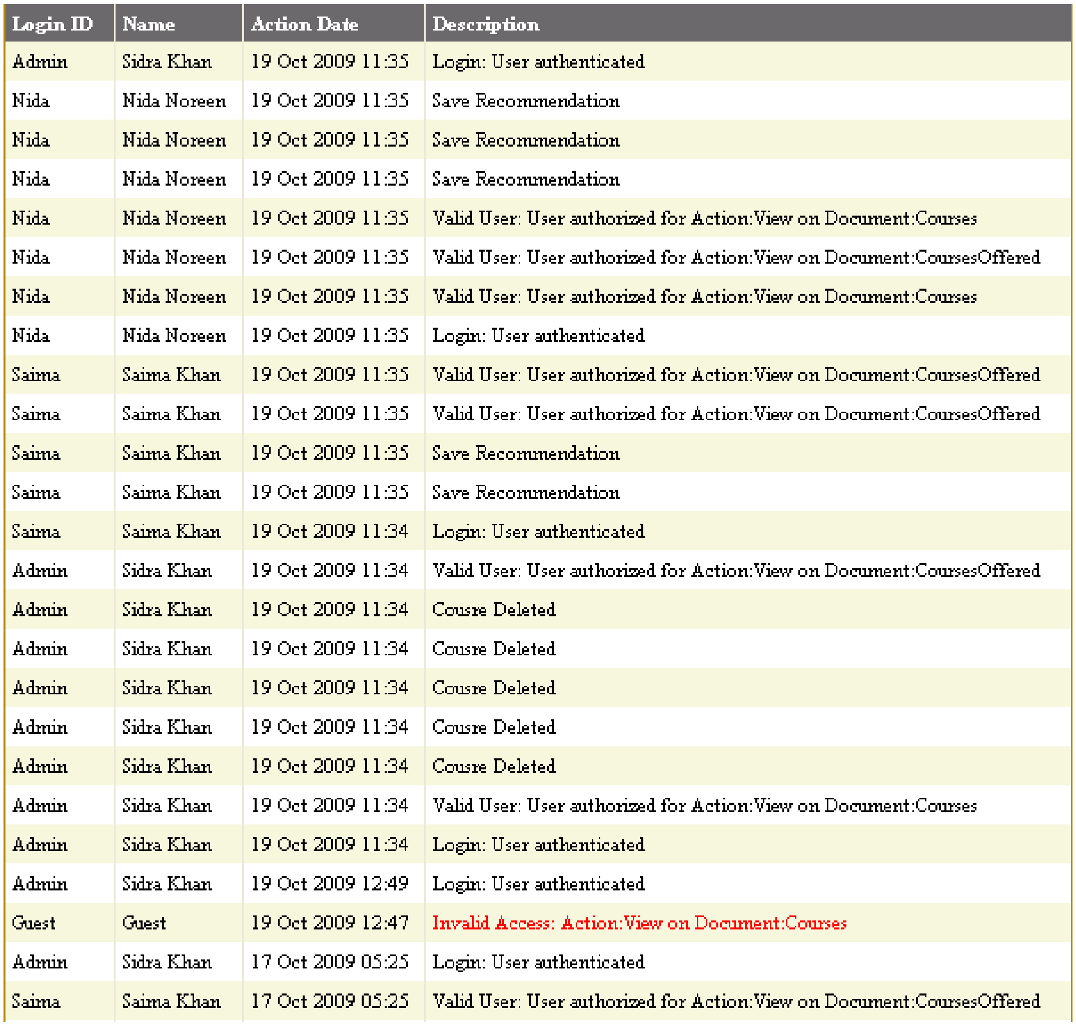
The screen shot of the user trust values before interactions and recommendations are shown in figure 4-10.



usrFullName	usrRegistrationID	usrCNICNumber	usrEmailAddress	usrPhoneNumber	usrMobileNumber	regRegistration...	RegistrationDate	CurrentTrustVa...
Sidra Khan	MS000000001	2102-00120029-2	Sidra@gmail.com	929292992	*NULL*	0	8/1/2009 1:43:4...	1
Guest	000	000	guest@nust.edu...	0000	0000	4	9/6/2009 10:29:...	0
Dr.....	8888	99999	a@a.com	99999	7777	3	8/3/2009 12:00:...	0
Saima Khan	MS1210099	12212-0001001-7	saima@nust.edu...	2122323212	123423424534	1	9/6/2009 8:56:5...	0.14092857142...
Nida Noreen	665454545454	21213-989988-0	Nida@nust.edu.pk	2342343234		2	9/6/2009 8:59:0...	0.01565573770...
Vinny	134567	13101-3453434	ss@gmail.com	0992-328289	0301-3214321	2	10/15/2009 4:3...	0
Hamza Khan	1233322	13101-33444356	hamza@gmail.com	0992-388765	0300-353563	2	10/15/2009 4:4...	0
Hassan Khan	2324	13101-5647363	hassan@gmail.com	0992-388745	0302-3453563	3	10/15/2009 4:4...	0

Figure 4-10 The Trust Values of Users after Observations and Recommendations

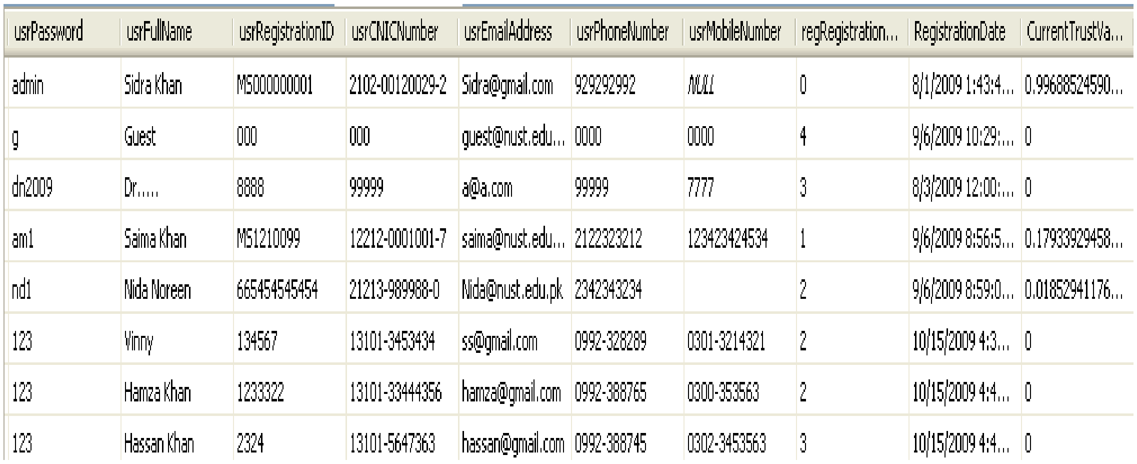
The activities performed by the users during their session time are shown in figure 4-11.



Login ID	Name	Action Date	Description
Admin	Sidra Khan	19 Oct 2009 11:35	Login: User authenticated
Nida	Nida Noreen	19 Oct 2009 11:35	Save Recommendation
Nida	Nida Noreen	19 Oct 2009 11:35	Save Recommendation
Nida	Nida Noreen	19 Oct 2009 11:35	Save Recommendation
Nida	Nida Noreen	19 Oct 2009 11:35	Valid User: User authorized for Action:View on Document:Courses
Nida	Nida Noreen	19 Oct 2009 11:35	Valid User: User authorized for Action:View on Document:CoursesOffered
Nida	Nida Noreen	19 Oct 2009 11:35	Valid User: User authorized for Action:View on Document:Courses
Nida	Nida Noreen	19 Oct 2009 11:35	Login: User authenticated
Saima	Saima Khan	19 Oct 2009 11:35	Valid User: User authorized for Action:View on Document:CoursesOffered
Saima	Saima Khan	19 Oct 2009 11:35	Valid User: User authorized for Action:View on Document:CoursesOffered
Saima	Saima Khan	19 Oct 2009 11:35	Save Recommendation
Saima	Saima Khan	19 Oct 2009 11:35	Save Recommendation
Saima	Saima Khan	19 Oct 2009 11:34	Login: User authenticated
Admin	Sidra Khan	19 Oct 2009 11:34	Valid User: User authorized for Action:View on Document:CoursesOffered
Admin	Sidra Khan	19 Oct 2009 11:34	Cousre Deleted
Admin	Sidra Khan	19 Oct 2009 11:34	Cousre Deleted
Admin	Sidra Khan	19 Oct 2009 11:34	Cousre Deleted
Admin	Sidra Khan	19 Oct 2009 11:34	Cousre Deleted
Admin	Sidra Khan	19 Oct 2009 11:34	Cousre Deleted
Admin	Sidra Khan	19 Oct 2009 11:34	Valid User: User authorized for Action:View on Document:Courses
Admin	Sidra Khan	19 Oct 2009 11:34	Login: User authenticated
Admin	Sidra Khan	19 Oct 2009 12:49	Login: User authenticated
Guest	Guest	19 Oct 2009 12:47	Invalid Access: Action:View on Document:Courses
Admin	Sidra Khan	17 Oct 2009 05:25	Login: User authenticated
Saima	Saima Khan	17 Oct 2009 05:25	Valid User: User authorized for Action:View on Document:CoursesOffered

Figure 4-71 Log Containing all the Actions performed by every user who joins the sytem

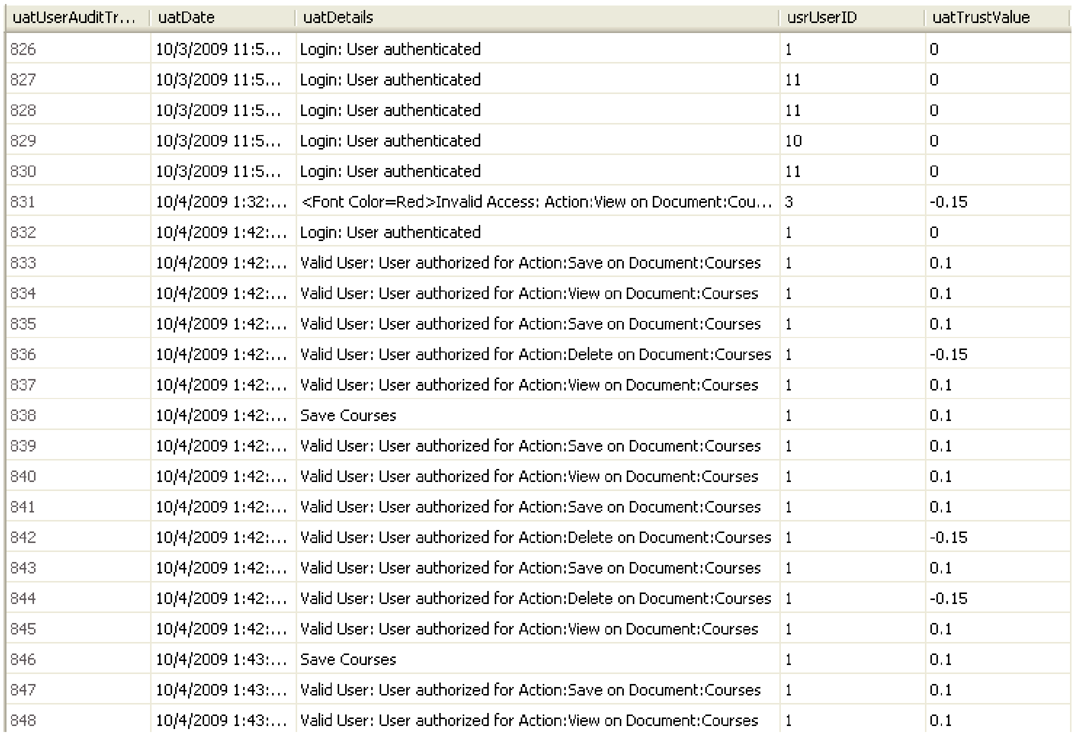
After their activities the trust values of the users changes. As its clear from the log the admin performed the delete operation thrice its trust value decreases. Figure 4-11 is a log file maintained for the framework containing action of the user with its details. The guest performed an invalid action on the specified date and time and is highlighted in red colour. The dbo.user table in figure 4-12 shows the change in trust value next time when user logins.



usrPassword	usrFullName	usrRegistrationID	usrCNICNumber	usrEmailAddress	usrPhoneNumber	usrMobileNumber	regRegistration...	RegistrationDate	CurrentTrustVa...
admin	Sidra Khan	MS000000001	2102-00120029-2	Sidra@gmail.com	929292992	*NULL*	0	8/1/2009 1:43:4...	0.99688524590...
g	Guest	000	000	guest@nust.edu...	0000	0000	4	9/6/2009 10:29:...	0
dn2009	Dr.....	8888	99999	a@a.com	99999	7777	3	8/3/2009 12:00:...	0
am1	Saima Khan	MS1210099	12212-0001001-7	saima@nust.edu...	2122323212	123423424534	1	9/6/2009 8:56:5...	0.17933929458...
nd1	Nida Noreen	665454545454	21213-989988-0	Nida@nust.edu.pk	2342343234		2	9/6/2009 8:59:0...	0.01852941176...
123	Vinny	134567	13101-3453434	ss@gmail.com	0992-328289	0301-3214321	2	10/15/2009 4:3...	0
123	Hamza Khan	1233322	13101-33444356	hamza@gmail.com	0992-388765	0300-353563	2	10/15/2009 4:4...	0
123	Hassan Khan	2324	13101-5647363	hassan@gmail.com	0992-388745	0302-3453563	3	10/15/2009 4:4...	0

Figure 4-12 Trust Values of User after Interactions

At the backend of the application the action values are retrieved and applied for each interaction of the user in figure 4-13.



uatUserAuditTr...	uatDate	uatDetails	usrUserID	uatTrustValue
826	10/3/2009 11:5...	Login: User authenticated	1	0
827	10/3/2009 11:5...	Login: User authenticated	11	0
828	10/3/2009 11:5...	Login: User authenticated	11	0
829	10/3/2009 11:5...	Login: User authenticated	10	0
830	10/3/2009 11:5...	Login: User authenticated	11	0
831	10/4/2009 1:32:...	<Font Color=Red>Invalid Access: Action:View on Document:Cou...	3	-0.15
832	10/4/2009 1:42:...	Login: User authenticated	1	0
833	10/4/2009 1:42:...	Valid User: User authorized for Action:Save on Document:Courses	1	0.1
834	10/4/2009 1:42:...	Valid User: User authorized for Action:View on Document:Courses	1	0.1
835	10/4/2009 1:42:...	Valid User: User authorized for Action:Save on Document:Courses	1	0.1
836	10/4/2009 1:42:...	Valid User: User authorized for Action:Delete on Document:Courses	1	-0.15
837	10/4/2009 1:42:...	Valid User: User authorized for Action:View on Document:Courses	1	0.1
838	10/4/2009 1:42:...	Save Courses	1	0.1
839	10/4/2009 1:42:...	Valid User: User authorized for Action:Save on Document:Courses	1	0.1
840	10/4/2009 1:42:...	Valid User: User authorized for Action:View on Document:Courses	1	0.1
841	10/4/2009 1:42:...	Valid User: User authorized for Action:Save on Document:Courses	1	0.1
842	10/4/2009 1:42:...	Valid User: User authorized for Action:Delete on Document:Courses	1	-0.15
843	10/4/2009 1:42:...	Valid User: User authorized for Action:Save on Document:Courses	1	0.1
844	10/4/2009 1:42:...	Valid User: User authorized for Action:Delete on Document:Courses	1	-0.15
845	10/4/2009 1:42:...	Valid User: User authorized for Action:View on Document:Courses	1	0.1
846	10/4/2009 1:43:...	Save Courses	1	0.1
847	10/4/2009 1:43:...	Valid User: User authorized for Action:Save on Document:Courses	1	0.1
848	10/4/2009 1:43:...	Valid User: User authorized for Action:View on Document:Courses	1	0.1

Figure 4-13 The Trust Values of the Actions with Details are Recorded in the Database

4.3.4 Trust Value Calculation

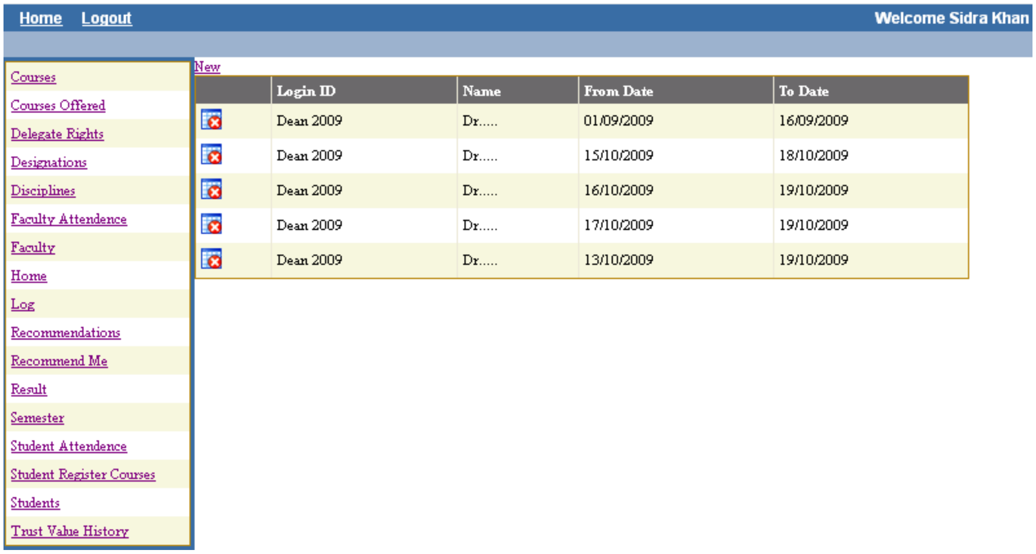
Delegation is a process in which a user containing a set of rights can delegate this role membership to another user or another role for a timestamp. In such a way the delegatee becomes an authorized user for the resources or services allocated with the permission of the delegator. The approach is powerful in a sense that the association partners do not have to change the access policies since a delegate is a replica of the delegator [25]. This leads to an open area of research. It minimizes the amount of administration overhead and facilitates access control in dynamic and pervasive coalition environments.

Types of delegations

According to [26] the types of delegation are:

- ✓ **Time Bound Delegation**: This is a delegation that is valid only for a certain time period.
- ✓ **Group Delegation**: This delegation is to agents from a group who satisfy certain conditions.
- ✓ **Action Restricted Delegation**: This delegation type requires delegate to satisfy certain conditions before the action can be carried out.
- ✓ **Redelegatable Delegation**: This delegation allows that a right can be delegated along with the permission to grant re-delegating the right.
- ✓ **Strictly Redelegatable Delegation**: This allows a right to be re-delegated without giving the delegatee the right to actually execute the action.

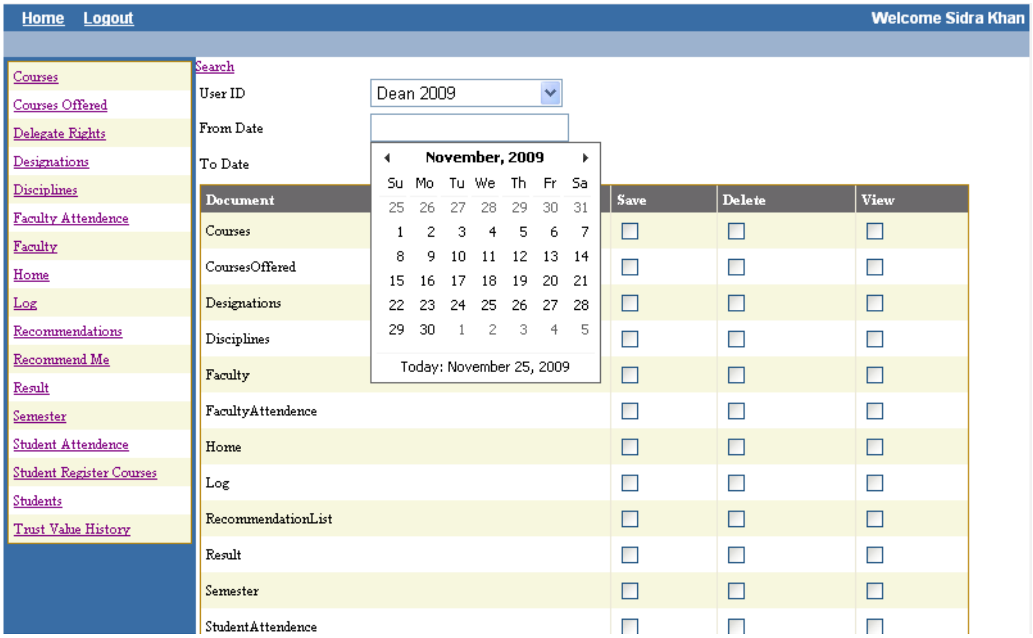
As in the figure 4-14 there are some delegate acknowledgments which show that the dean is delegated from this date to that date by admin. The details of the delegation can be seen from the log maintained in audit log.



	Login ID	Name	From Date	To Date
☒	Dean 2009	Dr.....	01/09/2009	16/09/2009
☒	Dean 2009	Dr.....	15/10/2009	18/10/2009
☒	Dean 2009	Dr.....	16/10/2009	19/10/2009
☒	Dean 2009	Dr.....	17/10/2009	19/10/2009
☒	Dean 2009	Dr.....	13/10/2009	19/10/2009

Figure 0-84 The Set of Delegate Rights

The new option is shown if figure 4-15.



User ID: Dean 2009

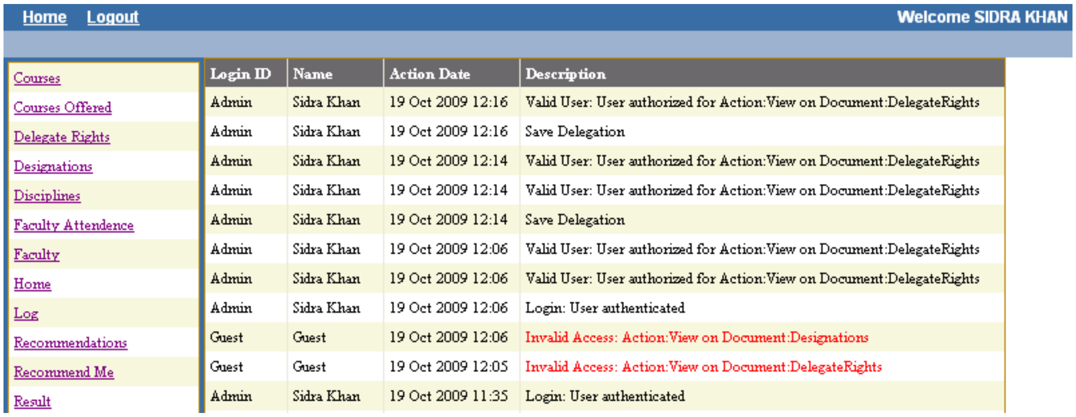
From Date:

To Date:

Document	Save	Delete	View
Courses	□	□	□
CoursesOffered	□	□	□
Designations	□	□	□
Disciplines	□	□	□
Faculty	□	□	□
FacultyAttendence	□	□	□
Home	□	□	□
Log	□	□	□
RecommendationList	□	□	□
Result	□	□	□
Semester	□	□	□
StudentAttendence	□	□	□

Figure 4-15 The New Delegation

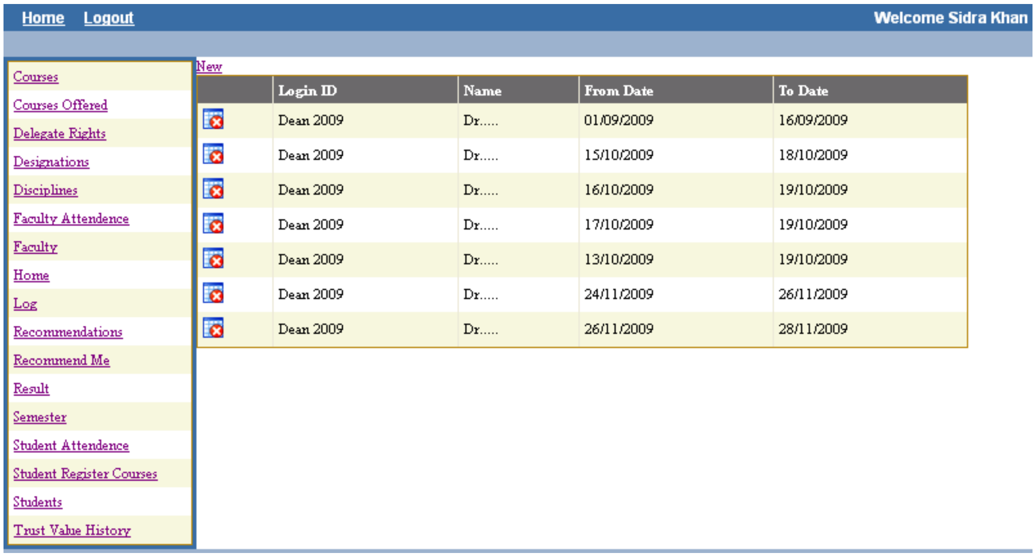
Specific privileges on certain documents are given to the delegetee for the specified time period. The details of the delegation can be seen from the log maintained in audit log in figure 4-16.



Login ID	Name	Action Date	Description
Admin	Sidra Khan	19 Oct 2009 12:16	Valid User: User authorized for Action:View on Document:DelegateRights
Admin	Sidra Khan	19 Oct 2009 12:16	Save Delegation
Admin	Sidra Khan	19 Oct 2009 12:14	Valid User: User authorized for Action:View on Document:DelegateRights
Admin	Sidra Khan	19 Oct 2009 12:14	Valid User: User authorized for Action:View on Document:DelegateRights
Admin	Sidra Khan	19 Oct 2009 12:14	Save Delegation
Admin	Sidra Khan	19 Oct 2009 12:06	Valid User: User authorized for Action:View on Document:DelegateRights
Admin	Sidra Khan	19 Oct 2009 12:06	Valid User: User authorized for Action:View on Document:DelegateRights
Admin	Sidra Khan	19 Oct 2009 12:06	Login: User authenticated
Guest	Guest	19 Oct 2009 12:06	Invalid Access: Action:View on Document:Designations
Guest	Guest	19 Oct 2009 12:05	Invalid Access: Action:View on Document:DelegateRights
Admin	Sidra Khan	19 Oct 2009 11:35	Login: User authenticated

Figure 4-16 The Privileges of the dean delegated by the admin specified with details

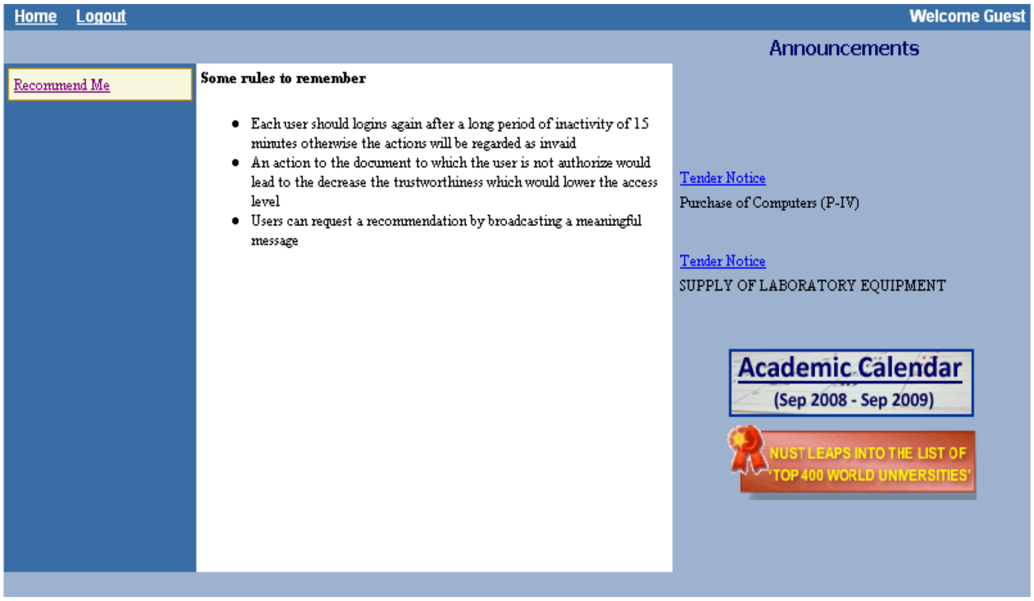
The details of delegation can be seen by the search option which acknowledges the new delegation as portrayed in figure 4-17



	Login ID	Name	From Date	To Date
☒	Dean 2009	Dr.....	01/09/2009	16/09/2009
☒	Dean 2009	Dr.....	15/10/2009	18/10/2009
☒	Dean 2009	Dr.....	16/10/2009	19/10/2009
☒	Dean 2009	Dr.....	17/10/2009	19/10/2009
☒	Dean 2009	Dr.....	13/10/2009	19/10/2009
☒	Dean 2009	Dr.....	24/11/2009	26/11/2009
☒	Dean 2009	Dr.....	26/11/2009	28/11/2009

Figure 4-17 Search shows the acknowledgement of delegations

Figure 4-17 contains the details of the actions like the user was authorized for delegating the right view to Dean and so on. The dean initially, before delegation is treated like a guest and has no privileges and access to documents as shown in figure 4-18.



Some rules to remember

- Each user should logins again after a long period of inactivity of 15 minutes otherwise the actions will be regarded as invaid
- An action to the document to which the user is not authorize would lead to the decrease the trustworthiness which would lower the access level
- Users can request a recommendation by broadcasting a meaningful message

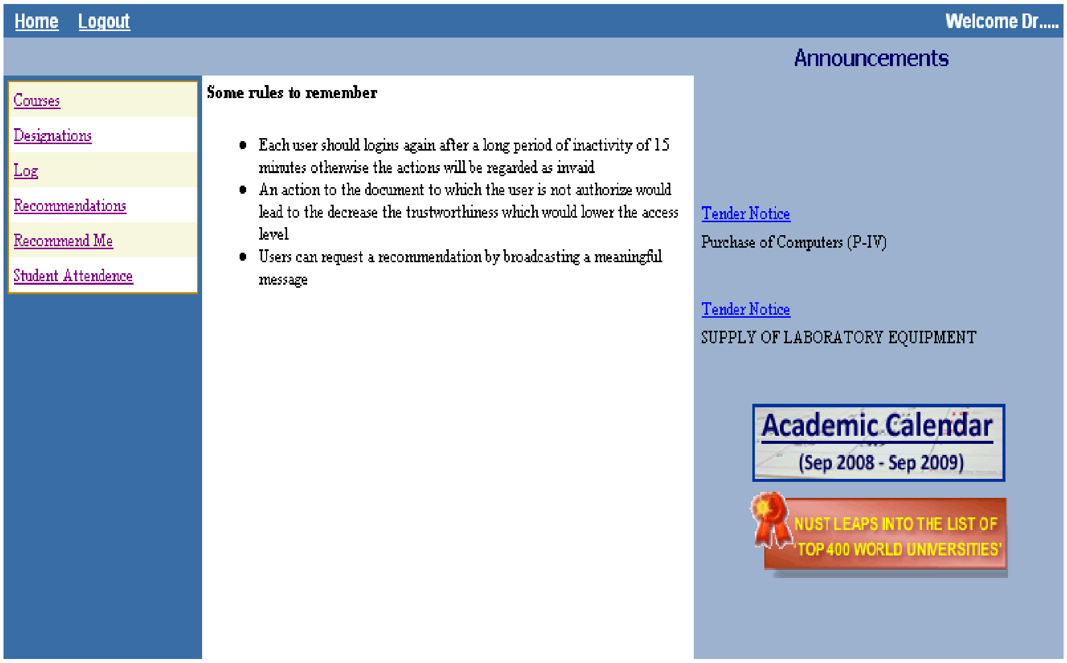
Announcements

Tender Notice
Purchase of Computers (P-IV)

Tender Notice
SUPPLY OF LABORATORY EQUIPMENT

Figure 4-18 Arrival of an Ad-hoc user as Guest or Dean

Dean without any delegations is like a guest which has no access to documents and has no privileges on services as seen in figure 4-18. But admin can delegate rights as defined so that afterwards the dean is responsible for these delegated actions and its log would be maintained likewise defined in figure 4-19. After the expiry of the time the delegated actions are regarded as invalid and become unavailable.



Some rules to remember

- Each user should logins again after a long period of inactivity of 15 minutes otherwise the actions will be regarded as invaid
- An action to the document to which the user is not authorize would lead to the decrease the trustworthiness which would lower the access level
- Users can request a recommendation by broadcasting a meaningful message

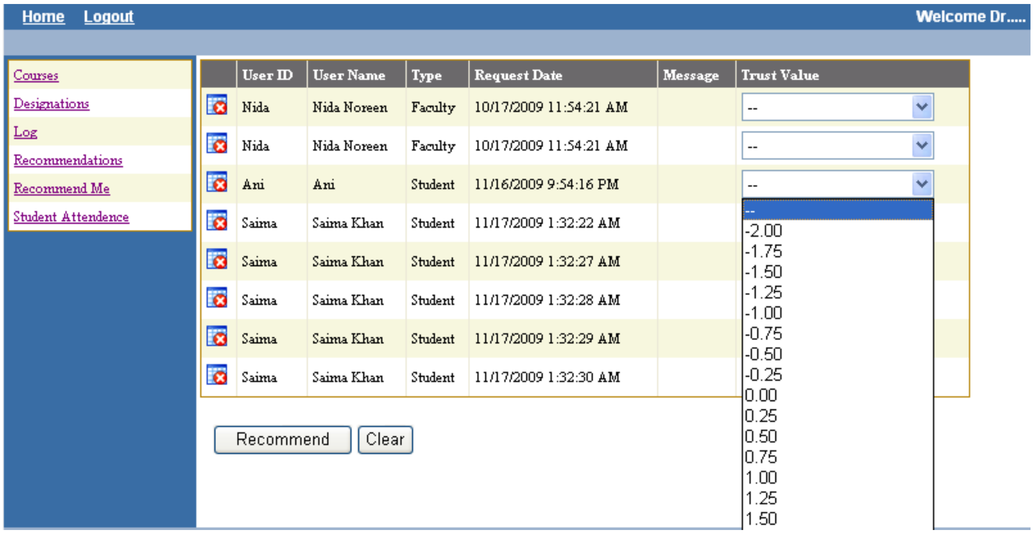
Announcements

Tender Notice
Purchase of Computers (P-IV)

Tender Notice
SUPPLY OF LABORATORY EQUIPMENT

Figure 4-19 Documents on the menu bar which are delegated

The home screen in figure shows the basic rights or documents delegated by the administrator to the dean. The rights to the documents have also been delegated as shown in figure 4-19. The dean is been delegated the right to recommend shown in figure 4-20.



	User ID	User Name	Type	Request Date	Message	Trust Value
☒	Nida	Nida Noreen	Faculty	10/17/2009 11:54:21 AM		--
☒	Nida	Nida Noreen	Faculty	10/17/2009 11:54:21 AM		--
☒	Ani	Ani	Student	11/16/2009 9:54:16 PM		--
☒	Saima	Saima Khan	Student	11/17/2009 1:32:22 AM		
☒	Saima	Saima Khan	Student	11/17/2009 1:32:27 AM		
☒	Saima	Saima Khan	Student	11/17/2009 1:32:28 AM		
☒	Saima	Saima Khan	Student	11/17/2009 1:32:29 AM		
☒	Saima	Saima Khan	Student	11/17/2009 1:32:30 AM		

Figure 4-20 Dean has the Access to do Recommendation on Delegation of this right

Chapter 5

SYSTEM ANALYSIS AND RESULTS

5.1 Introduction

This chapter is based on the analysis of the user behavior which results in increase or decrease of user trust level. The various trust entities used for the computation, modification and assignment of trust categories are analyzed with line and bar graphs and a comparison is also made between direct, indirect, trust values and the number of interactions are made, which reflects effectiveness of the system. The comparison of this technique with the SUKI trust based evaluation framework is also made on the basis of the trust values.

5.2 Weight (W1) of Observation against Trustworthiness

As described in chapter 4 in detail the procedure to calculate the user trust values against each action and recommendation. The recommendations and observations are controlled by a positive nuemeric variable taken as weight which range between 0 to 1. Depending on the context i.e. the application environment, weights are set accordingly. In the development of the framework the weights are defined in such a way that observations are more given weightage than recommendations. To observe the behavior of these multiplying factors in the assumed scenario, we are taking the assumed case I as shown in table 1.

Table 0-2 Assumed Scenario of Case I W.R.T W1

Observation value	**0.5**
Recommendation value	**0.75**
Weight (W2)	**0.2**

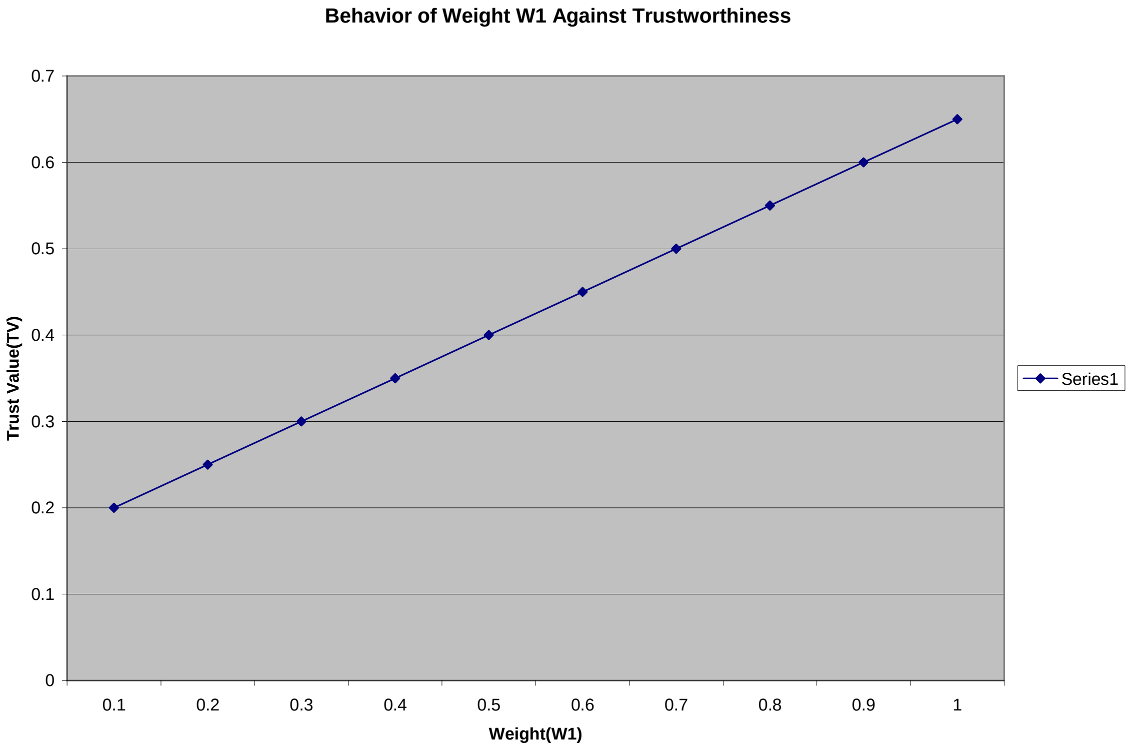


Figure 5-9 Weight (W1) Observation against Trustworthiness

The line Graph 5-1 shows that varying the weights of observations i.e. w1 the trustworthiness of the user is increasing slowly. The rate of increase in trust values is very slow as trust is not easily acquired but is lost easily.

5.3 Weight (W2) Observation against Trustworthiness

The observation weight is kept constant and the parameter w2 of recommendation varies. Now observe the behaviour of line graph.

Table 0-3 Assumed Scenario of Case II W.R.T W2

Observation	**0.5**
Recommendation	**0.75**
Weight W1	**0.8**

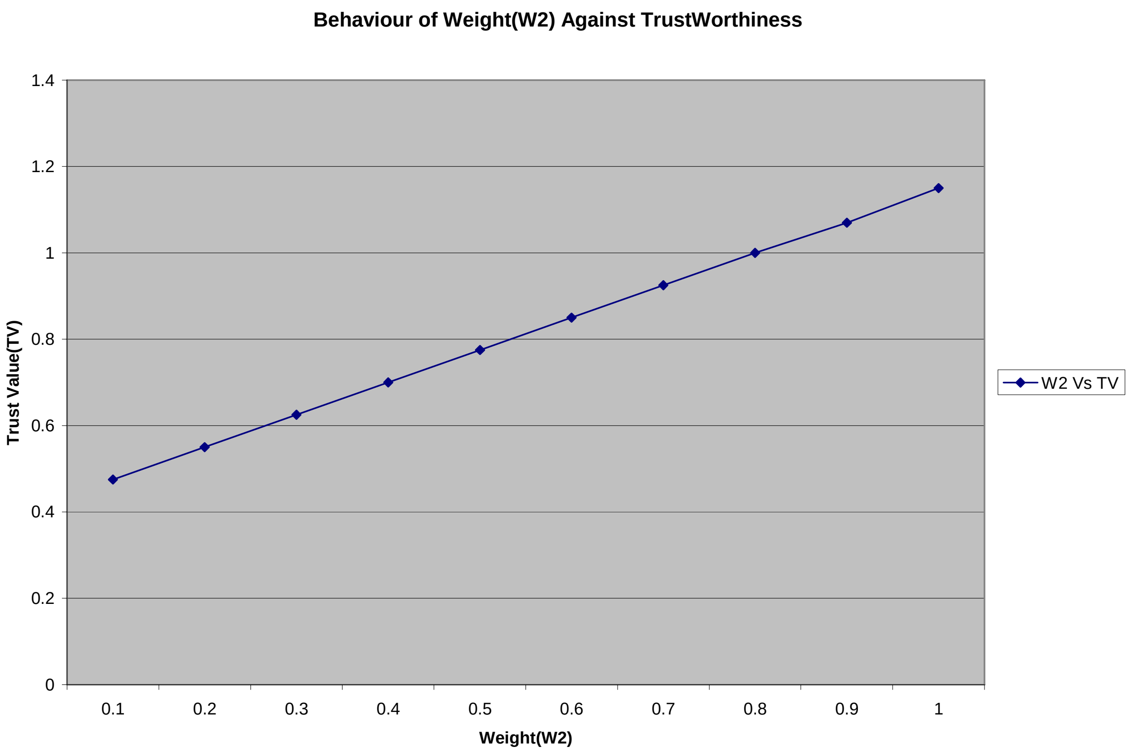


Figure 5-2 Weight (W2) of Recommendation against Trustworthiness

The analysis of the figure 5-2 shows that recommendation increases the rate of the trustworthiness more quickly. So in this project the ad-hoc user is granted the minimum access level initially, then sending Recommend Me message continuously increases the access level of the user if it's recommended positively.

5.4 Comparison of Weight Changing behaviour Against Trustworthiness

The comparison of the figure 5-1 and figure 5-2 is made in figure 5-3 as a line graph. The ad-hoc user gains trust initially by recommendations which are positive else the recommendations takes the trust value to the negative range. On this when user gains access to the services the actions then performed are taken as observations and observations increases the trust level slowly as each interaction is observed and evaluated against the calculate trust value procedure.

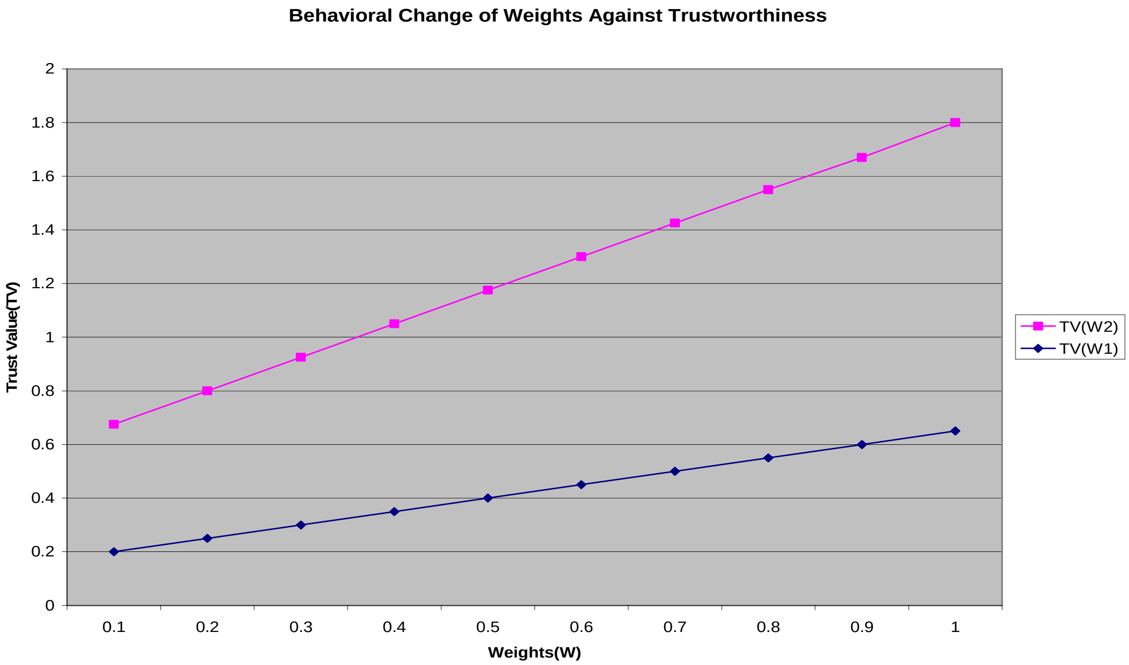


Figure 5-3 Weight Changing behaviour Against Trustworthiness

Figure 5-4 shows the comparison of the Observation weight Vs Recommendation weight with the help of vertical-bar graph.

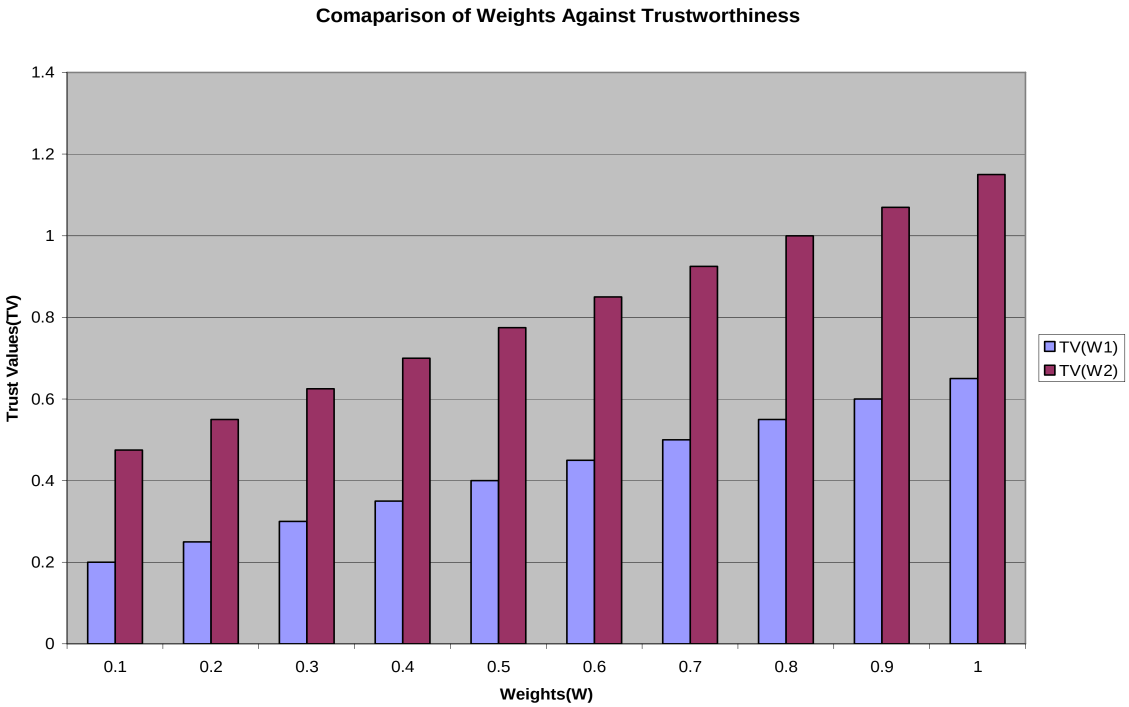


Figure 5-4 Bar Graph Showing Comparisons of Observations and Recommendation Weights with Trust Value

5.5 Analysis of Trusted User Actions against Trust Values

The analysis of the user actions against the trustworthiness is plotted in figure 5-5. The trusted user is doing actions on the services and an observation is applied on the actions. The graph in figure 5-5 shows that the trust level of the user is increasing and increasing but during this increase there are some fall down points in the graph for a bad action.

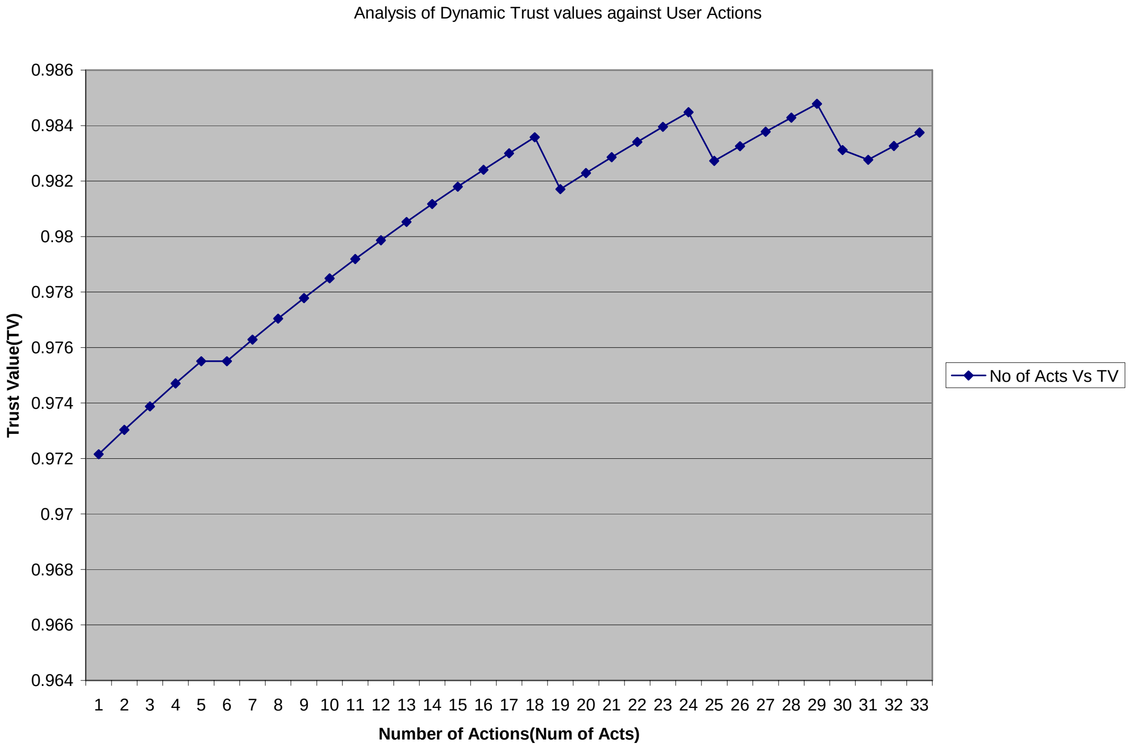


Figure 5-5 Analysis of Trusted User Actions against Trust Values

Figure 5-5shows the repeated valid actions by user keep on increasing the user trust level access while a bad action leads towards decreased trust level. But overall the trustworthiness increases for majority of actions.

5.6 Analysis of Ad-Hoc User Actions against Trust Values

The actions of the ad-hoc user results in the trust level to go up or down widely between trust and distrust.

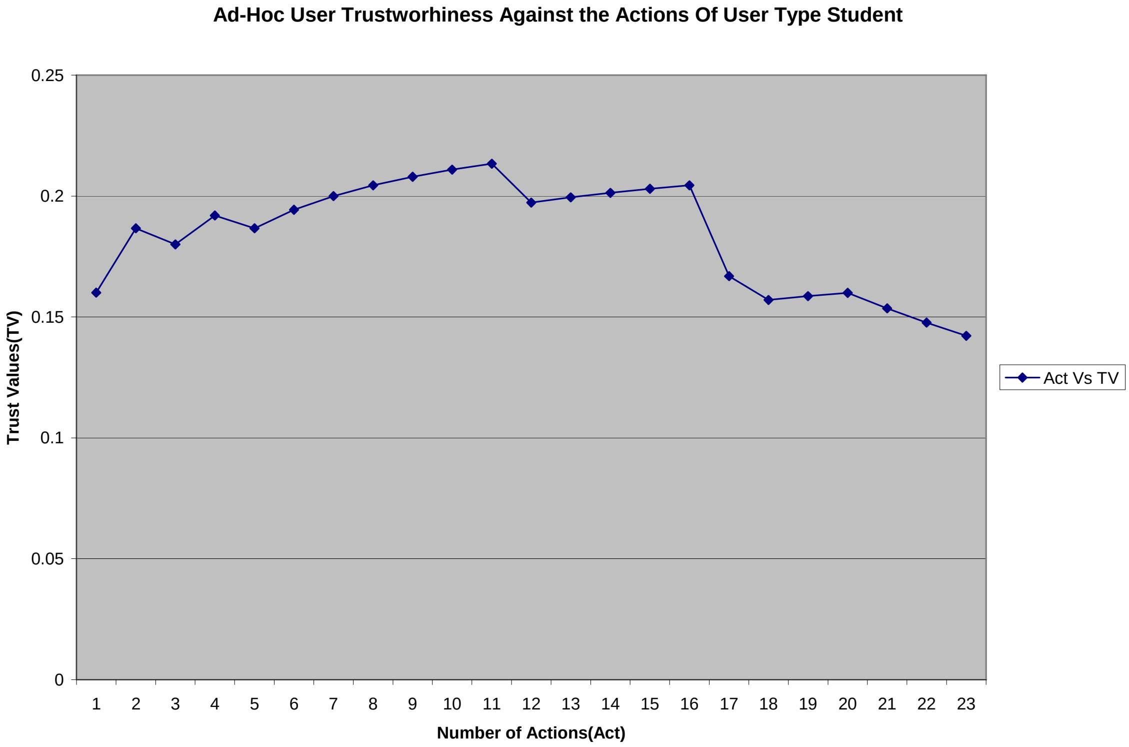


Figure 5-6 Ad-Hoc User's number of actions Vs Trust Value

The varying trustworthiness of the ad-hoc user is cleared from the figure 5-6. It will move towards trustworthiness with positive recommendations and valid interactions.

5.7 Average of the Observations against Trust Value

The figure 5-7 shows the average of all the observations against the trust values. The behavior of the direct trust is mapped for analyzing the user trustworthiness. The direct trust is computed as,

AVG (Obs) = AVERAGE (Observation values) * w1; w1=0.8

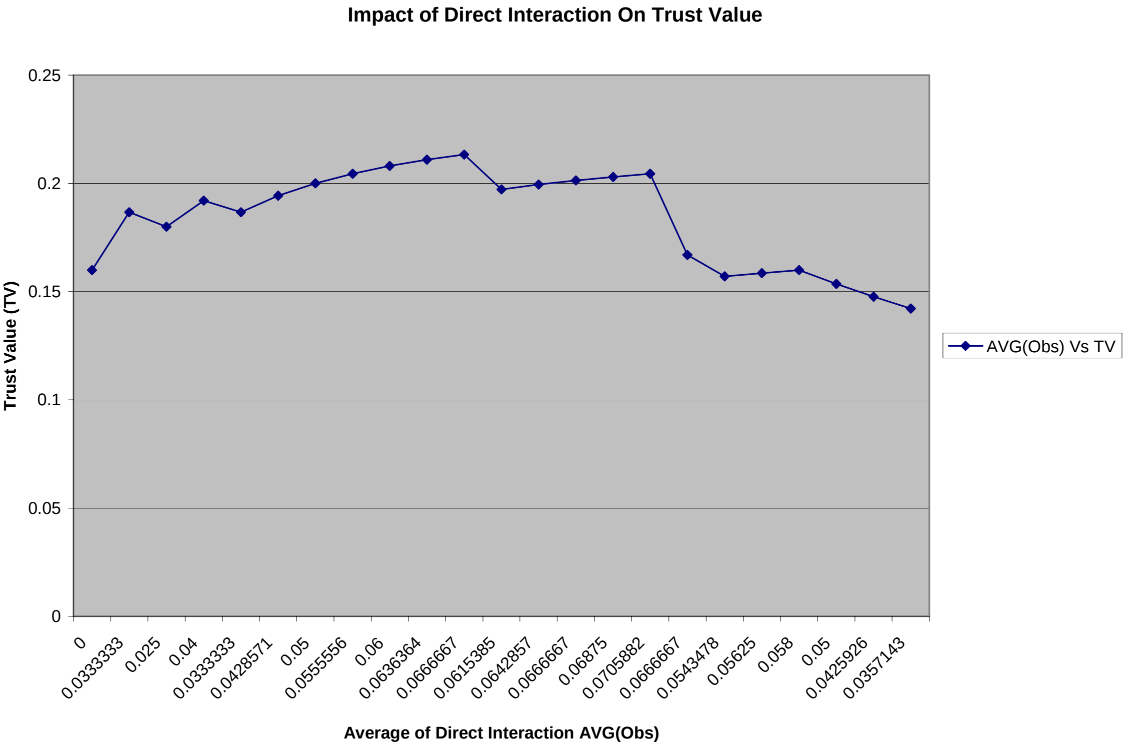


Figure 5-7 Average of the Observations against Trust Value

The user's actions multiplied with the controlling factor are plotted against each resultant trust value of the ad-hoc user displayed in figure 5-7.

5.8 Average of the Recommendation against Trust Values

The figure 5-8 shows the mapping of the user recommended trust values against the user trust values that are ad-hoc.

The AVG(Rec) is calculated as:

AVG(Rec)= Average(Recommended values) * w2;   w2=0.2;

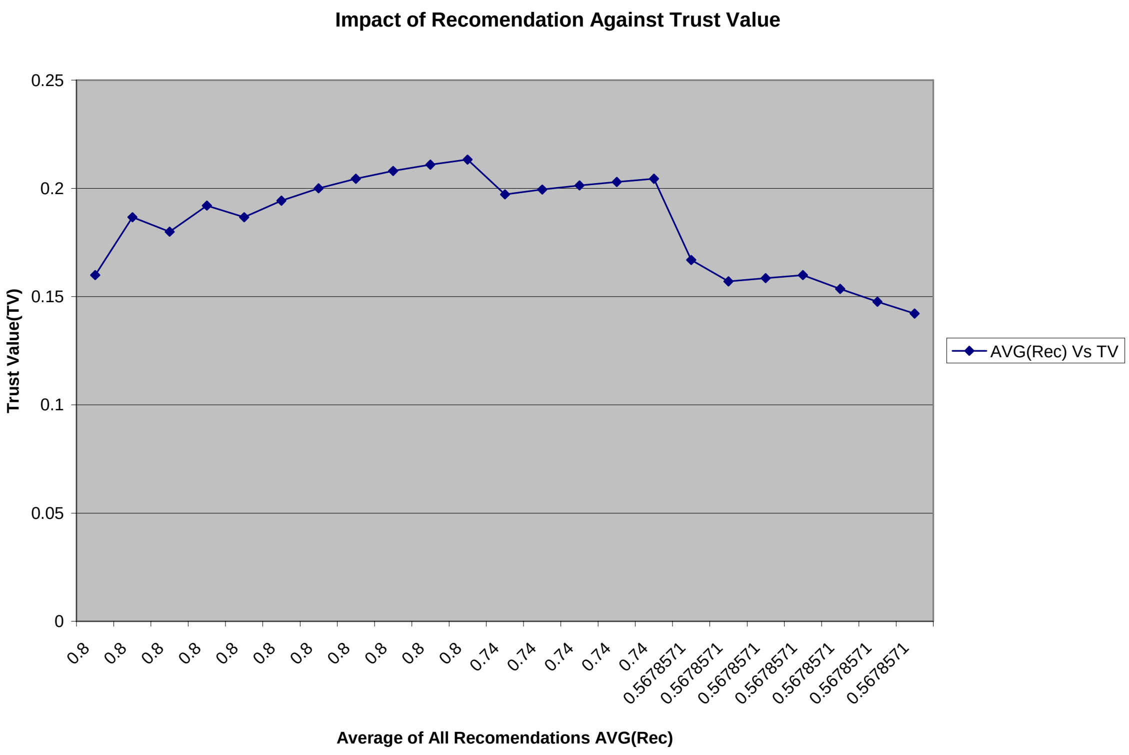


Figure 5-8 Average of Recommendation against Trust Values

The contribution of the average recommended values to the trust values of the ad-hoc user is shown in above line graph where only the factor of recommendation is involved. The greater recommended value increases or decreases the graph abruptly.

5.9 Direct & Indirect Interactions against Trust Value

The comparison of the direct weighted trust and the recommended weighted trust is shown in figure 5-9. The relation of the recommendation and observation is very important due to the fact that the user which is new in the system gains trust due to the recommendations from the trustworthy users initially. The more the user is recommended positively the more trustworthy he/she is, having greater access to services.

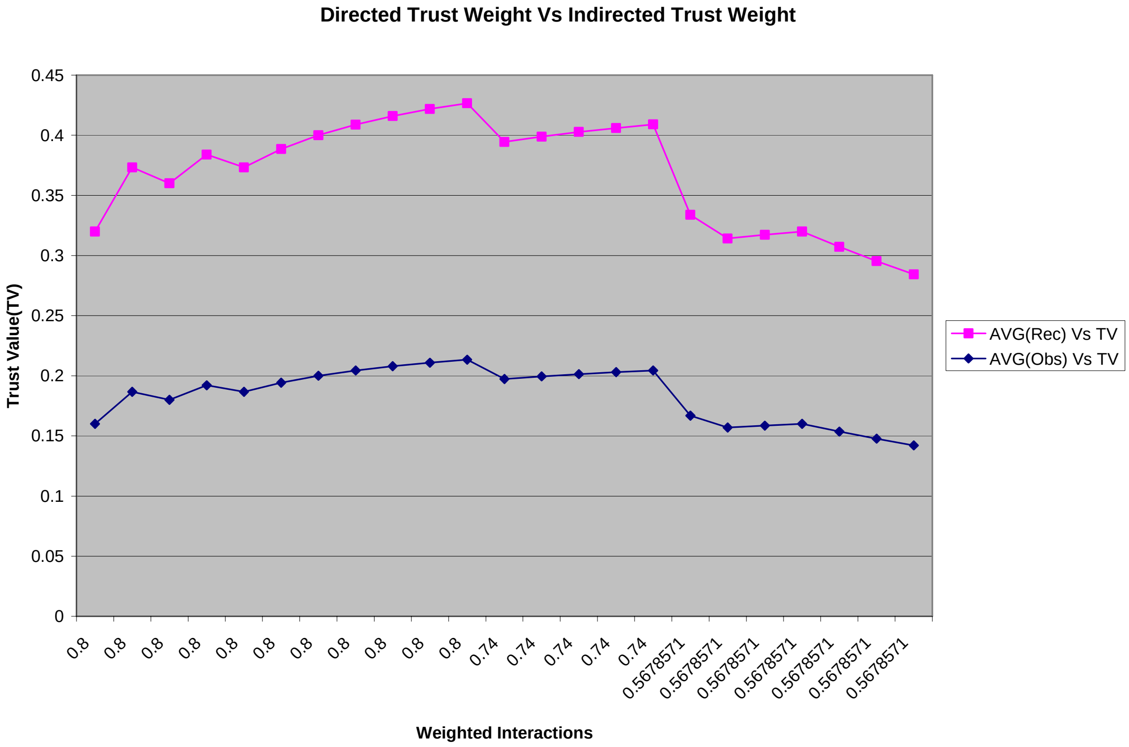


Figure 5-9 Direct & Indirect Interactions against Trust Value

Figure 5-9 shows the line graph of both the weighted observations and the weighted recommendations. The observation change is very small and the recommendation value can be very high or low in between -1 to 1. As 0.1 is assigned to a valid action which leads to overall increase of trust value 0.1 and (-0.25) is added for an invalid action.

5.10 Direct Trust i.e. Observations Against Indirect Trust i.e. Recommendation

The graph in figure 5-10 is the mapping of the direct trust against indirect trust. The domain is the weighted recommendation and the range is the weighted observations.

Domain (X-axis): AVG(Obs)= Average(Observation value) * w1;

w1=0.8;

Range(Y-axis): AVG(Rec)= Average(Recommended values) * w2;

w2=0.2;

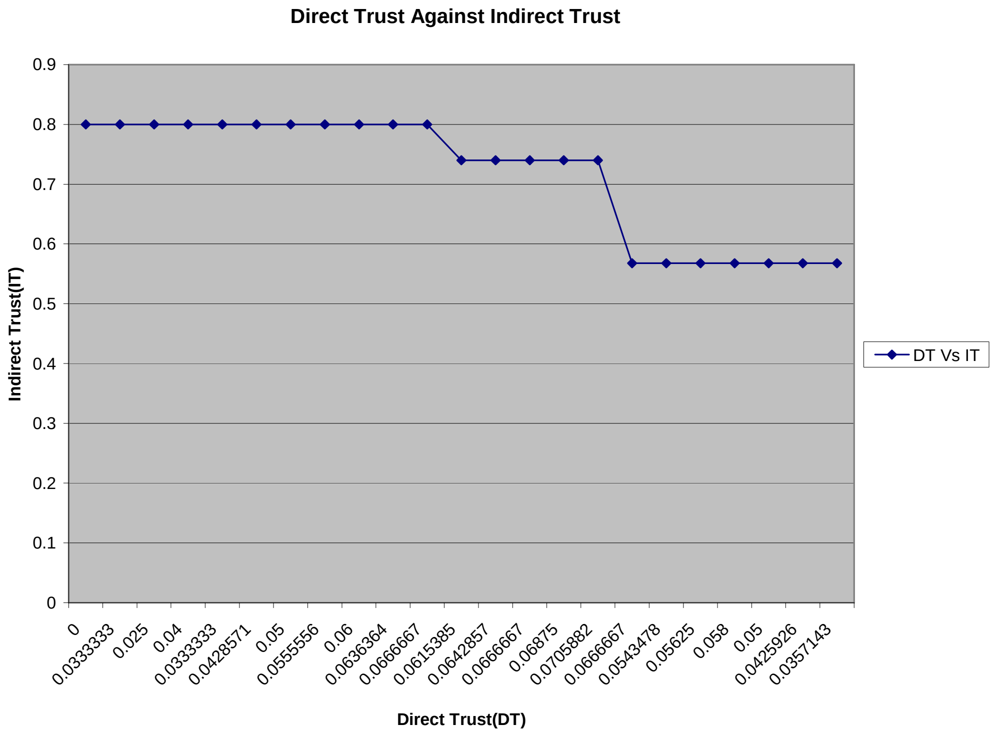


Figure 5-10 Direct Trust i.e. Observations against Indirect Trust i.e. Recommendation

The DT Vs IT line shows that the initially there was no observation then only recommendation was contributing to the user trust value but then user is able to perform some actions and again a set of recommendation against no observations. The indirect trust on x-axis and direct weighted trust is taken on y-axis

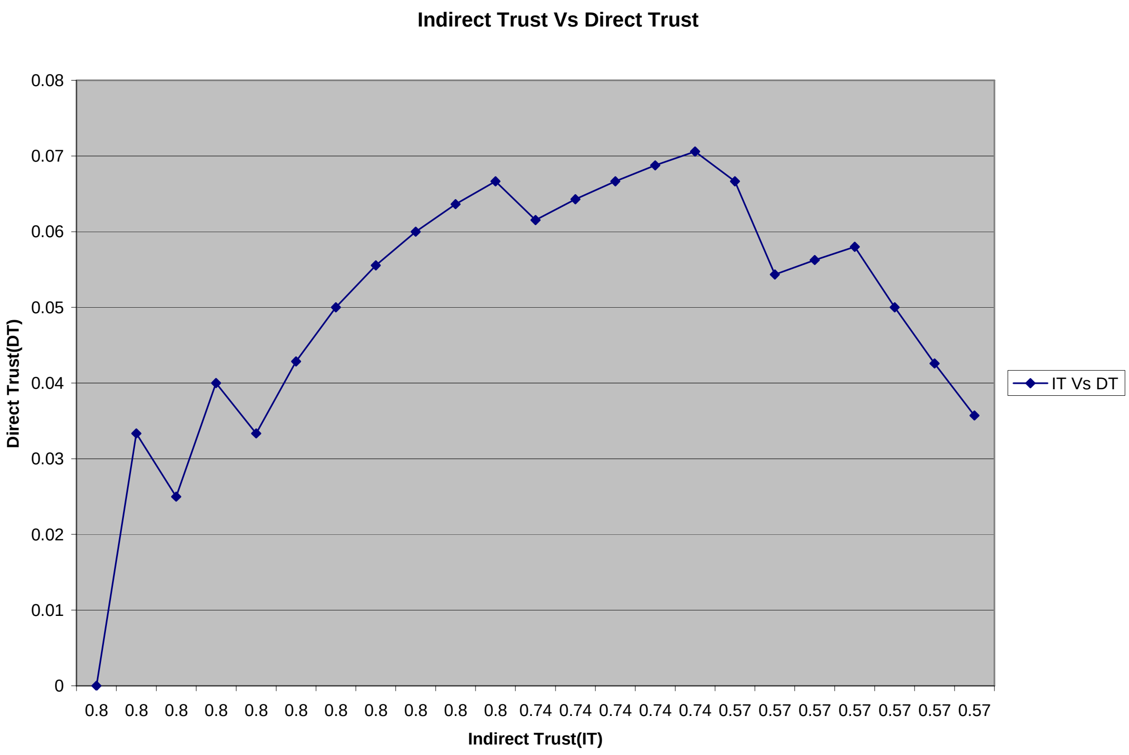


Figure 5-11 Indirect Trust against Direct Trust

Against a recommended value 0.8 the set of observations are shown until the recommended value falls to 0.74 where the user actions are evaluated and drawn in the line graph.

5.11 Evaluation

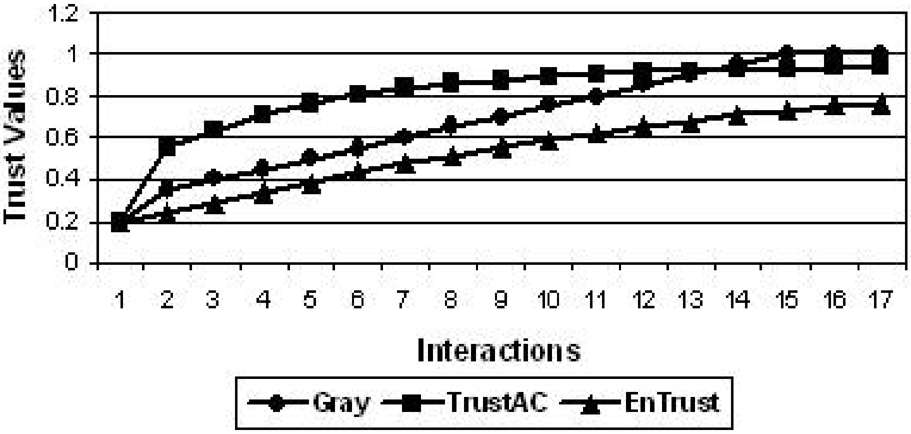


Figure 5-12 Trust Evolution 1

Evaluating trust-based access control for social interaction [27], the figure 5-12 and 5-13 shows the trust evolution under a scenario by three different trust based models like Gray, TrustAC & EnTrust. In figure 5-12 the graph increases for the three models as all the interactions are positive. Where as in figure 5-13 the downfall of the three started as the SPAM message is send. The decrease after the eleventh interaction is sudden in TrustAC, gradually in gray and in between lies the EnTrust.

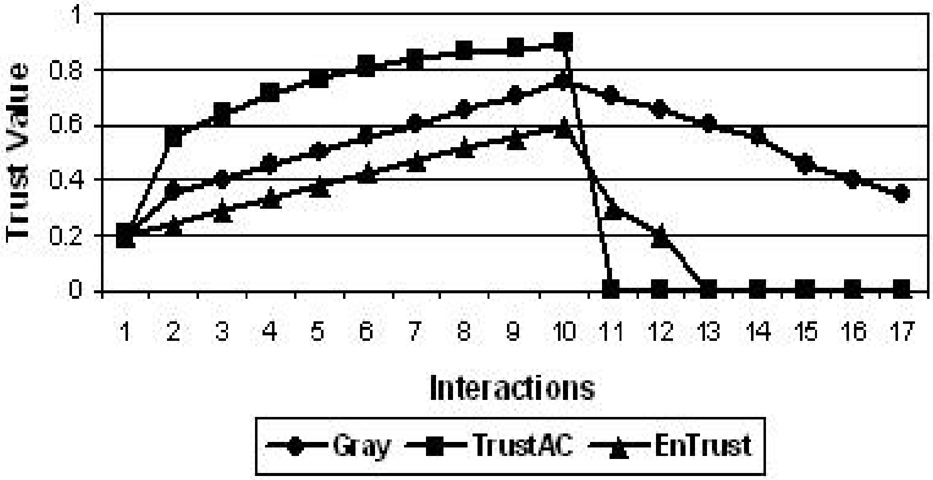


Figure 5-13 Trust Evolution 2

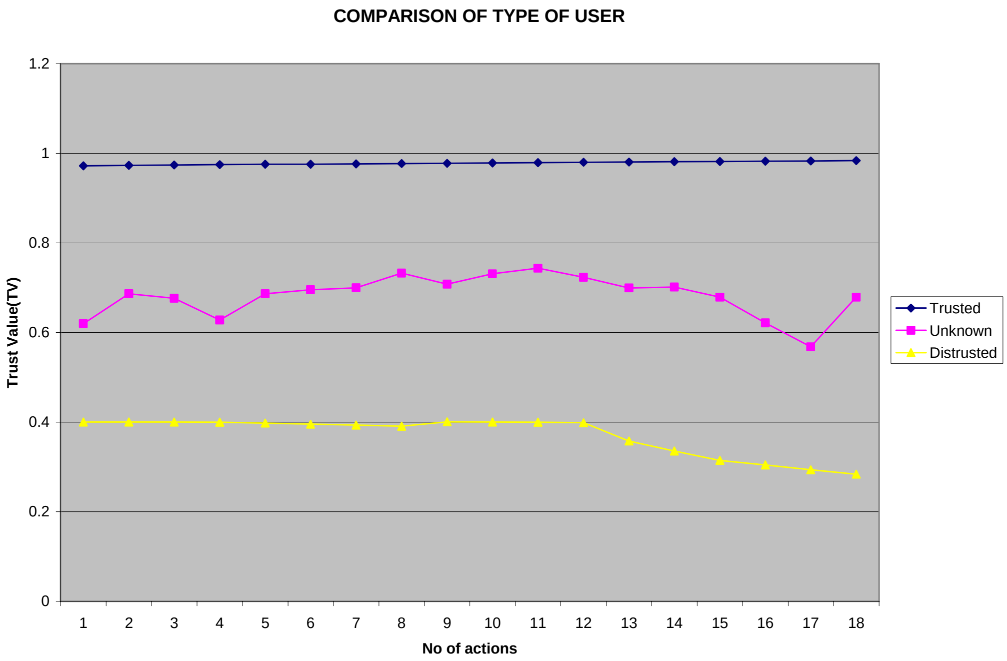


Figure 5-14 Comparison of Trustworthiness of Ad-Hoc and Trusted User

The figure 5-14 shows the behavior for the trusted, unknown user and the distrusted user. The trusted user graph remains to some extent on the same level except for few ups and down. The unknown user is observed for its actions, thus leading to the either way of extremity. As the user trust value crosses the threshold value, more services become available to him/her. Third one is the distrusted user whose trust level goes on decreasing on doing bad actions. Against every action, notice the change at each point. It is better than the other approaches with a fact that the decision made at each action is reliable and smooth. An update in trust level is not abrupt instead occurs in a constant defined way. Good and bad actions both are evaluated for each user type where as in figure 5-13 the three graphs shows the same behavior e.g. for good actions the graph rises whereas for bad ones it falls continuously which does not rise again. It is necessary to know that each entity stores information about untrustworthy entities because distrust differs from not to have any trust value [29].

Evaluation of different models of trust based system depends on the chain of the interactions between the entities. As TBAC models are evaluated under different conditions so it is impossible to compare the pros and cons of the related models [27].

5.12 Conclusion:

The chapter 5 briefs the analysis of the framework developed. The data set is taken by the different users after interrogating with the system. The graphs are made on the various parameters of the distributed trust based access control and the analysis shows that the actions of the users results in their trust level to be either increasing or decreasing. Various properties of the trust are exhibited by the graphs. The relativity of the recommendations and observations are also depicted clearly.

Chapter 6

CONCLUSION & FUTURE WORK

6.1 CONCLUSION

Trust is an important phenomenon for establishing security in a pervasive computing environment for access control and authentication. Trust is a phenomenon which requires a long time to build but one can loose it vey easily. Trust establishes with time and is shattered in no time regarding attitude and behaviour.

For this proposed architecture, a web based framework is developed which is well suited and easily adaptable on any application. Nothing is hardcoded and the computational framework is dynamic, flexible, user friendly, open and scalability. The development on web makes the availability every time and everywhere. It is easily configurable for action on services. As framework is generic so it can be easily upgradable to new versions and dimensions.

The recommendations are not taken from all the entities, but only from trustworthy users who have trust values greater than or equal to threshold value so that the broadcasted request is received, thus making the system more secure and reliable. Log is maintained for each user action from which the observations can be monitored.

The action performed against which the response occurs in the form of the calculation of trust value, these actions trust values also changes dynamically. The level of trust varies with the response of the actions and the recommendation.

Slow development of trust which contributes to the security of the system. The good reputation of a user is not build immediately but takes time. Time is needed to build the trust bridge in reality and so is the case with trust based access control.

Trust is computed by both observation and recommendation to introduce reliability in the system. Furthermore the observations and recommendation are being updated according to the behavior of the user and the context. Weights assigned to observation and recommendations are controlling factors and can change depending on the context.

Users are divided into categories so as to manage the number of ad-hoc users. Category based access level is introduced in this system as the actions are divided into the categories of low, medium and high access. Each category is assigned to the user type. Each invalid access to the service by the user is encountered by the system.

Delegation is time frame based delegation, which is authorized to give delegations to a person on whom the delegator trusts. Combining delegations with trust based access brings a new dimension in the pervasive computing.

6.2 FUTURE WORK

The future work for the existing system can be done in a direction to enhance the security of the system. The policy control module can be introduced in the system with reference to the context. Policies are defined in the policy definition language where the policies and rules related to the authorization can be implemented. As in this system the trusted user can give delegation with time constraint but the added functionality can be the redelegation by the dean to Rector or some other trusted user from the set of right that the delegator possesses. One way to give delegation is on the basis of the request by the delegatee. The details of the request involves the user details, the request message for the services. It can be taken towards the peer to peer system and reviewing.

REFERENCES

1. Authentication in Ubiquitous Computing(2002) , Laurent Bussard , Yves Roudier, France
2. M. Weiser, “The Computer for the 21st Century,” Sci. Amer., Sept., 1991.
3. M. Satyanarayanan, Carnegie Mellon University, Pervasive Computing: Vision and Challenges. IEEE Personal Communications, August 200.
4. http://en.wikipedia.org/wiki/Ubiquitous_service
5. Lakshmi Eswari.P.R, Raghuram.N.C, Chaithanya.M.K, Manjulatha.B, Jyostna.G, Sarat Chandra Babu.N, “A Comprehensive Security, Privacy & Trust Management Framework for Ubiquitous Computing Environment”, Centre for Development of Advanced Computing (C-DAC), Hyderabad
6. TrustAC: Trust-Based Access Control for Pervasive Devices-Florina Almenarez, Andres Marin, Celeste Campo, Carlos Garcia R. Security in pervasive computing. International conference , ALLEMAGNE, april,2005 , vol. 3450, pp. 225-238
7. M. Blaze, J. Ioannidis, and A. Keromytis, “The KeyNote Trust-Management System”, Version 2. RFC 2704, September 1999.
8. Trust-Based Security in Pervasive Computing Environments. In: IEEE Computer, pages 154–157, Dec. 2001
9. Jalal Al-Muhtadi, Anand Ranganathan, Roy Campbell,M. Dennis Mickunas, “Cerberus: A Context-Aware Security Scheme for Smart Spaces”. In IEEE International Conference on Pervasive Computing and Communications (PerCom 2003), Dallas-Fort Worth, Texas, March 23-26, 2003
10. Rainer Steffen, Rudi Knorrs “A Trust Based Delegation System for managing access control”, Third International Conference on Pervasive Computing, 2005, Germany

11. Lalana Kagal, Scott Cost, Timothy Finin, Yun Peng ,"A Framework for Distributed Trust Management"- Second Workshop on Norms and Institutions in MAS, Autonomous Agents. May 01, 2001
12. Munirul Haque and Sheikh Iqbal Ahamed, "Security in Pervasive Computing: Current Status and Open Issues", International Journal of Network Security, vol. 3, No. 3, pp. 203–214, 2006.
13. Tao Sun & Mieso K. Denko "A Distributed Trust Management Scheme in the Pervasive Computing Environment" , Electrical and Computer Engineering, CCECE, Canadian Conference, April 2007
14. Matt Blaze, Joan Feigenbaum, Jack Lacy: Decentralized Trust Management, Proceedings IEEE Conference on Security and Privacy, Oakland CA, 1996
15. Matt Blaze, Joan Feigenbaum, Angelos D. Keromytis: KeyNote: Trust Management for Public-key Infrastructures, In Proc. Cambridge 1998 Security Protocols International Workshop, pages 59--63, 1998.
16. Yang-Hua Chu, Joan Feigenbaum, Brian LaMacchia, Paul Resnick, Martin Strauss: REFEREE: Trust Management for Web Applications, Computer Networks and ISDN Systems Vol 29, 1997
17. Heiko Kopp, Ulrike Lucke, Djamshid Tavangarian, "Security Architecture for Service-based Mobile Environments", Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications Workshops , Pp: 199 – 203, march 2005.
18. Grandinson, T Sloman,M, "A Survey of Trust in Internet Applications" IEEE Communications Surveys 2000.
19. F.AbdulRahman, S.Hailes, "Supporting Trust in Virtual Communities" - Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 6 - Volume 6, Pp: 6007, 2000
20. http://en.wikipedia.org/wiki/Scientific_method
21. http://en.wikipedia.org/wiki/
22. http://en.wikipedia.org/wiki/Prototype

23. V. Shmatikov and C. Talcott. Reputation-based trust management. Journal of Computer Security, 13(1):167–190, 2005.

24. Ramiro Liscano, Kaining Wang, "A Context-Based Delegation Access Control Model for Pervasive Computing," ainaw, vol. 2, pp.44-51, 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07), 2007

25. http://www.stylusinc.com/Common/AboutUs/Services.php

26. Programming for Pervasive Computing Environments Robert Grimm, Janet Davis, eric Lemar, Adam Macbeth, Steven Swanson, Tom Anderson, Brian Bershad, Gaetano Borriello, Steven Gribble, and David Wetherall, 2001

27. Jia, Lang; Collins, Michael and Nixon, Paddy. "Evaluating Trust-based Access Control for Social Interaction".
To appear in proceedings of the Third International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM 2009), October 11 - October 16, 2009, Sliema, Malta.

28. http://searchnetworking.techtarget.com/sDefinition/0, sid7_gci759337,00.html

29. Managing Ad-Hoc Trust Relationships in Pervasive Computing Environments Florina Almenarez, Andres,
Mar´ın, Celeste Campo, Carlos Garcıa, SPPC: Workshop on Security and Privacy in Pervasive Computing, April 20, 2004, Vienna, Austria, The First Workshop on Security and Privacy at the Conference on Pervasive Computing, 2004

30. D. Gambetta. Can We Trust Trust? In D. Gambetta, editor, Trust: Making and Breaking Cooperative Relations, pages 213.238. Basil Blackwell. Oxford, 1990.

31. Weiser, M.: Ubiquitous computing (1997)