**1. INTRODUCTION**

**1.1 Abstract**

The project titled **“Enhanced Cloud Armor: Using Privacy and Preservation Framework for ESMTP Server”** is developed using ASP.NET as front end, Coding language C# and SQL Server as back end. .NET Framework 4.0 used with the version of visual Studio 2010.

Trust management is one of the most challenging issues for the adoption and growth of cloud computing. There are several challenging issues in the trust management such as privacy, security, and availability. These process threatening the entire cloud users. This is because cloud can be access anywhere at any time. The disadvantage levels are equal to the advantage level. The communication across the world is must in the modern age communications through postal may take more time. It may be days or weeks to make the message available to others.

In the New age communication cloud has been introduced. There are various services Available in the cloud server like (IaaS) – Infrastructure as a service, (PaaS)- Platform as a service, (SaaS) – Sofware as a service. From these services for security purpose (TaaS) – Trust as a service has been implemented. But we need more improvement in the TaaS.

In order to improve and overcome the Trust as a Service, various strategy has been implemented, they are follows:

Cloud E-Mail service that deals with the web application that manages the electronic way of communication. Through this project we can create our own cloud mail server with ESMTP – Enhanced Simple Mail Transfer Protocol.

The problem facing in all IT companies are hacking, through hacking a web server the company’s total technology can be stolen from the database. Still this problem is in all IT Companies. And the most important problem is trust. Also confidential code team should be monitor and should make a efficient security ring.

In order to overcome these problems in this project we introducing a secured DNS with enhanced database which supports on cloud mail server. DNS is a relatively simple, text-based protocol, in which one or more recipients of a message are specified along with the message text and possibly other encoded objects performed in the absolute database. The message is then transferred to a remote server using a procedure of queries and responses between the client and server. Either an end-user's email client, MUA (Mail User Agent), or a relaying server's MTA (Mail Transport Agents) can act as an SMTP client in the server database.

Here we introducing a procedure based security methodology called as instruction detection system (IDS) which trace the ip details, date, time and the password percentage of the hacker from the hacker’s side. Hacker’s location can be found out using their ip address. The details will be stored in the database from the server side. The DNS client initiates a TCP connection to server's port 25 (unless overridden by configuration). It is quite easy to test an SMTP server using the telnet program. DNS is a push protocol that does not allow one to pull messages from a remote server on demand.

So that the main object is to create privacy preservation for the confidential database the proposed architecture implements the real world anonymous database by implementing the generalization and suppression. It deals with preventing malicious parties and intrusion using trust aware routing framework. The efficiency and security of data can be achieved by maintaining single database with specific access rights. With the action performed with IDS with ESMTP in Anonymous and Confidential Databases.

**Trust Routing Framework**

Works on ESMTP and Cloud Mail server to customize the user right for Outgoing mail server and user data access

**Aware Routing Framework**

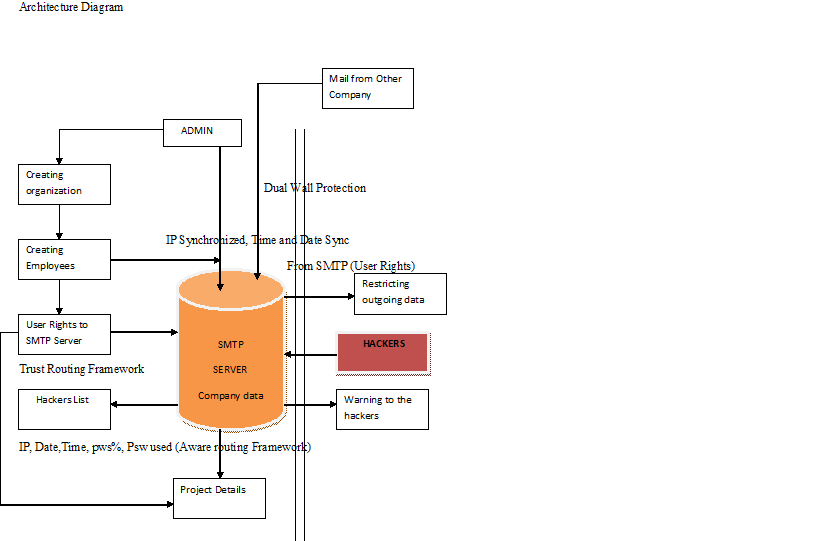
Works on entire cloud sever, and monitors the third party activates and takes security decisions

**1.2 OBJECTIVE**

The Primary objective of this project is to develop a trust aware routing environment using ESMTP server. Here an email environment is developed for an organization; trust is implemented for user rights as well as aware routing is implemented for security purpose. For special security purpose here we introducing a latest method called as IDS (Instruction detection system), which identified the third party intruder or hacker from other networks. The basic IDS can able capture the ip details, here we using a advanced IDS method which can able to capture ip address of the hacker, data, time and the password which he tries to hack. In added with the trust method will provide the user rights within the organization.

* To create secured cloud armor based organization and mail server using Trust and Aware routing (Privacy and preservation)
* Credibility and Trustworthy of the users can be calculated according to the user behavior model
* The whole architecture has been implemented in a mail server.
* Trust aware routing framework has been implanted for Privacy purpose
* Aware routing framework has been implemented for Preservation purpose
* User’s feedback will be taken as the suggestion model. And user’s confidential level can be calculated.
* IDS and IP synchronization has been implemented.
* The result will prove that an enhanced cloud armor has been implemented

**1.3 THE MODEL**

E-mail communication is indispensable nowadays, but the e-mail spam problem continues growing drastically. In recent years, the notion of collaborative spam filtering with near-duplicate similarity matching scheme has been widely discussed. The primary idea of the similarity matching scheme for spam detection is to maintain a known spam database, formed by user feedback, to block subsequent near-duplicate spams. On purpose of achieving efficient similarity matching and reducing storage utilization, prior works mainly represent each e-mail by a succinct abstraction derived from e-mail content text. However, these abstractions of e-mails cannot fully catch the evolving nature of spasm, and are thus not effective enough in near-duplicate detection. In this paper, we propose a novel e-mail abstraction scheme, which considers e-mail layout structure to represent e-mails. We present a procedure to generate the e-mail abstraction using HTML content in e-mail, and this newly devised abstraction can more effectively capture the near-duplicate phenomenon of spasm. Moreover, we design a complete spam detection system Codes (standing for Collaborative Spam Detection System), which possesses an efficient near-duplicate matching scheme and a progressive update scheme. The progressive update scheme enables system Codes to keep the most up-to-date information for near-duplicate detection. We evaluate Codes on a live data set collected from a real e-mail server and show that our system outperforms the prior approaches in detection results and is applicable to the real world.

**1.4 MODULES:**

* Creating Cloud Organization Environment
* Configuring ESMTP
* Trust as a Service
* Implementing Trust routing framework – For Privacy
* Implementing Aware routing framework – For Preservation

**1.5 MODULES DESCRIPTION**

**Company organization Environment:**

This is the initial module of this project. It consists of creating a cloud based organization for the trust aware routing framework procedure. This can be done by web based centralization and other basic details of the company will be provided. This module is only enabled for admin those who creates the company. While creating company all the basic company details should be entered, along with the DNS and ID of the mail server.

**Configuring ESMTP**

Here users are the configuring components of the ESMTP. This is because the object of this thesis to protect the ESMTP by the dual wall protection methods. Here the organization admin can create various users for their company, as well as they can able to share the group mails inside the group of companies. These mails will not be stored in the junk mails, because these all are confidential mails handling in the ESMTP architecture. Whenever the organization is created the default ESMTP will be assigned for both user and the company. The default company DNS will be in xxxx@companyname.com and the default user name in the DNS will be username@companyname.com.

**Trust as a Service**

This is the core module in this project. Here the system will be comes under a security zone for user monitoring. There are three types of trust methods are used in this module namely IP synchronization, Data Synchronization and Time Synchronization. As mentioned above these process will generate warning to the admin in case of any misbehaving activates of the employee. This method will increase the trust level of the employee between the companies. On time monitoring methods will be used here.

**Implementing Trust Routing Framework**

In order to implement the trust routing framework, customized SMTP is needed. So that here customization options of the ESMTP server will take place for providing the user rights. After creating the user by the admin, admin can provide customized user rights to the users. The customizations are customizing compose option, inbox, outbox, sent items and etc. So from the methods the users can be categorized as well as the prior user of the mail server will not be responsible for any other problem. These rights can be given to all the users of this mail server.

**Implementing aware routing framework**

In order to implement the aware routing framework here we are introducing IDS (Instruction detection system). An intrusion detection system (IDS) is a device or software application that monitors network or system activities for malicious activities or policy violations and produces reports to a management station. Some systems may attempt to stop an intrusion attempt but this is neither required nor expected of a monitoring system. Intrusion detection and prevention systems (IDPS) are primarily focused on identifying possible incidents, logging information about them, and reporting attempts. In addition, organizations use IDPSes for other purposes, such as identifying problems with security policies, documenting existing threats and deterring individuals from violating security policies. IDPSes have become a necessary addition to the security infrastructure of nearly every organization.The intrusion detection will capture the unauthorized user’s details like ip address, password used for hacking, Data and time of hacking, Password matching percentage. So the user can identify who is the other intruding into the network. So that user can identify the hackers easily through their ip address.

**2. LITERATURE SURVEY**

**1. Privacy, Security and Trust in Cloud Computing**

**S. Pearson, “Privacy, Security and Trust in Cloud Computing,” in Privacy and Security for Cloud Computing, ser. Computer Communications and Networks, 2013,pp.3–42**

Cloud computing refers to the underlying infrastructure for an emerging model of service provision that has the advantage of reducing cost by sharing computing and storage resources, combined with an on demand provisioning mechanism relying on a pay-per-use business model. These new features have a direct impact on information technology (IT) budgeting but also affect traditional security, trust and privacy mechanisms. The advantages of cloud computing – its ability to scale rapidly, store data remotely, and share services in a dynamic environment – can become disadvantages in maintaining a level of assurance sufficient to sustain confidence in potential customers. Some core traditional mechanisms for addressing privacy (such as model contracts) are no longer flexible or dynamic enough, so new approaches need to be developed to fit this new paradigm. In this chapter we assess how security, trust and privacy issues occur in the context of cloud computing and discuss ways in which they may be addressed.

Although there is no definitive definition for cloud computing, a definition that is commonly accepted is provided by the United States National Institute of Standards and Technologies (NIST): This shared pool of resources is unified through virtualisation or job scheduling techniques. Virtualisation is the creation of a set of logical resources (whether it be a hardware platform, operating system, network resource or other resource) usually implemented by software components that act like physical resources. In particular, software called a ‘hypervisor’ emulates physical computer hardware and thus allows the operating system software running on the virtual platform — a virtual machine (VM) — to be separated from the underlying hardware resources. The resources made available through cloud computing include hardware and systems software on remote data centres, as well as services based upon these that are accessed through the Internet; these resources can be managed to dynamically scale up to match the load, using a pay-per resources business model. Key features advertised are elasticity, multi tenancy, maximal resource utilization and pay-per-use. These new features provide the means to leverage large infrastructures like data centres through virtualization or job management and resource management. Cloud computing (or, more simply, ‘cloud’) provides a market opportunity with a huge potential both for efficiency and new business opportunities (especially in service composition), and is almost certain to deeply transform our information technology infrastructures, models and services. Not only are there cost savings due to economies of scale on the service provider side and pay-as-you-go models, but business risk is decreased because there is less need to borrow money for upfront investment in infrastructure.

The adoption of cloud computing may move quite quickly depending on local requirements, business context and market specificities. We are still in the early stages but cloud technologies are becoming adopted widely in all parts of the world. The economic potential of cloud computing and its capacity to accelerate innovation are putting business and governments under increased pressure to adopt cloud computing based solutions. Although the hype around cloud tends to encourage people to think that it is a universal panacea, this is not the case and quite often promoters ignore the inherent complexities added by the cloud. There are a number of challenges to providing cloud computing services: the need to comply with local and regional regulations, obtaining the necessary approvals when data is accessed from another jurisdiction, some additional complexity in terms of governance, maintenance and liability inherent to cloud, and a perceived lack of trust in cloud services.

**2. Trust Mechanisms for Cloud Computing**

**J. Huang and D.M.Nicol, “Trust Mechanisms for Cloud Computing,” Journal of Cloud Computing, vol.2,no.1,pp.1–14,2013**

Trust is a critical factor in cloud computing; in present practice it depends largely on perception of reputation, and self assessment by providers of cloud services. We begin this paper with a survey of existing mechanisms for establishing trust, and comment on their limitations. We then address those limitations by proposing more rigorous mechanisms based on evidence, attribute certification, and validation, and conclude by suggesting a framework for integrating various trust mechanisms together to reveal chains of trust in the cloud.

Cloud computing has become a prominent paradigm of computing and IT service delivery. Furthermore, what exactly does “trust” mean in the context of cloud computing? What is the basis of that trust? If the attributes of a cloud service (or a service provider) are used as evidence for trust judgment on the service (or provider respectively), on what basis should users believe the attributes claimed by cloud providers? Who are authorities to monitor, measure, assess, or validate cloud attributes? The answers to those questions are essential for wide adoption of cloud computing and for cloud computing to evolve into a trustworthy computing paradigm. As addressed in, “the growing importance of cloud computing makes it increasingly imperative that we grapple with the meaning of trust in the cloud and how the customer, provider, and society in general establish that trust.” The issues and challenges of trust in cloud computing have been widely discussed from different perspectives. A number of models and tools have been proposed. Each contributes a partial view of cloud trust, but lacking still is a complete picture illustrating how cloud entities work together to form a “societal” system, with a solid grounding in trust, serving to facilitate trusted paths to trusted cloud services. The NIST Cloud Computing Reference Architecture identified cloud brokers and cloud auditors as entities who conduct assessment of cloud services; however, there are few studies on trust relation analysis and the chains of trust from cloud users to cloud services (or providers) through those intermediary cloud entities. In this paper, we investigate trust mechanisms for the cloud, present our vision of the “societal systems mechanisms” of trust and a framework for analyzing trust relations in the cloud, and suggest trust mechanisms which combine attribute certification, evidence-based trust and policy-based trust.

Because of the criticality of many computing services and tasks, some cloud clients cannot make decisions about employing a cloud service based solely on informal trust mechanisms (e.g. web-based reputation scores); these decisions need to be based on formal trust mechanisms, which are more certain, more accountable, and more dependable. Here, the word “formal” is meant to carry the sense of “official” assessment in a society. In our suggested cloud trust mechanisms, the attributes of a cloud service (or its provider) are used as evidence for the user’s trust judgment on the service (or provider), and the belief in those attributes is based on “formal” certification and chains of trust for validation.

In this paper, we focus somewhat informally on the conceptual basis for analysis of trust in the cloud; we do not at this time address mathematical modelling, which would involve many more precise details, formal languages, and specific use cases.

**3. Trusted Cloud Computing with Secure Resources and Data Colouring**

**K.Hwang and D.Li, “Trusted Cloud Computing with Secure Resources and Data Coloring,” IEEE Internet Computing, vol.14, no.5,pp.14–22,2010**

Trust and security have prevented businesses from fully accepting cloud platforms. To protect clouds, providers must first secure virtualized data centre resources, uphold user privacy, and preserve data integrity. The authors suggest using a trust-overlay network over multiple data centres to implement a reputation system for establishing trust between service providers and data owners. Data colouring and software watermarking techniques protect shared data objects and massively distributed software modules. These techniques safeguard multi-way authentications, enable single sign-on in the cloud, and tighten access control for sensitive data in both public and private clouds.

Cloud computing enables a new business model that supports on demand, pay-for-use, and economies-of-scale IT services over the Internet. The Internet cloud works as a service factory built around virtualized data centres. Cloud platforms are dynamically built through virtualization with provisioned hardware, software, networks, and datasets. The idea is to migrate desktop computing to a service oriented platform using virtual server clusters at data centres. However, a lack of trust between cloud users and providers has hindered the universal acceptance of clouds as outsourced computing services. To promote multi tenancy, we must design the cloud ecosystem to be secure, trustworthy, and dependable.

In reality, trust is a social problem, not a purely technical issue. However, we believe that technology can enhance trust, justice, reputation, credibility, and assurance in Internet applications. To increase the adoption of Web and cloud services, cloud service providers (CSPs) must first establish trust and security to alleviate the worries of a large number of users. A healthy cloud ecosystem should be free from abuses, violence, cheating, hacking, viruses, rumours, pornography, spam, and privacy and copyright violations. Both public and private clouds demand “trusted zones” for data, virtual machines (VMs), and user identity, as VMware and EMC 3 originally introduced.

Data integrity issues in the cloud differ from those in traditional database systems. Cloud users are most concerned about whether data-centre owners will abuse the system by randomly using private datasets or releasing sensitive data to a third party without authorization. Cloud security hinges on how to establish trust between these service providers and data owners. To address these issues, we propose a reputation-based trust-management scheme augmented with data colouring and software watermarking. Information about related trust models is available elsewhere.

The Cloud Security Alliance has identified a few critical issues for trusted cloud computing, and several recent works discuss general issues on cloud security and privacy. Public and private clouds demand different levels of security enforcement. We can distinguish among different service-level agreements (SLAs) by their variable degree of shared responsibility between cloud providers and users. Critical security issues include data integrity, user confidentiality, and trust among providers, individual users, and user groups. The infrastructure-as-a-service (IaaS) model sits at the innermost implementation layer, which is extended to form the platform-as-a service (PaaS) layer by adding OS and middleware support. PaaS further extends to the software-as-a-service (SaaS) model by creating applications on data, content, and metadata using special APIs. This implies that SaaS demands all protection functions at all levels. At the other extreme, IaaS demands protection mainly at the networking, trusted computing, and compute/storage levels, whereas PaaS embodies the IaaS support plus additional protection at the resource-management level.

**4. Towards a Trust Management System for Cloud Computing.**

**S.Habib, S.Ries, and M.Muhlhauser, “Towards a Trust Management System for Cloud Computing,” in Proc.of TrustCom’11, 2011**

Cloud computing provides cost-efficient opportunities for enterprises by offering a variety of dynamic, scalable, and shared services. Usually, cloud providers provide assurances by specifying technical and functional descriptions in Service Level Agreements (SLAs) for the services they offer. The descriptions in SLAs are not consistent among the cloud providers even though they offer services with similar functionality. Therefore, customers are not sure whether they can identify a trustworthy cloud provider only based on its SLA. To support the customers in reliably identifying trustworthy cloud providers, we propose a multi-faceted Trust Management (TM) system architecture for a cloud computing marketplace. This system provides means to identify the trustworthy cloud providers in terms of different attributes (e.g., security, performance, compliance) assessed by multiple sources and roots of trust information

Cloud computing offers dynamic, scalable, shared resources (e.g., computing power, storage, software) over the internet from remote data centres to the users (e.g., business organizations, government authorities, individuals). The highly distributed and non-transparent nature of cloud computing represents a considerable obstacle for the acceptance and market success of cloud services. Potential customers of these services often feel that they lose the control over their data, and they are not sure whether they can trust the cloud providers. A recent survey, conducted among more than 3000 cloud consumers from 6 countries, shows that 84% of the consumers are concerned about their data storage location and 88% of the consumers worry about who has access to their data.

Consumer concerns can be mitigated by using preventive measures for privacy (e.g., demonstrating compliance standards) and security (e.g., secure hypervisors, TPM based servers). At present, although cloud providers demonstrate their preventive measures by including related descriptions in the SLAs, assurances and compensations for SLA violations are not convincing enough for the consumers. Especially, SLAs with vague clauses and unclear technical specifications lead the consumers into a decision dilemma when considering them as the only basis to identify trustworthy providers.

As the business market is growing rapidly with new providers entering the market, cloud providers will increasingly compete for customers by providing services with similar functionality. However, there can be huge differences regarding the provided quality level of those services. Such a competitive market needs means to reliably assess the quality level of the service providers. Trust and reputation (TR) systems are successfully used in numerous application scenarios to support users in identifying the reliable and trustworthy providers, e.g., on eBay, Amazon, and app markets for mobile applications. Similar approaches are needed to support customers in selecting appropriate trustworthy cloud providers.

Existing TR systems rely on customer feedback without considering other sources and roots of information (e.g., property certificates, compliance with audit standards, provider statements). Moreover, there are additional parameters that are required to support the customers in selecting providers in a cloud marketplace. Therefore, TR systems have to evolve into Trust Management (TM) systems – as defined in – to support the customers in making transparent assessments before selecting reliable trustworthy cloud providers.

**5. Reputation Attacks Detection for Effective Trust Assessment of Cloud Services**

**T.H.Noor, Q.Z.Sheng, and A.Alfazi, “Reputation Attacks Detection for Effective Trust Assessment of Cloud Services,” in Proc.of TrustCom’13, 2013**

Consumers’ feedback is a good source to help assess overall trustworthiness of cloud services. However, it is not unusual that a trust management system experiences malicious behaviours from its users (i.e., collusion or Sybil attacks). In this paper, we propose techniques for the detection of reputation attacks to allow consumers to effectively identify trustworthy cloud services. We introduce a credibility model that not only identifies misleading trust feedbacks from collusion attacks but also detects Sybil attacks, either strategic (in a long period of time) or occasional (in a short period of time). We have collected a large collection of consumer’s trust feedbacks given on real-world cloud services (over 10,000 records) to evaluate and demonstrate the applicability of our approach and show the capability of detecting such malicious behaviours.

Due to highly dynamic, distributed and non-transparent nature of cloud services, trust management is considered as a challenging problem in cloud environments. SLAs alone are inadequate to establish trust between cloud consumers and providers because of its unclear and inconsistent clauses. For instance, in a recent survey, 46.6% of consumers agree that SLA’s legal contents are unclear. Many researchers agree that consumers’ feedback is a good source to assess trust and proposed trust management techniques based on feedback. However, in reality, it is not unusual that a trust management service experiences malicious behaviours (e.g., collusion or Sybil attacks) from its users. Attackers can trick users into trusting untrustworthy cloud services through creating several accounts, producing numerous transactions (e.g., creating multiple virtual machines for a short period of time), and leaving misleading trust feedbacks. Such manipulation makes it hard for consumers to identify trustworthy providers.

This paper focuses on the detection of reputation attacks to allow consumers to effectively identify trustworthy cloud services. The detection of reputation attacks involves several issues including i) Consumers Dynamism where new users join the cloud environment and old users leave around the clock which makes the detection of feedback collusion a significant challenge, ii) Multiplicity of Identities where users may have multiple accounts for a particular cloud service1 which makes it difficult to detect whether a Sybil attack is performed because multiple identities can be used to give misleading information, iii) Attackers Behaviours where it is difficult to predict when such malicious behaviours take place either in a long or short period of time (i.e., strategic vs. occasional behaviours), and iv) Consumers’ Privacy where the detection of attacks can make users subject to privacy breaches especially when the interactions involve sensitive information.

In this paper, we overview the design and the implementation of a credibility model that allows consumers to effectively identify trustworthy cloud services. Our model exploits novel techniques that help detect collusion and Sybil attacks without breaching consumers’ privacy. In a nutshell, the salient features of our model are i) Zero-Knowledge Credibility Proof Protocol that enables the trust management service to prove the credibility of consumers’ feedback without breaching consumers’ privacy (i.e., without the use of sensitive information); ii) Collusion Attacks Detection where we propose several detection metrics including the Feedback Density and Occasional Feedback Collusion to distinguish between misleading and credible feedbacks no matter attacks occur in a strategic or occasional behaviour (i.e., attackers who intend to manipulate the trust results by giving multiple feedbacks to a certain cloud service in a long or short period of time); iii) Sybil Attacks Detection where we propose several metrics for the Sybil attacks detection including the Multi-Identity Recognition and Occasional Sybil Attacks to identify misleading feedbacks from Sybil attacks that occur strategically or occasionally (i.e., attackers who create multiple identities and give feedbacks in a short or long period of time to trick consumers into trusting cloud services that are not trustworthy).

**6. Service-oriented Computing and Cloud Computing: Challenges and Opportunities**

**Y .W ei and M.B.Blake, “Service-oriented Computing and Cloud Computing: Challenges and Opportunities,” Internet Computing, IEEE, vol.14,no.6, pp.72–75,2010.**

SOC (service-oriented computing) represents a new generation of distributed computing framework for applications development by means of the composition of services. The visionary promise of SOC is a world-scale network of loosely coupled services that can be assembled with little effort in agile applications that may span organizations and computing platforms. Since services may be offered by different enterprises and communicate over the Internet, they provide an advanced distributed computing infrastructure for both intra- and cross-enterprise application integration and collaboration. The distinction between SOC and traditional computing is that application builders no longer construct software using a programming language. Instead, they specify the application logic in a high-level specification language, utilizing standard services as components. Services implement functions that can range from answering simple requests to executing sophisticated business processes requiring peer-to-peer relationships between possibly multiple layers of service consumers and providers. The delivery of software as a set of distributed services can help to solve problems like software reuse, deployment and evolution. The “software as a service model” will open the way to the rapid creation of new value-added composite services based on existing ones. Although service-oriented computing in cloud computing environments presents a new set of research challenges, their combination provides potentially transformative opportunities. So any user community, regardless of its discipline, should be supplied with the technological approach to build their own distributed compute-intensive multidisciplinary applications rapidly. This paper contains the roadmap of development of the Engineering Design Platform, based on SOC and intended, in particular, for modelling and optimization of Nonlinear Dynamic Microsystems, being consisted of components of different physical nature and being widely spread in different scientific and engineering fields.

The SOC (service-oriented computing) paradigm refers to the set of concepts, principles, and methods that represent computing in SOA (service-oriented architecture) in which software applications are constructed based on independent component services with standard interfaces. SOC represents a new approach in application development moving away from tightly coupled monolithic software towards software of loosely coupled, dynamically bound services. Since services may be offered by different enterprises and communicate over the Internet, they provide a distributed computing infrastructure for both intra- and cross-enterprise application integration and collaboration. Service clients (end-user organizations that use some service) and service aggregators (organizations that consolidate multiple services into a new, single service offering) utilize service descriptions to achieve their objectives. So, SOC is a paradigm and Service Oriented Architecture is an architectural model which allows interoperability, re-usability, loose-coupling of its components and provides mechanisms to describe publish and discover available services.

The distinction between SOC and traditional computing is that application builders no longer construct software from scratch using a programming language. Instead, they specify the application logic in a high-level specification language, utilizing standard services as components. The central definition being used here is a service, the most important concept of the service-oriented paradigm. The definition of service for the W3C Working Group is: “A service is an abstract resource that represents a capability of performing tasks that form a coherent functionality from the point of view of provider entities and requester entities. To be used, a service must be realized by a concrete provider agent.” This definition is correct but it is too abstract because too many things could be a service. There are various definitions of a service within the context of Service Oriented Computing in the literature, among of which there are the following: “A service is a system function that is well defined, self-contained and does not depend on the context or state of other services”; “A service is a unit of work to be performed on behalf of some computing entity, such as a human user or another program”. Services perform functions that can range from answering simple requests to executing sophisticated business processes requiring peer-to-peer relationships between possibly multiple layers of service consumers and providers.

**7. Peer trust: Supporting Reputation-based Trust for Peer-to-Peer Electronic Communities**

**L.Xiong and L.Liu, “Peertrust: Supporting Reputation-based Trust for Peer-to-Peer Electronic Communities,” IEEE Transactions on Knowledge and Data Engineering, vol.16,no.7,pp.843– 857,2004**

The open and anonymous nature of a P2P network makes it an ideal medium for attackers to spread malicious content. In this paper, we propose a reputation-based trust management system for P2P networks that aims to build confidence among the good members of the community and identify the malicious ones. The proposed system is simple and efficient in design and can be integrated into most first generation P2P systems easily. A diverse set of simulation experiments conducted to test the performance of the system show that it can be highly effective in preventing the spread of malicious content. The proposed system has other potential benefits as well, such as supporting the detection of free riders in a file sharing application.

A peer-to-peer (P2P) network is a computer network that does not have fixed clients and servers but a number of peer nodes that function as both clients and servers to the other nodes in the network. Although in general any networking technology that uses this model can be considered as P2P, such as the NNTP protocol used for transferring Usenet news or a wireless ad hoc network, the term is most frequently used to refer to file sharing networks over the Internet, such as Gnutella, FastTrack, and Napster. We also focus on this particular application of the more general P2P concept in this paper and use the term to refer to P2P file sharing systems unless otherwise is stated. By the nature of its architecture, a P2P file sharing system provides an open, unrestricted environment for content sharing. This openness of a P2P network also makes it an ideal environment for attackers to spread malicious content, such as the VBS. Gnutella worm. Reputation-based systems are used to establish trust among members of on-line communities where parties with no prior knowledge of each other use the feedback from their peers to assess the trustworthiness of members of the community. One well-known such system is the rating scheme used by the eBay on-line auction site. In this paper, we propose a reputation-based distributed trust architecture for P2P networks to identify malicious peers and to prevent the spread of malicious content. The protocol is based on the query-response architecture of the first generation P2P networks and is suitable for operation in a Gnutella- or Kazaa-like system. The design is simple and intuitive and, as the simulation results show, it can be highly effective in preventing the malicious content.

When a resource is queried in a P2P file sharing system, many responses offering various alternative versions may be received, among which some may be malicious. An important security question at this juncture is how to distinguish the malicious alternatives from the benign ones. Our protocol aims to achieve this distinction by evaluating the alternatives according to the reputation of the peers who offer them. When the reputation information is not available locally, the P2P infrastructure is used to query the necessary information over the network.

**8. A Reputation-Based Trust Model for Peer-to-Peer e Commerce Communities**

**P . Melland T. Grance, “A Reputation-Based Trust Model for Peer-to-Peer e Commerce Communities ”Sep 2011, accessed: 05/06/2012, A available at:** [**http://csrc.nist.gov/publications/drafts/800-145/Draft-SP-800-145cloud-definition.pdf**](http://csrc.nist.gov/publications/drafts/800-145/Draft-SP-800-145cloud-definition.pdf)

Peer-to-Peer e Commerce communities are commonly perceived as an environment offering both opportunities and threats. One way to minimize threats in such an open community is to use community-based reputations to help evaluating the trustworthiness and predicting the future behaviour of peers. This paper presents Peer Trust a coherent adaptive trust model for quantifying and comparing the trustworthiness of peers based on a transaction-based feedback system. There are two main features of our model. First, we introduce three basic trust parameters in computing trustworthiness of peers. In addition to feedback a peer receives through its transactions with other peers, we incorporate the total number of transactions a peer performs, and the credibility of the feedback sources into the model for evaluating the trustworthiness of peers. We argue that the trust models based solely on feedback from other peers in the community is inaccurate and ineffective. Second, we introduce two adaptive trust factors, the transaction context factor and the community context factor, to allow the basic trust metric to incorporate different contexts (situations) and to address common problems encountered in a variety of online e Commerce communities. We present a concrete method to validate the proposed trust model and report the set of simulation-based experiments, showing the feasibility and benefit of the Peer Trust model.

Peer-to-peer (P2P) electronic commerce (e Commerce) communities can be seen as truly distributed computing applications in which peers (members) communicate directly with one another to exchange information, distribute tasks, or execute transactions. Such communities can be implemented either on top of a P2P network or using a conventional client-server platform. Gnutella is an example of P2P e Commerce communities that are built on top of a P2P computing platform. Person-to-person online auction sites such as eBay and many business-to-business (B2B) services such as supply-chain-management networks are examples of P2P communities built on top of a client server computing architecture. In e Commerce settings P2P communities are often established dynamically with peers that are unrelated and unknown to each other. Peers of such communities have to manage the risk involved with the transactions without prior experience and knowledge about each other’s reputation. One way to address this uncertainty is to develop strategies for establishing community-based trust through reputations. In a buyer-seller market, buyers are vulnerable to risks because of potential incomplete or distorted information provided by sellers. Trust is critical in such electronic markets as it can provide buyers with high expectations of satisfying exchange relationships. A recent study reported results from both an online experiment and an online auction market, which confirmed that trust can mitigate information asymmetry (the difference between the amounts of information the two transacting parties possess) by reducing transaction-specific risks, therefore generating price premiums for reputable sellers. Recognizing the importance of trust in such communities, an immediate question to ask is how to build trust. There is an extensive amount of research focused on building trust for electronic markets through trusted third parties or intermediaries. However, it is not applicable to P2P e Commerce communities where peers are equal in their roles and are independent entities, thus no peers can serve as trusted third parties or intermediaries. Reputation systems provide a way for building trust through social control without trusted third parties.

Most research on reputation-based trust utilizes information such as community-based feedbacks about past experiences of peers to help making recommendation and judgment on quality and reliability of the transactions. Community based feedbacks are often simple aggregations of positive and negative feedbacks that peers have received for the transactions they have performed and cannot accurately capture the trustworthiness of peers. In addition, peers can misbehave in a number of ways, such as providing false feedbacks on other peers. The challenge of building a trust mechanism is how to effectively cope with such malicious behaviour of peers. Another challenge is that trust context varies from transactions to transactions and from communities to communities. It is important to build a reputation based system that is able to adapt to different communities and different situations. Furthermore, there is also a need for experimental evaluation methods of a given trust model in terms of the effectiveness and benefits. Most traditional trust models only give an analytical model without any experimental validation due to the subjective nature of trust. There is a need of general metrics for evaluating the effectiveness and benefits of trust mechanisms.

**9. DEPSKY: Dependable and Secure Storage in a Cloud-of-Clouds**

**A.Bessani, M.Correia, B.Quaresma, F.Andre, and P .Sousa. DEPSKY: Dependable and Secure Storage in a Cloud-of-Clouds. In Proc.of ACM Euro Sys, 2011**

The increasing popularity of cloud storage services has lead companies that handle critical data to think about using these services for their storage needs. Medical record databases, power system historical information and financial data are some examples of critical data that could be moved to the cloud. However, the reliability and security of data stored in the cloud still remain major concerns. In this paper we present DEPSKY, a system that improves the availability, integrity and confidentiality of information stored in the cloud through the encryption, encoding and replication of the data on diverse clouds that form a cloud-of-clouds. We deployed our system using four commercial clouds and used Planet Lab to run clients accessing the service from different countries. We observed that our protocols improved the perceived availability and, in most cases, the access latency when compared with cloud providers individually. Moreover, the monetary costs of using DEPSKY on this scenario are twice the cost of using a single cloud, which is optimal and seems to be a reasonable cost, given the benefits.

The increasing maturity of cloud computing technology is leading many organizations to migrate their IT infrastructure and/or adapting their IT solutions to operate completely or partially in the cloud. Even governments and companies that maintain critical infrastructures (e.g., healthcare, telcos) are adopting cloud computing as a way of reducing costs [Greer 2010]. Nevertheless, cloud computing has limitations related to security and privacy, which should be accounted for, especially in the context of critical applications. This paper presents DEPSKY, a dependable and secure storage system that leverages the benefits of cloud computing by using a combination of diverse commercial clouds to build a cloud-of-clouds. In other words, DEPSKY is a virtual storage cloud, which is accessed by its users by invoking operations in several individual clouds. More specifically, DEPSKY addresses four important limitations of cloud computing for data storage in the following way: Loss of availability: temporary partial unavailability of the Internet is a well-known phenomenon. When data is moved from inside of the company network to an external data center, it is inevitable that service unavailability will be experienced. The same problem can be caused by denial-of-service attacks, like the one that allegedly affected a service hosted in Amazon EC2 in 2009 [Metz 2009]. DEPSKY deals with this problem by exploiting replication and diversity to store the data on several clouds, thus allowing access to the data as long as a subset of them is reachable. Loss and corruption of data: there are several cases of cloud services losing or corrupting customer data. For example, in October 2009 a subsidiary of Microsoft, Danger Inc., lost the contacts, notes, photos, etc. of a large number of users of the Sidekick service. The data was recovered several days later, but the users of Ma.gnolia were not so lucky in February of the same year, when the company lost half a terabyte of data that it never managed to recover. DEPSKY deals with this problem using Byzantine fault-tolerant replication to store data on several cloud services, allowing data to be retrieved correctly even if some of the clouds corrupt or lose data Loss of privacy: the cloud provider has access to both the data stored in the cloud and metadata like access patterns. The provider may be trustworthy, but malicious insiders are a wide-spread security problem. This is an especial concern in applications that involve keeping private data like health records. An obvious solution is the customer encrypting the data before storing it, but if the data is accessed by distributed applications this involves running protocols for key distribution (processes in different machines need access to the cryptographic keys). DEPSKY employs a secret sharing scheme and erasure codes to avoid storing clear data in the clouds and to improve the storage efficiency, amortizing the replication factor on the cost of the solution.

**10. MapReduce : Simplified Data Processing on Large Clusters :**

**J. Dean and S. Ghemawat, “Map Reduce: Simplified Data Processing on Large Clusters, ”Proc. Sixth Symp. Operating System Design and Implementation (OSDI ’04),pp. 137-150, Dec. 2009.**

Map Reduce is a programming model and an associated implementation for processing and generating large data sets. Users specify a map function that processes a key/value pair to generate a set of intermediate key/value pairs, and a reduce function that merges all intermediate values associated with the same intermediate key. Many real world tasks are expressible in this model, as shown in the paper. Programs written in this functional style are automatically parallelized and executed on a large cluster of commodity machines. The run-time system takes care of the details of partitioning the input data, scheduling the program’s execution across a set of machines, handling machine failures, and managing the required inter-machine communication. This allows programmers without any experience with parallel and distributed systems to easily utilize the resources of a large distributed system. Our implementation of Map Reduce runs on a large cluster of commodity machines and is highly scalable: a typical Map Reduce computation processes many terabytes of data on thousands of machines. Programmers find the system easy to use: hundreds of Map Reduce programs have been implemented and upwards of one thousand Map Reduce jobs are executed on Google’s clusters every day.

**3. SYSTEM STUDY**

**3.1 EXISTING SYSTEM**

* Trust management – Customer feedback based
* Trust feedbacks, trustworthiness – The trust of the customer and to believe or not
* Multi-Identity Recognition – Finding a user’s trustworthiness

As per existing the cloud armor deals with customer feedback in a public cloud. According to the activity of the users, the trustworthiness of the user or customer will be evaluated. The non trustworthy user or customer will be eliminated from the network. The following methods are the key points used in the existing system.

* A protocol to prove the credibility of trust feedbacks, and Preserve user privacy.
* An adaptive and robust credibility model for measuring the credibility of trust feedbacks to protect cloud services from malicious users and to compare the trustworthiness of cloud services.
* An availability model to manage the availability of the decentralized implementation of the trust management service. The feasibility and benefits of our approach have been validated by a prototype and experimental studies using a collection of real-world trust feedbacks on cloud services.

In the existing system the process was well started with user’s credibility of trust feedbacks. But the problem in the existing system is the user’s privacy is not completely established, because after the data publishing only the trust management will find out the credibility. But the result is, the data will be published in public. The published non trustworthy data may cause some wrong feed backs. In the existing system there is no special environment has been used.

In the second point, the adaptive and robust credibility model has been used for finding the malicious users. But literally there are many and more practical problems may occur in finding out the malicious users. In case of eliminating the real user instead of malicious user may cause more issues. By till data the success ratio of finding out the malicious is about 60 % to 70 % only, even in a secured network. So again trustworthy became a major issue in the existing system. Also sometimes original users may turn out to a malicious user for various reasons. These types of users can’t be find out.

And finally the existing system deals with decentralized implementation of the trust management service. In this paper, the overview design and the implementation of Cloud Armor (cloud consumer’s credibility Assessment and trust management of cloud services): a framework for reputation based trust management in cloud environments. In Cloud Armor, trust is delivered as a service (TaaS) where TMS spans several Distributed nodes to manage feedbacks in a decentralized way. But the problem in decentralization may cause more malicious users. The decentralization of the cloud may invite more users worldwide. So the vulnerability of the system may increase in trust as a service.

**3.1.1 Disadvantage of the existing system:**

* There is no specific application has been implemented for applying trust as a service.
* There are various drawbacks in finding out the malicious users.
* Identifying trust became more tedious in the existing system.
* No rules and algorithm has been implemented for identifying the trust users.
* No preservation of data has been implemented.
* These TMS nodes expose interfaces so that users can give their feedback or inquire the trust results in a decentralized way. But as mentioned above decentralize may minimize the trust ratio.
* The Only way to preserve privacy is to use cryptographic encryption techniques
* Also known as collusive malicious feedback behaviors, such attacks occur when several vicious users collaborate together to give numerous misleading feedbacks to increase the trust result of failures

**3.2 PROPOSED SYSTEM:**

**Techniques Used:**

ESMTP – Enhanced Simple Mail Transfer

Trust as a Service – User Rights

Aware as a Service – IDS (Intrusion Detection System)

IP Synchronization

All the drawback in the existing has been over came in the proposed system. In the proposed system an organization environment has been used for implementing both trust and aware framework. Here SMTP server has been enhanced as ESMTP for implement Trust as a service and IDS has been implemented for Aware as a service. The Enhanced Simple Mail Transfer Protocol (ESMTP) service provided by IIS(Internet Information Service) is a component for delivering outgoing e-mail messages for a DNS(Domain Name server) based SMTP. Delivery of a message is initiated by transferring the message to a designated ESMTP server. Based on the domain name of the recipient e-mail address, the SMTP server initiates communications with a Domain Name System (DNS) server, which looks up and then returns the host name of the destination SMTP server for that domain. The originating SMTP server communicates with the destination SMTP server directly through Transmission Control Protocol/Internet Protocol (TCP/IP) on port 25. Each port can make up to 1 lakh communication.

The ESMTP is a flexible mail server which can be implemented in various environments like IT Organizations, Private Organization, Government sectors and etc. And ESTMP can be customized more up to the user favor. This protocol supports multiple file types, various security levels, multiple organization and etc. All these techniques come under a single architecture. As per existing the cloud armor deals with customer feedback in a public cloud. According to the activity of the users, the trustworthiness of the user or customer will be evaluated.

Here the trust as a service has been modified according to trust routing framework. The ESMTP has been customized according to the admin procedure. So that admin can provide user rights to the employee by customizing the mail components like compose, inbox, send items and etc. The customized mail server will reflect to the users login and mail will response for unidirectional. For Aware routing framework, intrusion detection has been implemented. The IDS will capture all the intruders’ details like data, time, IP address, Password used, Password matching percentage and times of hacking. This makes the admin more from the intruder and hacker side. And finally the color code has been implemented for secured login. For color code quantum cryptography has been used. A color grid is available with various colors. User can select any three colors for secured login. In added with Gaussian mixture and keystroke has been implemented for pattern based password login.

**3.2.1 Advantages of the proposed system**

* A private mail server will be created for the company, So that from the mail server each user will be provided with the individual DNS.
* More trust has been implemented.
* A Prior user rights will be provided for the users. This control will be made by the admin.
* Admin can customize the mail component of the users like compose, inbox, outbox, send items and etc.
* In case of change of password by the user, a notification will be sent to the admin. So that admin can able to view the changed username and password.
* Intruder will be identified using instruction detection technique. In case of hacking the user’s hacking date, time, password used and ip address of the hacker will be recorded in the data grid.
* The intruder detection method will be applicable for both admin and user.
* Colour code and Gaussian has been implemented.

**4. SYSTEM SPECIFICATION**

**4.1 Application Specification**

**Client/Server Architecture:**

Benefits offered by client/server architecture:

* Increased user communication because of flexible data access.
* Highly interactive user interface.
* Increased developer productivity through usage of easy to use easy tools.
* Improved access to information because of networking.
* Better control of corporate data through centralized data, systems & network management.
* Easier maintenance of application & data.
* Protection of hardware investments by making use of existing installations of Hardware, software & network and at same time getting maximum leverage out of the available

**4.2 Network Specification**

This thesis is purely developed for multi user system and presently we are using the following network specification.

(i) Windows 7 Platform

Windows 7 is a powerful multitasking operating system with high security. It is user friendly and supports multithreading and lot of tools for developing any application. This OS has number of enhancements, including performance improvements, better hardware support and closer integration with the Net. Windows support dynamic linking. This OS has the concept of plug and play.

(ii) IIS -Application Server

IIS is the Internet Information Server. The thesis is a web- based thesis. It needs an application server to run. IIS is an application server where the thesis runs. This application server is chosen because the thesis is developed in ASP and both of them are Microsoft products. Performance will be good if the product is form the same company. IIS is user-friendlier than other application servers. Some of its features are,

* High performance network and application server.
* The server includes the Secure Sockets Layer (SSL) encrypted communication standard for private communication between the clients and server.
* Active Server page allows application with scripts and components to perform multiple actions.
* With Windows NT service pack. It also acts as a web server.

(iii) **Personal Web Server**

Personal Web Server (PWS) is the server designed for developing sites off

line. It supports many of the features seen in the full version of IIS including virtual directories and Active Server Pages. Server administration is straightforward and simple-much like administration of its larger counter part. IIS administration, which is dialog-based while PWS administration is web based. Personal Web Server properties dialog box reveals the two services: HTTP and FTP. It does not provide the Gopher service available in IIS.

**4.3 Hardware Specification:**

PROCESSOR : Intel Pentium IV 1.8 GHz

MOTHERBOARD : Intel 915GVSR chipset board

RAM : 1 GB DDR2 RAM

HARD DISK DRIVE : 160 GB

DVD/CD DRIVE : Sony 52 x Dual layer drive

MONITOR : 17” Color TFT Monitor

KEYBOARD : Multimedia Keyboard 108 Keys

MOUSE : Logitech Optical Mouse

CABINET : ATX iball.

HUB : Compex 16 lines.

BANDWIDTH : 100 mbps.

**4.4 SOFTWARE SPECIFICATION:**

FRONTEND : ASP.NET 2012

BACKEND : SQL Server 2010

CODING LANGUAGE : C#

OPERATING SYSTEMS : Microsoft windows xp

DOCUMENTATION : Microsoft word 2003.

SCRIPTING LANGUAGE : Java Script

**5. TRUST AWARE ROUTING PROCEDURE**

**5.1 ABOUT ROUTING PROCEDURE**

Security is a critical issue in a mobile ad hoc network (MANET). In most of the previous protocols security is an added layer above the routing protocol. We propose a Trust Aware Routing Protocol (TARP) for secure-trusted routing in mobile ad hoc networks. In TARP, security is inherently built into the routing protocol where each node evaluates the trust level of its neighbors based on a set of attributes and determines the route based on these attributes.

This paper evaluates the proposed TARP protocols on two important attributes, the battery power and the software configuration. A secure route between a source and destination is established based on a confidence level prescribed by a user or application in terms of these attributes. Our performance evaluation shows that TARP is a robust and adaptive trust routing algorithm that reacts quickly and effectively to the dynamics of the network while still finding the shortest path to the destination. TARP is able to improve security and at the same time reduce the total routing traffic sent and received in the network by directing the traffic based on the requested sender attributes

**5.2 TRUST AWARE ROUTING PROTOCOL**

TARP selects routes to the destination based not only on the shortest path but also on several other security oriented attributes of the nodes. Only nodes that match the sender requirements would forward the packet.

The main objectives of the proposed TARP suite are:

(a) Implement security that is inherently built into the routing protocol,

(b) Deliver messages that are received with a user defined or best available level of confidence,

(c) Allow users and applications to prescribe their required level of security,

(d) Achieve efficiency in routing that is improved by limiting control message exchanges, (e) optimize resource usage,

(f) Obtain graceful network performance degradation, and (g) develop a protocol suite that adapts to changes in the environment, such as the network topology, the power-level of nodes, etc.

In TARP, the security parameters considered in computing the trust-level of a node in a given route include: software configuration, hardware configuration, battery power, credit history, exposure and organizational hierarchy. Each node evaluates the trust level of its neighbors based on the above parameters and includes it in computing the next hop node in the overall shortest route computation. Due to page limitations, this

paper will focus on the implementation and evaluation of the battery power and the software configuration attributes. Below is a description of the battery power and software configuration attributes:

• **Power**: In wireless networks, the battery power with which nodes operate is a limited resource. Each node uses its power to not only send and receive, it also behaves as a router by forwarding routing messages and updates. The cryptographic techniques that provide security are computationally intensive, which further increase the power consumption of a node. The node's trust level should be set to low since it cannot guarantee its service. This illustrates that power is an important parameter for evaluating the trust level of a node.

• **Software Configuration**: The software configuration includes the encryption ability of a node. To satisfy CAI (Confidentiality, Availability and Integrity), different cryptographic mechanisms have been proposed. Some are based on symmetric encryption and others on asymmetric encryption. Each node is given either a shared secret key or a public/private key pair depending on the type of cryptographic mechanism. Different encryption algorithms are available such as RSA, DES/3DES, BLOWFISH, IDEA, SEAL RC2/RC4/RC5/RC6 [12]. Strong encryption is often discerned by the key length used by the algorithm. In general, a node with a stronger encryption algorithm has a higher trust level than a node with a weaker encryption algorithm.

**5.3 ABOUT SMTP**

The Simple Mail Transfer Protocol (SMTP) service provided by IIS is a simple component for delivering outgoing e-mail messages. Delivery of a message is initiated by transferring the message to a designated SMTP server. Based on the domain name of the recipient e-mail address, the SMTP server initiates communications with a Domain Name System (DNS) server, which looks up and then returns the host name of the destination SMTP server for that domain.

Next, the originating SMTP server communicates with the destination SMTP server directly through Transmission Control Protocol/Internet Protocol (TCP/IP) on port 25. If the user name of the recipient e-mail address matches one of the authorized user accounts on the destination server, the original e-mail message is transferred to that server, waiting for the recipient to pick up the message through a client program.

In the case where the originating SMTP server cannot communicate directly with the destination server, the SMTP service can transfer messages through one or more intermediate relay SMTP servers. A relay server receives the original message and then delivers it to the destination server, or redirects it to another relay server. This process is repeated until the message is delivered or a designated timeout period passes.

The SMTP service is not installed by default. You must install the SMTP service using the Control Panel. Installing the SMTP service creates a default SMTP configuration which you can then customize to your needs using IIS Manager.

**5.3.1 SMTP vs mail retrieval**

SMTP is a delivery protocol only. It cannot pull messages from a remote server on demand. Other protocols, such as the Post Office Protocol (POP) and the Internet Message Access Protocol (IMAP) are specifically designed for retrieving messages and managing mail boxes. However, SMTP has a feature to initiate mail queue processing on a remote server so that the requesting system may receive any messages destined for it (see Remote Message Queue Starting below). POP and IMAP are preferred protocols when a user's personal computer is only intermittently powered up, or Internet connectivity is only transient and hosts cannot receive message during off-line periods. Remote Message Queue Starting is a feature of SMTP that permits a remote host to start processing of the mail queue on a server so it may receive messages destined to it by sending the TURN command. This feature however was deemed insecure[15] and was extended in RFC 1985 with the ETRN command which operates more securely using an authentication method based on Domain Name System information.

**5.3.2** **Outgoing mail SMTP server**

An e-mail client needs to know the IP address of an SMTP server and this has to be given as part of its configuration (usually given as a DNS name). The server will deliver outgoing messages on behalf of the user.

**5.3.3 Outgoing mail server access restrictions**

Server administrators need to impose some control on which clients can use the server. This enables them to deal with abuse, for example spam. Two solutions have been in common use:

In the past, many systems imposed usage restrictions by the location of the client, only permitting usage by clients whose IP address is one that the server administrators control. Usage from any other client IP address is disallowed.

Modern SMTP servers typically offer an alternative system that requires authentication of clients by credentials before allowing access.

**5.3.4 Restricting access by location**

Under this system, an ISP's SMTP server will not allow access by users who are 'outside the ISP's network'. More precisely, the server may only allow access to users with an IP address provided by the ISP, which is equivalent to requiring that they are connected to the Internet using that same ISP. A mobile user may often be on a network other than that of their normal ISP, and will then find that sending email fails because the configured SMTP server choice is no longer accessible.

This system has several variations. For example, an organisation's SMTP server may only provide service to users on the same network, enforcing this by firewalling to block access by users on the wider Internet. Or the server may perform range checks on the client's IP address. These methods were typically used by corporations and institutions such as universities which provided an SMTP server for outbound mail only for use internally within the organisation. However, most of these bodies now use client authentication methods, as described below.

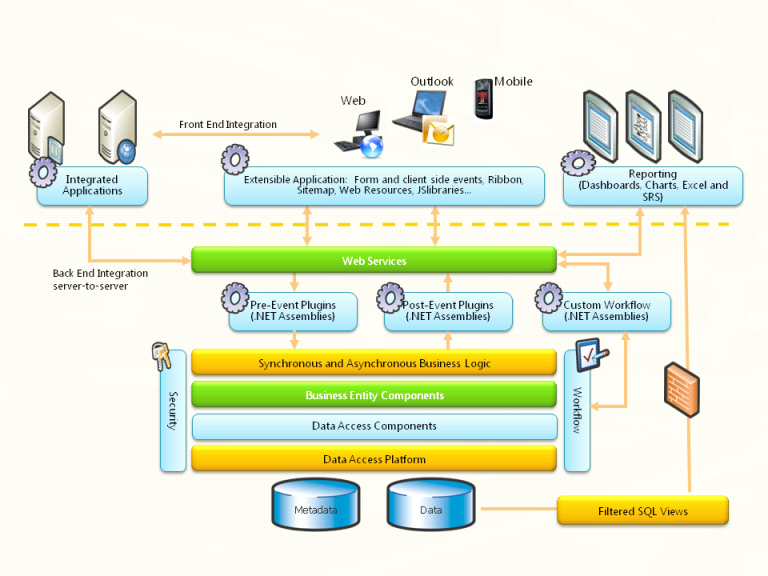
By restricting access to certain IP addresses, server administrators can readily recognise the IP address of any abuser. As it will be a meaningful address to them, the administrators can deal with the rogue machine or user.

Where a user is mobile, and may use different ISPs to connect to the internet, this kind of usage restriction is onerous, and altering the configured outbound email SMTP server address is impractical. It is highly desirable to be able to use email client configuration information that does not need to change.

**5.3.5 Client authentication**

Modern SMTP servers typically require authentication of clients by credentials before allowing access, rather than restricting access by location as described earlier. This more flexible system is friendly to mobile users and allows them to have a fixed choice of configured outbound SMTP server.

**5.4 MAIL PROCESSING MODEL**



Email is submitted by a mail client (MUA, mail user agent) to a mail server (MSA, mail submission agent) using SMTP onTCP port 587. Most mailbox providers still allow submission on traditional port 25. From there, the MSA delivers the mail to its mail transfer agent (MTA, mail transfer agent). Often, these two agents are just different instances of the same software launched with different options on the same machine. Local processing can be done either on a single machine, or split among various appliances; in the former case, involved processes can share files; in the latter case, SMTP is used to transfer the message internally, with each host configured to use the next appliance as a smart host. Each process is an MTA in its own right; that is, an SMTP server.

The boundary MTA has to locate the target host. It uses the Domain name system (DNS) to look up the mail exchanger record (MX record) for the recipient's domain (the part of the address on the right of @). The returned MX record contains the name of the target host. The MTA next connects to the exchange server as an SMTP client. (The article on MX record discusses many factors in determining which server the sending MTA connects to.)

Once the MX target accepts the incoming message, it hands it to a mail delivery agent (MDA) for local mail delivery. An MDA is able to save messages in the relevant mailbox format. Again, mail reception can be done using many computers or just one —the picture displays two nearby boxes in either case. An MDA may deliver messages directly to storage, or forward them over a network using SMTP, or any other means, including the Local Mail Transfer Protocol (LMTP), a derivative of SMTP designed for this purpose.

Once delivered to the local mail server, the mail is stored for batch retrieval by authenticated mail clients (MUAs). Mail is retrieved by end-user applications, called email clients, using Internet Message Access Protocol (IMAP), a protocol that both facilitates access to mail and manages stored mail, or the Post Office Protocol (POP) which typically uses the traditional mbox mail file format or a proprietary system such as Microsoft Exchange/Outlook or Lotus Notes/Domino. Webmail clients may use either method, but the retrieval protocol is often not a formal standard.

SMTP defines message transport, not the message content. Thus, it defines the mail envelope and its parameters, such as the envelope sender, but not the header or the body of the message itself. STD 10 and RFC 5321 define SMTP (the envelope), while STD 11 and RFC 5322 define the message (header and body), formally referred to as the Internet Message Format.

**6. PROTOCOL OVERVIEW**

SMTP is a connection-oriented, text-based protocol in which a mail sender communicates with a mail receiver by issuing command strings and supplying necessary data over a reliable ordered data stream channel, typically a Transmission Control Protocol (TCP) connection. An SMTP session consists of commands originated by an SMTP client (the initiating agent, sender, or transmitter) and corresponding responses from the SMTP server (the listening agent, or receiver) so that the session is opened, and session parameters are exchanged. A session may include zero or more SMTP transactions. An SMTP transaction consists of three command/reply sequences (see example below.) They are:

MAIL command, to establish the return address, a.k.a. Return-Path, 5321.From, mfrom, or envelope sender. This is the address for bounce messages.

**RCPT** command, to establish a recipient of this message. This command can be issued multiple times, one for each recipient. These addresses are also part of the envelope.

**DATA** to send the message text. This is the content of the message, as opposed to its envelope. It consists of a message header and a message body separated by an empty line.

DATA is actually a group of commands, and the server replies twice: once to the DATA command proper, to acknowledge that it is ready to receive the text, and the second time after the end-of-data sequence, to either accept or reject the entire message.

Besides the intermediate reply for DATA, each server's reply can be either positive (2xx reply codes) or negative. Negative replies can be permanent (5xx codes) or transient (4xx codes). A reject is a permanent failure by an SMTP server; in this case the SMTP client should send a bounce message. A drop is a positive response followed by message discard rather than delivery.

The initiating host, the SMTP client, can be either an end-user's email client, functionally identified as a mail user agent (MUA), or a relay server's mail transfer agent (MTA), that is an SMTP server acting as an SMTP client, in the relevant session, in order to relay mail. Fully capable SMTP servers maintain queues of messages for retrying message transmissions that resulted in transient failures.

A MUA knows the outgoing mail SMTP server from its configuration. An SMTP server acting as client, i.e. relaying, typically determines which SMTP server to connect to by looking up the MX (Mail eXchange) DNS resource record for each recipient's domain name. Conformant MTAs (not all) fall back to a simple A record in case no MX record can be found. Relaying servers can also be configured to use a smart host.

An SMTP server acting as client initiates a TCP connection to the server on the "well-known port" designated for SMTP: port 25. MUAs should use port 587 to connect to an MSA. The main difference between an MTA and an MSA is that SMTP Authentication is mandatory for the latter only.

**6.1 ALGORITHM IMPLEMENTATION**

Assume that a hash function selects each array position with equal probability. If *m* is the number of bits in the array, and *k* is the number of hash functions, then the probability that a certain bit is not set to 1 by a certain hash function during the insertion of an element is then



The probability that it is not set to 1 by any of the hash functions is



If we have inserted *n* elements, the probability that a certain bit is still 0 is



the probability that it is 1 is therefore



Now test membership of an element that is not in the set. Each of the *k* array positions computed by the hash functions is 1 with a probability as above. The probability of all of them being 1, which would cause the algorithm to erroneously claim that the element is in the set, is often given as



This is not strictly correct as it assumes independence for the probabilities of each bit being set. However, assuming it is a close approximation we have that the probability of false positives decreases as *m* (the number of bits in the array) increases, and increases as *n* (number of inserted elements) increases. For a given *m* and *n*, the value of *k* (the number of hash functions) that minimizes the probability is



which gives the false positive probability of



The required number of bits *m*, given *n* (the number of inserted elements) and a desired false positive probability *p* (and assuming the optimal value of *k* is used) can be computed by substituting the optimal value of *k* in the probability expression above:





This results in:



This means that for a given false positive probability *p*, the length of a Bloom filter *m* is proportionate to the number of elements being filtered *n*.[2] While the above formula is asymptotic (i.e. applicable as *m*,*n* → ∞), the agreement with finite values of *m*,*n* is also quite good; the false positive probability for a finite bloom filter with *m* bits, *n* elements, and *k*hash functions is at most



So we can use the asymptotic formula if we pay a penalty for at most half an extra element and at most one fewer bit.

**7.CONCLUSION**

This project has been implemented successfully according to the committed abstract and all outputs have been verified. All the outputs are generating according to the given input. Data validations are done according to the user and admin input data. The employee’s user name and password are generated in admin login, all the login has been verified successfully.

Trust routing and aware routing framework has been implemented successfully and result has been verified. Both routing frameworks are working according to the expected level. ‘TaaS’ working well for the 3 types of synchronization methods. And finally untrusted users can be find out easily using the above mention methods. So dual level security has been provided to the centralized server

Thus cloud armour has been implemented successfully and in efficient manner.

**8.Reference:**

[http://www.asp.net](http://www.asp.net/)

[http://www.osborne.com](http://www.osborne.com/)

[http://www.microsoft.com](http://www.microsoft.com/)

[1] A. Wood and J. Stankovic, “Denial of service in sensor networks,”Computer, vol. 35, no. 10, pp. 54–62, Oct 2002.

[2] M. Jain and H. Kandwal, “A survey on complex wormhole attackin wireless ad hoc networks,” in Proceedings of International Con-ference on Advances in Computing, Control, and Telecommunication Technologies (ACT ’09), 28-29 2009, pp. 555 –558.

[3] I. Krontiris, T. Giannetsos, and T. Dimitriou, “Launching a sink-hole attack in wireless sensor networks; the intruder side,” in Proceedings of IEEE International Conference on Wireless and Mobile Computing, Networking and Communications(WIMOB ’08), 12-14 2008, pp. 526 –531.

[4] L. Bai, F. Ferrese, K. Ploskina, and S. Biswas, “Performance analy-sis of mobile agent-based wireless sensor network,” in Proceedings of the 8th International Conference on Reliability, Maintainability and Safety (ICRMS 2009), 20-24 2009, pp. 16 –19.

[5] A. Perrig, R. Szewczyk, W. Wen, D. Culler, and J. Tygar, “SPINS: Security protocols for sensor networks,” Wireless Networks Journal (WINET), vol. 8, no. 5, pp. 521–534, Sep. 2002.

[6] R. Watro, D. Kong, S. Cuti, C. Gardiner, C. Lynn, and P. Kruus, “Tinypk: securing sensor networks with public key technology,” in Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks (SASN ’04). New York, NY, USA: ACM, 2004, pp. 59–64.

[7] A. Liu and P. Ning, “Tinyecc: A configurable library for elliptic curve cryptography in wireless sensor networks,” in Proceedings of the 7th international conference on Information processing in sensor networks (IPSN ’08). IEEE Computer Society, 2008, pp. 245–256.

[8] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” IEEE Communications Magazine, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[21] J. L. X. Li, M. R. Lyu, “Taodv: A trusted aodv routing protocol for mobile ad hoc networks,” in Proceedings of Aerospace Conference, 2004.

[9] T. Zahariadis, H. Leligou, P. Karkazis, P. Trakadas, I. Papaefs-tathiou, C. Vangelatos, and L. Besson, “Design and implementa-tion of a trust-aware routing protocol for large wsns,” International Journal of Network Security & Its Applications (IJNSA), vol. 2, no. 3,Jul. 2010.

[10] A. Rezgui and M. Eltoweissy, “Tarp: A trust-aware routing protocol for sensor-actuator networks,” in IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems (MASS 2007), 8-11 2007.

[11] S. Chang, S. Shieh, W. Lin, and C. Hsieh, “An efficient broadcast authentication scheme in wireless sensor networks,” in Proceedings of the 2006 ACM Symposium on Information, computer and communications security (ASIACCS ’06). New York, NY, USA: ACM, 2006, pp. 311–320.

[12] S. Ganeriwal, L. Balzano, and M. Srivastava, “Reputation-based framework for high integrity sensor networks,” ACM Trans. Sen. Netw., 2008.

[13] G. Zhan, W. Shi, and J. Deng, “Poster abstract: Sensortrust - a re-silient trust model for wsns,” in Proceedings of the 7th International Conference on Embedded Networked Sensor Systems (SenSys’09), 2009.

[14] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, “Collection tree protocol,” in Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys ’09). New York, NY, USA: ACM, 2009, pp. 1–14.

[15] G. Zhan, W. Shi, and J. Deng, “Design, implementation and evaluation of tarf: A trust-aware routing framework for dynamic wsns,” <http://mine.cs.wayne.edu/guoxing/tarf.pdf>, Wayne State University, Tech. Rep. MISTTR2010-003, Oct. 2010.

[16] X. Hong, M. Gerla, G. Pei, and C. Chiang, “A group mobility model for ad hoc wireless networks,” in Proceedings of the 2nd ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM ’99). New York, NY, USA: ACM, 1999, pp. 53–60.

[17] “Motelab,” http://motelab.eecs.harvard.edu, 2005.

[18] Z. Cao, J. Hu, Z. Chen, M. Xu, and X. Zhou, “Fbsr: feedback-based secure routing protocol for wireless sensor networks,” International Journal of Pervasive Computing and Communications, 2008.

[19] T. Ghosh, N. Pissinou, and K. Makki, “Collaborative trust-based secure routing against colluding malicious nodes in multi-hop adhoc networks,” in Local Computer Networks, 2004. 29th Annual IEEE International Conference on, Nov. 2004, pp. 224 – 231.