RESEARCH ARTICLE OPEN ACCESS

Cloud Information Sharing And Accountability Using Logging And Auditing Techniques

Umamaheswara Rao[1], B.Srinivasulu[2], R.V.Gandhi [3], B.Venkata Ramana [4]
Department of Computer Science & Engineering, St.Martin's Engineering College

ABSTRACT
Cloud computing enables highly scalable services to be easily consumed over the Internet on an as-needed basis. A major feature of the cloud services is that users' data are usually processed remotely in unknown machines that users do not own or operate. While enjoying the convenience brought by this new emerging technology, users' fears of losing control of their own data (particularly, financial and health data) can become a significant barrier to the wide adoption of cloud services. To address this problem, here, we propose a novel highly decentralized information accountability framework to keep track of the actual usage of the users' data in the cloud. In particular, we propose an object-centered approach that enables enclosing our logging mechanism together with users' data and policies. We leverage the JAR programmable capabilities to both create a dynamic and traveling object, and to ensure that any access to users' data will trigger authentication and automated logging local to the JARs. To strengthen user's control, we also provide distributed auditing mechanisms. We provide extensive experimental studies that demonstrate the efficiency and effectiveness of the proposed approaches.
Key Terms: Cloud Computing, Information Sharing, logging, audit ability, accountability, data sharing, secure JVM.

I. INTRODUCTION

Cloud computing presents a new way to supplement the current consumption and delivery model for IT services based on the Internet, by providing for dynamically scalable and often virtualized resources as a service over the Internet. To date, there are a number of notable commercial and individual cloud computing services, including Amazon, Google, Microsoft, Yahoo, and Sales force. Details of the services provided are abstracted from the users who no longer need to be experts of technology infrastructure. Moreover, users may not know the machines which actually process and host their data. While enjoying the convenience brought by this new technology, users also start worrying about losing control of their own data. The data processed on clouds are often outsourced, leading to a number of issues related to accountability, including the handling of personally identifiable information. Such fears are becoming a significant barrier to the wide adoption of cloud services. There is a lot of advancement takes place n the system with respect to the internet as a major concern in its implementation in a well effective manner respectively. There is an advancement of the internet is termed as the computation of the cloud in a well oriented fashion respectively. Many of the users are getting attracted to this particular technology due to the services involved in it followed by the reduced computation followed by the cost and also the reliable data transmission takes place in the system in a well effective manner respectively. In this paper a method is designed with a well effective strategy oriented framework in a well effective manner used for the implementation of the system in terms of the performance based strategy followed by the accurate analysis with respect to the system respectively. Here the present designed method is shown in the below figure in the form of the block diagram and explains in the elaborative fashion respectively. Here the present method completely overcomes the drawbacks of the several previous methods in a well efficient manner that is in terms of the performance followed by the accurate analysis respectively. There is a huge challenge for the present method where it is implemented by the effective analysis and there is an accurate analysis of the system in a well effective manner followed by the performance based evaluation in a well oriented aspect respectively. Cloud computing presents a new way to supplement the current consumption and delivery model for IT services based on the Internet, by providing for dynamically scalable and often virtualized resources as a service over the Internet. To date, there are a number of notable commercial and individual cloud computing services, including Amazon, Google, Microsoft, Yahoo, and Sales force. Details of the services provided are abstracted from the users who no longer need to be experts of technology infrastructure. Moreover, users may not know the machines which actually process and host their data. While enjoying the convenience brought by this new technology, users also start worrying about losing control of their own data. The data processed on

clouds are often outsourced, leading to a number of issues related to accountability, including the handling of personally identifiable information. Such fears are becoming a significant barrier to the wide adoption of cloud services. To allay users' concerns, it is essential to provide an effective mechanism for users to monitor the usage of their data in the cloud. For example, users need to be able to ensure that their data are handled according to the service level agreements made at the time they sign on for services in the cloud. Conventional access control approaches developed for closed domains such as databases and operating systems, or approaches using a centralized server in distributed environments, are not suitable, due to the following features characterizing cloud environments. First, data handling can be outsourced by the direct cloud service provider (CSP) to other entities in the cloud and theses entities can also delegate the tasks to others, and so on. Second, entities are allowed to join and leave the cloud in a flexible manner. As a result, data handling in the cloud goes through a complex and dynamic hierarchical service chain which does not exist in conventional environments. To overcome the above problems, we propose a novel approach, namely Cloud Information Accountability (CIA) framework, based on the notion of information accountability. Unlike privacy protection technologies which are built on the hide-it-or-lose-it perspective, information accountability focuses on keeping the data usage transparent and tractable. Our proposed CIA framework provides end-toned accountability in a highly distributed fashion. One of the main innovative features of the CIA framework lies in its ability of maintaining lightweight and powerful accountability that combines aspects of access control, usage control and authentication. By means of the CIA, data owners can track not only whether or not the service-level agreements are being honored, but also enforce access and usage control rules as needed. Associated with the accountability feature, we also develop two distinct modes for auditing: push mode and pull mode. The push mode refers to logs being periodically sent to the data owner or stakeholder while the pull mode refers to an alternative approach whereby the user (or another authorized party) can retrieve the logs as needed.

II. SYSTEM ANALYSIS

Existing System:

To allay users' concerns, it is essential to provide an effective mechanism for users to monitor the usage of their data in the cloud. For example, users need to be able to ensure that their data are handled according to the service level agreements made at the time they sign on for services in the cloud. Conventional access control approaches developed for closed domains such as databases and operating systems, or approaches using a centralized server in distributed environments, are not suitable, due to the following features characterizing cloud environments.

Problems on existing system:

First, data handling can be outsourced by the direct cloud service provider (CSP) to other entities in the cloud and theses entities can also delegate the tasks to others, and so on. Second, entities are allowed to join and leave the cloud in a flexible manner. As a result, data handling in the cloud goes through a complex and dynamic hierarchical service chain which does not exist in conventional environments.

Proposed System:

We propose a novel approach, namely Cloud Information Accountability (CIA) framework, based on the notion of information accountability. Unlike privacy protection technologies which are built on the hide-it-or-lose-it perspective, information accountability focuses on keeping the data usage transparent and tractable. Our proposed CIA framework provides end-toend accountability in a highly distributed fashion. One of the main innovative features of the CIA framework lies in its ability of maintaining lightweight and powerful accountability that combines aspects of access control, usage control and authentication. By means of the CIA, data owners can track not only whether or not the service-level agreements are being honored, but also enforce access and usage control rules as needed. Associated with the accountability feature, we also develop two distinct modes for auditing: push mode and pull mode. The push mode refers to logs being periodically sent to the data owner or stakeholder while the pull mode refers to an alternative approach whereby the user (or another authorized party) can retrieve the logs as needed.

Our main contributions are as follows:

- We propose a novel automatic and enforceable logging mechanism in the cloud.
- Our proposed architecture is platform independent and highly decentralized, in that it does not require any dedicated authentication or storage system in place.
- We go beyond traditional access control in that we provide a certain degree of usage control for the protected data after these are delivered to the receiver.
- We conduct experiments on a real cloud tested. The results demonstrate the efficiency, scalability, and granularity of our approach. We also provide a detailed security analysis and discuss the reliability and strength of our architecture.

Methodology:

In this paper a method is designed with a well effective strategy oriented framework in a well effective manner used for the implementation of the system in terms of the performance based strategy followed by the accurate analysis with respect to the system respectively [1][4]. Here the present

designed method is shown in the below figure in the form of the block diagram and explains in the elaborative fashion respectively [5][6]. Here the present method completely overcome the drawbacks of the several previous methods in a well efficient manner that is in terms of the performance followed by the accurate analysis respectively [7][8]. There is a huge challenge for the present method where it is implemented by the effective analysis and there is an accurate analysis of the system in a well effective manner followed by the performance based evaluation in a well oriented aspect respectively.

Expected Results:

A lot of analysis is made in the present method and a huge number of the computations have been applied on the large number of the data sets in a well oriented fashion respectively. A comparative analysis is made between the present method to that of the several previous methods in a well effective fashion and is shown in the below figure in the form of the graphical representation and explains in an elaborative fashion respectively. There is a huge challenge for the present method where it is supposed to overcome the drawbacks of the several previous methods followed by the accurate analysis of the system in a well oriented fashion respectively.

III. RELATED WORK

In this section, we first review related works addressing the privacy and security issues in the cloud. Survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy n company strength. Once these things are satisfied, then next steps are to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration are taken into account for developing the proposed system. The design of the CIA framework presents substantial challenges, including uniquely identifying CSPs, ensuring the reliability of the log, adapting to a highly centralized infrastructure, etc. Our basic approach toward addressing these issues is to leverage and extend the programmable capability of JAR (Java ARchives) files to automatically log the usage of the users' data by any entity in the cloud. Users will send their data along with any policies such as access control policies and logging policies that they want to enforce, enclosed in JAR files, to cloud service providers. Any access to the data will trigger an automated and authenticated logging mechanism local to the JARs. We refer to this type of enforcement as "strong binding" since the policies and the logging mechanism travel with the data. This strong binding exists even when copies of the JARs are created; thus, the user will have control over his data at any location. Such decentralized logging mechanism meets the dynamic nature of the cloud but also imposes challenges on ensuring the integrity of the logging. To cope with this issue, we provide the JARs with a central point of contact which forms a link between them and the user. It records the error correction information sent by the JARs, which allows it to monitor the loss of any logs from any of the JARs. Moreover, if a JAR is not able to contact its central point, any access to its enclosed data will be denied. Currently, we focus on image files since images represent a very common content type for end users and organizations (as is proven by the popularity of Flicker) and are increasingly hosted in the cloud as part of the storage services offered by the utility computing paradigm featured by cloud computing. Further, images often reveal social and personal habits of users, or are used for archiving important files from organizations. In addition, our approach can handle personal identifiable information provided they are stored as image files (they contain an image of any textual content, for example, the SSN stored as a .jpg file).

Remotely hosted: Services or data are hosted on remote infrastructure.

Ubiquitous: Services or data are available from anywhere.

Commodified: The result is a utility computing model similar to traditional that of traditional utilities, like gas and electricity - you pay for what you would want!

IaaS- Infrastructure as a Service:

The IaaS layer extends the virtualization layer by providing the mechanisms to provision and control the virtual machines in a utility computing manner. The end user is able to control cost by knowing exactly how much each virtual machine instance costs per minute/hour.

PaaS- Platform as a Service:

Seeks to minimize the hassle and complexity in deploying an application in the cloud. A programming platform is presented to the end user, typically a developer at this point, which leverages and API and programming language.

SaaS- Software as a Service:

Simple way to get the application functionality you need without incurring the cost of developing that application. In this layer, even the platform has been abstracted away from you as an end user

Basic Cloud Characteristics:

The "no-need-to-know" in terms of the underlying details of infrastructure, applications interface with the infrastructure via the APIs. The "flexibility and elasticity" allows these systems to scale up and down at will – utilizing the resources of all kinds (CPU, storage, server capacity, load balancing, and databases). The "pay as much as used and needed" type of utility computing and the "always on!,

anywhere and any place" type of network-based computing.

IV. PROBLEM STATEMENT

We begin this section by considering an illustrative example which serves as the basis of our problem statement and will be used throughout the paper to demonstrate the main features of our system.

Example 1. Alice, a professional photographer, plans to sell her photographs by using the SkyHigh Cloud Services. For her business in the cloud, she has the following requirements: There photographs are downloaded only by users who have paid for her services. Potential buyers are allowed to view her pictures first before they make the payment to obtain the download right. Due to the nature of some of her works, only users from certain countries can view or download some sets of photographs. For some of her works, users are allowed to only view them for a limited time, so that the users cannot reproduce her work easily. In case any dispute arises with a client, she wants to have all the access information of that client. She wants to ensure that the cloud service providers of SkyHigh do not share her data with other service providers, so that the accountability provided for individual users can also be expected from the cloud service providers. With the above scenario in mind, we identify the common requirements and develop several guidelines to achieve data accountability in the cloud. A user who subscribed to a certain cloud service, usually needs to send his/her data as well as associated access control policies (if any) to the service provider. After the data are received by the cloud service provider, the service provider will have granted access rights, such as read, write, and copy, on the data. Using conventional access control mechanisms, once the access rights are granted, the data will be fully available at the service provider. In order to track the actual usage of the data, we aim to develop novel logging and auditing techniques which satisfy the following requirements:

Main Modules:

1. Cloud Information Accountability (CIA) Framework:

CIA framework lies in its ability of maintaining lightweight and powerful accountability that combines aspects of access control, usage control and authentication. By means of the CIA, data owners can track not only whether or not the service-level agreements are being honored, but also enforce access and usage control rules as needed.

2. Distinct mode for auditing:

Push mode:
The push mode refers to logs being periodically sent to the data owner or stakeholder.

Pull mode:
Pull mode refers to an alternative approach whereby the user.(Or another authorized party) can retrieve the logs as needed.

3. Logging and auditing Techniques:

1. The logging should be decentralized in order to adapt to the dynamic nature of the cloud. More specifically, log files should be tightly bounded with the corresponding data being controlled, and require minimal infrastructural support from any server.
2. Every access to the user's data should be correctly and automatically logged. This requires integrated techniques to authenticate the entity who accesses the data, verify, and record the actual operations on the data as well as the time that the data have been accessed.
3. Log files should be reliable and tamper proof to avoid illegal insertion, deletion, and modification by malicious parties. Recovery mechanisms are also desirable to restore damaged log files caused by technical problems.
4. Log files should be sent back to their data owners periodically to inform them of the current usage of their data. More importantly, log files should be retrievable anytime by their data owners when needed regardless the location where the files are stored.

5. The proposed technique should not intrusively monitor data recipients' systems, nor it should introduce heavy communication and computation overhead, which otherwise will hinder its feasibility and adoption in practice.

Cloud Information Accountability (CIA):

In this section, we present an overview of the Cloud Information Accountability framework and discuss how the CIA framework [8] meets the design requirements discussed in the previous section. The Cloud Information Accountability framework proposed in this work conducts automated logging and distributed auditing of relevant access performed by any entity, carried out at any point of time at any cloud service provider. It has two major components: logger and log harmonizer.

Major Components:

There are two major components of the CIA, the first being the logger, and the second being the log harmonizer. The logger is the component which is strongly coupled with the user's data, so that it is downloaded when the data are accessed, and is copied whenever the data are copied. It handles a particular instance or copy of the user's data and is responsible for logging access to that instance or copy. The log harmonizer forms the central component which allows the user access to the log files. The logger is strongly coupled with user's data (either single or multiple data items). Its main tasks include automatically logging access to data items that it contains, encrypting the log record using the

public key of the content owner, and periodically sending them to the log harmonizer. It may also be configured to ensure that access and usage control policies associated with the data are honored. For example, a data owner can specify that user X is only allowed to view but not to modify the data. The logger will control the data access even after it is downloaded by user X. The logger requires only minimal support from the server (e.g., a valid Java virtual machine installed) in order to be deployed. The tight coupling between data and logger, results in a highly distributed logging system, therefore meeting our first design requirement. Furthermore, since the logger does not need to be installed on any system or require any special support from the server, it is not very intrusive in its actions, thus satisfying our fifth requirement. Finally, the logger is also responsible for generating the error correction information for each log record and sends the same to the log harmonizer. The error correction information combined with the encryption and authentication mechanism provides a robust and reliable recovery mechanism, therefore meeting the third requirement. The log harmonizer is responsible for auditing. Being the trusted component, the log harmonizer generates the master key. It holds on to the decryption key for the IBE key pair, as it is responsible for decrypting the logs. Alternatively, the decryption can be carried out on the client end if the path between the log harmonizer and the client is not trusted. In this case, the harmonizer sends the key to the client in a secure key exchange. It supports two auditing strategies: push and pull. Under the push strategy, the log file is pushed back to the data owner periodically in an automated fashion. The pull mode is an on-demand approach, whereby the log file is obtained by the data owner as often as requested. These two modes allow us to satisfy the aforementioned fourth design requirement. In case there exist multiple loggers for the same set of data items, the log harmonizer will merge log records from them before sending back to the data owner. The log harmonizer is also responsible for handling log file corruption. In addition, the log harmonizer can itself carry out logging in addition to auditing. Separating the logging and auditing functions improves the performance. The logger and the log harmonizer are both implemented as lightweight and portable JAR files. The JAR file implementation provides automatic logging functions, which meets the second design requirement.

Advantages:

One of the main innovative features of the CIA framework lies in its ability of maintaining lightweight and powerful accountability that combines aspects of access control, usage control and authentication. Providing defenses against man in middle attack, dictionary attack, Disassembling Attack, Compromised JVM At-tack, Data leakage attack.PDP allows the users to remotely verify the integrity of there data It's Suitable for limited and large number of storages.

5.2. Algorithm of Log Retrieval for Push and Pull mode:

Pushing or Pulling strategies have interesting tradeoffs. The pushing strategy is beneficial when there are a large number of accesses to the data within a short period of time. The pull strategy is most needed when the data owner suspects some misuse of his data; The pull mode allows him to monitor the usage of his content immediately. Supporting both pushing and pulling modes helps protecting from some nontrivial attacks. The algorithm presents logging and synchronization steps with the harmonizer. The log retrieval algorithm for the push and pull modes:

5.2. Tools used for implementing cloud

In the proposed model we are using the following tools:

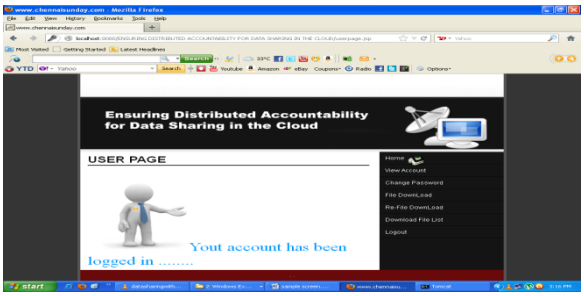
5.2.1. Eucalyptus Cloud

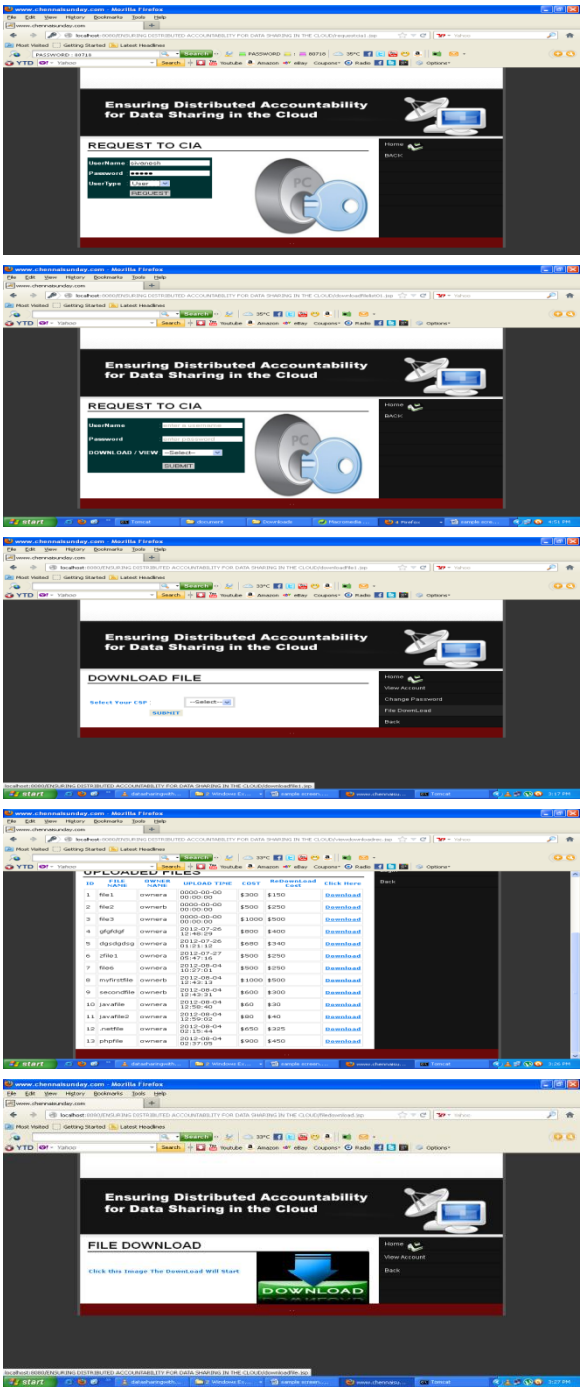
The Eucalyptus Cloud platform is open source software for building AWS-compatible private and hybrid clouds. Eucalyptus supports Amazon web services EC2 and S3 interfaces. It pools together existing virtualized infrastructure to create cloud resources for compute, network and storage.

5.2.2. Amazon EC2

Amazon Elastic compute cloud (EC2) is a central part of Amazon.com's cloud computing platform, Amazon web Services (AWS). EC2 allows users to rent virtual computers on which to run their own Computer Applications. They are designed for control and management of VM instances, EBS volumes, elastic IPs, and security groups and should work well with EC2 and Eucalyptus [10].

SCREEN SHOTS





V. CONCLUSION

This paper presents effective mechanism, which performs automatic authentication of users and create log records of each data access by the user. Data owner can audit his content on cloud, and he can get the confirmation that his data is safe on the cloud. Data owner also able to know the duplication of data made without his knowledge. Data owner should not worry about his data on cloud using this mechanism and data usage is transparent, using this mechanism. In future we would like to develop a cloud, on which we will install JRE and JVM, to do the authentication of JAR. Try to improve security of store data and to reduce log record generation time.

REFERENCES:

[1]. P. Ammann and S. Jajodia, "Distributed Timestamp Generation in Planar Lattice Networks," ACM Trans. Computer Systems, vol. 11, pp. 205-225, Aug. 1993.

[2]. G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," Proc. ACM Conf. Computer and Comm. Security, pp. 598-609, 2007.

[3]. E. Barka and A. Lakas, "Integrating Usage Control with SIP-Based Communications," J. Computer Systems, Networks, and Comm., vol. 2008, pp. 1-8, 2008.

[4] D. Boneh and M.K. Franklin, "Identity-Based Encryption from the Weil Pairing," Proc. Int'l Cryptology Conf. Advances in Cryptology, pp. 213-229, 2001.

[5] R. Bose and J. Frew, "Lineage Retrieval for Scientific Data Processing: A Survey," ACM Computing Surveys, vol. 37, pp. 1-28, Mar. 2005.

[6] P. Buneman, A. Chapman, and J. Cheney, "Provenance Management in Curated Databases," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '06), pp. 539-550, 2006.

[7] B. Chun and A.C. Bavier, "Decentralized Trust Management and Accountability in Federated Systems," Proc. Ann. Hawaii Int'l Conf. System Sciences (HICSS), 2004.

[8] OASIS Security Services Technical Committee, "Security Assertion Markup Language (saml) 2.0," committees/tc home.php?wg abbrev=security, 2012.

[9] R. Corin, S. Etalle, J.I. den Hartog, G. Lenzini, and I. Staicu, "A Logic for Auditing Accountability in Decentralized Systems," Proc. IFIP TC1 WG1.7 Workshop Formal Aspects in Security and Trust, pp. 187-201, 2005.

[10] B. Crispo and G. Ruffo, "Reasoning about Accountability within Delegation," Proc. Third Int'l Conf. Information and Comm. Security (ICICS), pp. 251-260, 2001.

[11] R. Hasan, R. Sion, and M. Winslett, "The Case of the Fake Picasso: Preventing History Forgery with Secure Provenance," Proc. Seventh Conf. File and Storage Technologies, pp. 1-14, 2009.

[12] J. Hightower and G. Borriello, "Location Systems for Ubiquitous Computing," Computer, vol. 34, no. 8, pp. 57-66, Aug. 2001.

[13] J.W. Holford, W.J. Caelli, and A.W.

Rhodes, "Using Self-Defending Objects to Develop Security Aware Applications in Java," Proc. 27th Australasian Conf. Computer Science, vol. 26, 341-349, 2004.

[14] P.T. Jaeger, J. Lin, and J.M. Grimes, "Cloud Computing and Information Policy: Computing in a Policy Cloud?," J. Information Technology and Politics, vol. 5, no. 3, pp. 269-283, 2009.

[15] R. Jagadeesan, A. Jeffrey, C. Pitcher, and J. Riely, "Towards a Theory of Accountability and Audit," Proc. 14th European Conf. Research in Computer Security (ESORICS), pp. 152-167, 2009.

Author's Profile

Mr. Umamaheswara Rao, Post Graduated in Computer Science & Engineering (M.Tech) , Jawaharlal Nehru Technological University Hyderabad , 2013 and Master of Computer Application (M.C.A) Jawaharlal Nehru Technological University, Kakinada, 2009. He is working presently as an Assistant Professor in Department of Computer Science & Engineering in **St.Martin's Engineering College**, Dhulapally, Secunderabad, RR Dist, and A.P, INDIA. He has 3+ years Experience. His Research Interests Include Network Security, Cloud Computing & Data Warehousing and Data Mining.

Mr. B.Srinivasulu, Post Graduated in Computer Science Engineering (**M.Tech**) From **JNTU**, Hyderabad in 2010 and Graduated in Computer Science Engineering (B.Tech) from JNTUH, in 2008. He is working as an Assistant Professor in Department of Computer Science & Engineering in **St.Martin's Engineering College**, R.R Dist, AP, India. He has 3+ years of Teaching Experience. His Research Interests Include Network Security, Cloud Computing & Data Warehousing and Data Mining.

Mr. R.V.GANDHI, Post Graduated in Computer Science & Engineering (M.Tech), Jawaharlal Nehru Technological University Hyderabad , 2009 and Bachelor of Technology (B.Tech) in Computer Science & Engineering, Jawaharlal Nehru Technological University, Hyderabad , 2007. He is working presently as an Assistant Professor in Department of Computer Science & Engineering in **Mother Theresa Institute of Engineering and Technology**, Melumoi, Palamaner, Chitoor Dist, A.P, INDIA. He has 4+ years Experience. His Research Interests Include Network Security, Cloud Computing & Data Warehousing and Data Mining.

Mrs. B.Venkata Ramana, Post Graduated in Computer Science (M.Tech), JNTUH, 2010, and Graduated in Computer Science & Engineering (B.Tech) From JNTU Hyderabad, 2005.She is working presently as an Assitant Professor in Department of Computer Science & Engineering in Holy Mary Institute of Engineering & Technology, RR Dist, A.P, INDIA. She has 6+ years Experience. Her Research Interests Include Software Engineering, Cloud Computing, Operating Systems & Information Security.

.