Improvising Distributed Accountability by Using Fog Methodology

K. Nagendra, A. Suresh Babu

Abstract— In cloud computing environment resources are shared among various clients and it's important for system provider to allocate the necessary resources for the clients. And IT infrastructure proceeds as the amount increases to grow, cloud computing is a new way of virtualization technologies that enable management of virtual machines over a plethora of physically connected systems [13]Cloud computing provides on demand services. Multiple users need to try and do business of their information exploitation cloud however they get worry to losing their information. Whereas data owner can store his/her information on cloud, he should get confirmation that his/her information is safe on cloud. To unravel higher than downside during this paper this offers effective mechanism to trace usage of information exploitation accountability. Accountability is verification of security policies and it's necessary for clear information access. In this paper shows automatic work mechanisms exploitation JAR programming that improves security and privacy of information in cloud. We provide an effective mechanism known as fog computing to protect user's data from theft by confusing attacker with unuseful information. Exploitation this mechanism data owner might apprehend his/her information is handled as per his demand or service level agreement.

Index Terms—Cloud computing, accountability, security, data sharing, privacy

I. INTRODUCTION

Cloud computing could be a technology that uses internet and remote servers to store information and application. In cloud there's no have to be compelled to install specific hardware, software package on user machine, therefore user will get the specified infrastructure on his machine in low rates. Cloud computing is an infrastructure that provides helpful, on demand network services to use numerous resources with less effort. options of Cloud computing are, immense access of information, application, resources and hardware while not installation of any software package, user will access the information from any machine or any wherever within the world, business will get resource in one place, that's means that cloud computing provides quantifiability in on demand services to the business users. Everybody unbroken their information in cloud, therefore it becomes public therefore security issue will increase towards non-public information.

Information usage in cloud is incredibly massive by users and businesses; therefore information security in cloud is incredibly vital issue to unravel. Several users need to try and do business of his information through cloud, however users might not recognize the machines that truly method and host their information.

Manuscript received August, 2013.

K.Nagendra, CSE, Jntuacep, Pulivendula, Kadapa, Andhra Pradesh, India.

Dr.A. Suresh Babu is with the Department of Cse, Jntuacep, Pulivendula, Kadapa, and Andhra Pradesh, India.

Whereas enjoying the convenience brought by this new technology, users additionally begin worrying concerning losing management of their own information.

Cloud provides 3 service models that are; platform as a service, infrastructure as a service and computer code as a service. Underneath the info as a service, this is often having four components as per mentioned below,

- Encryption and Decryption - For security purpose of data kept in cloud; encryption appears to be accurate security solution.
- Key Management - If encryption is necessary to store data in the cloud, then encryption keys are not saved, but the user needs key management.
- Authentication - For accessing stored data in cloud by authorized users.
- Authorization – Rights given to user as well as cloud provider.

To solve the protection issues in cloud; various users can't browse the individual user's data whereas not having access. Data owner mustn't trouble relating to his data, and will not get concern relating to harm of his data by hacker; there is would like of security mechanism that is ready to trace usage of information among the cloud. Accountability is very important for observation data usage, throughout this all actions of users like inflicting of file are cryptographically joined to the server, which executes them as well as it manages protected record of all the actions of past and server can use the past records to grasp the correctness of action. It together provides reliable data relating to usage of data and it observes all the records, therefore it helps in build trust, relationship and name. Therefore accountability is for verification of authentication and authorization. It's powerful tool to ascertain the authorization policies. Accountability describes authorization demand for data usage policies. Accountability mechanisms, that suppose once the actual fact verification are attractive implies that to enforce authorization policies.

There are 7 phases of accountability

1. Policy setting with data
2. Use of data by users
3. Logging
4. Merge logs
5. Error correctness in log
6. Auditing
7. Rectify and improvement.

hese phases will be modifies as per structure.

First information owner can set the policies with data and send it to cloud service supplier (CSP), information are use by users and logs of every record are created, then log are incorporate and error correction in log has been done and in auditing logs are checked and in last section improvement has been done [12].

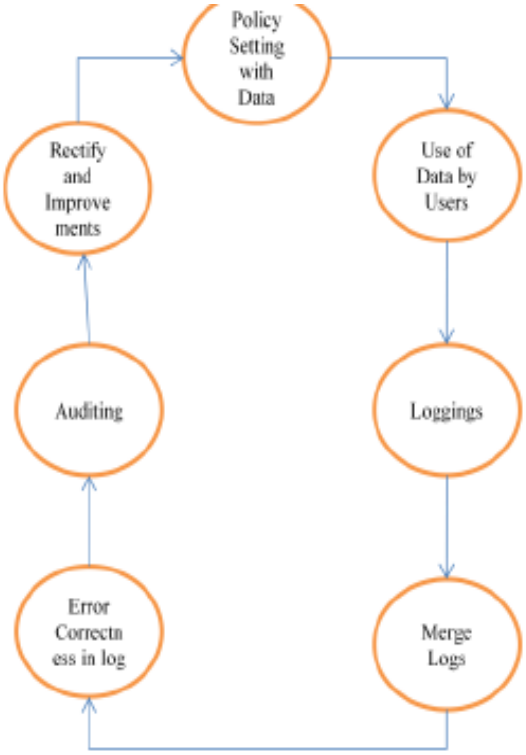

Fig 1: Phases of Accountability

In the Fig 1 Steps of accountability is given these are seven steps every step is very important to perform next step, accountability is nothing however validation of user actions means that user having rights for accessing this information or not. Suppose user can do misuse of information or resources then network or data owner can take action on that thus users, businesses and government mustn't trouble regarding their information on cloud.

II. PROBLEM SCENARIO

Now-a-days, cloud computing model is key aspects of various Internet services and increasing fraction of time people Spend on computers at present. It permits clients to only costs for the computing resources they need, when they need them. The cost effective manner and to lower the barrier to entry for such applications and it is a cloud-based applications to enabled supports [4] but at the same time the security issues has created the barriers to the wide adoption of the cloud services.

Suppose Santa wants to upload her data to some Web cloud service. User has the following requirements

a) User wants to sign a formal SLA with the Cloud service provider and user wants that her SLA should be followed strictly.

b) The expected user may view her application demo for a specific timing.

c) If some user wants to download her application then that user has to get permission from CKG (cloud key Generator).

d) User desire to assure that the cloud service provider of "Web cloud Service" 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.

e) All the user data that has downloaded Santa's application will be sent to her periodically or it will Store in a third party place from there Alice can take them.

Keep above model in thought, many principles have been fixed and the common requirements are also identified to achieve accountability in cloud. As user who desires to combine the cloud service has to give his/her personal data as well as access control policies. Then the Service provider will have granted access assistance on the information. It will be fully available to the cloud service provider when after the completion of transferring data in the cloud.

III. LITERATURE SURVEY

In this section review connected works addressing security in cloud. Security issue is incredibly necessary in cloud there are several techniques out there thus here is review of these.

S. Pearson et al describes privacy manager mechanism within which user's data is safe on cloud , during this technique the user's information is in encrypted type in cloud and evaluating is completed on encrypted knowledge, the privacy manager build clear information from results of analysis manager to induce the right result. In obfuscation data isn't gift on Service provider's machine thus there's no risk with data, thus data is safe on cloud, however this resolution isn't appropriate for all cloud application, once input file is massive this technique will still need an outsized quantity of memory. within the authors gift procedural and technical resolution each are manufacturing resolution to accountability to resolution security risk in cloud during this mechanism these policies are determined by the parties that use, store or share that information regardless of the jurisdiction within which info is processed. However it's limitation that information processed on SP is in unencrypted at the purpose of process thus there's a risk of information leak. In, the author offers a language which allows serving information with policies by agent; agent ought to prove their action and authorization to use specific information. During this logic data owner attach Policies with information, which contain an outline of that actions are allowed with that information, however there's the matter of Continuous auditing of agent, however they supply resolution that inaccurate behavior. They should be monitor and agent should be offer justification for his or her action, afterward authority can check the justification. In [5], authors offers a 3 layer design that defend info leak from cloud, it provides 3 layer to guard information, in 1st layer the service supplier shouldn't read confidential information in second layer service supplier shouldn't do the assortment of information, in third layer user specify use of his information and assortment in policies, thus policies continually travel with knowledge. In [6], authors gift accountability in united system to attain trust management. The reliability towards usage of raw materials is sophisticated through accountability thus to resolve drawback for trust management in united system they need given 3 layers design, in 1st layer is authentication and authorization during this authentication will victimization public key cryptography. Second layer is accountability that performs observation and work. The third layer is anomaly detection that detects misuse of resources. This mechanism needs third party services to watch network resources.

IV. ENHANCING THE ACCOUNTABILITY

Cloud computing may be a massive infrastructure which give several services to user while not installation of

resources on their own machine. This is often the pay as you utilize model. Samples of the cloud services are Yahoo email, Google, Gmail and Hotmail. There are several users, businesses, government uses cloud, thus knowledge usage in cloud is massive. Thus knowledge maintenance in cloud is advanced. Several Artists desires to try to business of their art victimization cloud. As an example one amongst the creative person need to sell his painting victimization cloud then he need that his paintings should be safe on cloud nobody will misuse his paintings.

A. Cloud Ingredients

There is need to be compelled to offer technique that is ready to audit information in cloud. On the idea of accountability, we've an inclination to projected one mechanism that keeps use information clear suggests that data owner got to get information regarding use of his information. This process support accountability in distributed area, data owner should not problem regarding his information, he may acknowledge his information is handled per service level agreement and his information is riskless on cloud. Data owner will determine the authorization principles and policies and user will handle information victimization this rule and logs of each information access are created. Throughout this mechanism there are unit two main parts i.e. logger and log harmonizer.

The feller is with the data owner's information, it provides work access to information and encrypts log record by pattern public key that's given by data owner and send it to log harmonizer. The log harmonizer is taking part in the observance and rectifying, it generates the key it holds cryptography key decrypting the logs, and at the consumer side cryptography it sends key to shopper. Throughout this mechanism data owner will creates personal key and public key, pattern generated key owner will produce feller that will be a JAR file, it encloses his authorization principles and work policies with information send to cloud service provider.

Authentication of cloud service provider has been done exploitation open SSL based totally certificates once authentication of cloud service provider user are able to access information in JAR, log of each data usage has been generated and encrypted exploitation public key and it automatically send to log harmonizer for integrity log records are signed by entity that's exploitation the information and log records are decrypted and accessed by owner. In push state logs are automatically transferred to data owner and in pull state owner may claim logs, therefore he may observe information access at anytime, anywhere and he can do inspection of his information.

B. Flow of Data

The overall CIA framework, combining information, users, logger and harmonizer is sketched in Fig. 2. At the start, every user creates a combine of public and personal keys supported Identity-Based encoding [4] (in Fig. 2). This IBE scheme could be a Weil-pairing-based IBE scheme that protects us against one among the most current attacks to our design as described in Section 7. Exploitation the generated key, the user can produce a logger part that may be a JAR file, to store its data items.

The JAR file includes a collection of easy access management rules specifying whether and the way the cloud servers, and probably different information stakeholders (users, companies) are licensed to access the content itself. At the same time, he transfers the JAR file to the cloud service provider that he subscribes to. To certify the CSP to the JAR (in Fig. 2), we have a tendency to use OpenSSL- primarily based certificates, whereby a trustworthy certificate authority certifies the CSP. Within the event that the access is requested by a user, we have a tendency to use SAML-based authentication [14], whereby a reliability identity provider problems certificates confirmative the user's identity supported his username.

Once the authentication succeeds, the service providers (or the user) are going to be allowed to access the information enveloped within the JAR. Depending on the configuration settings outlined at the time of creation, the JAR can give usage management related to logging, or can give solely work practicality. As for the work, when there's associate access to the information, the JAR can mechanically generate a log record, encipher it victimization the general public key distributed by the data owner, and store it alongside the information (in Fig. 2). The encoding of the log file prevents unauthorized changes to the file by attackers.

The data owner could opt to reuse the same key pair for all JARs or create different key pairs for different JARs. Using separate keys are able to improve the authorization (detailed discussion is in Section 7) without introducing any overhead except in the starting phase. In inclusion, some error correction data will be sent to the log harmonizer to handle possible log file corruption (in Fig. 1). To ensure reliability of the logs, each record is signed by the entity accessing the content. In earlier, own records are hashed together to create a chain formation, can easily identify possible errors or losts files. The encrypted log records may be decrypted afterward and their integrity checked. They will be accessed by the data owner and other authorized stakeholders at any time for auditing purposes with the aid of the log harmonizer (in Fig. 1).

Our proposed framework prevents various attacks such as detecting illegal copies of users' information. Hence our work is distinct from normal logging methods which use encryption to secure log records. Their logging techniques are neither automatic nor shared. They request the information to lie within the boundaries of the centralized system for the logging to be able, which is not appropriate in the cloud

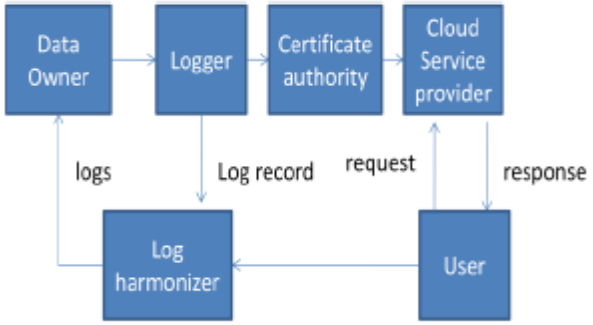

Fig 2: Accountability Mechanism in cloud

State transition diagram is machine that shows no of states, machine take input from outside world and every input will turn out machine to travel next step. Following transition diagram shows the various states of Accountability mechanism in cloud i.e. however it changes from one state to next state.

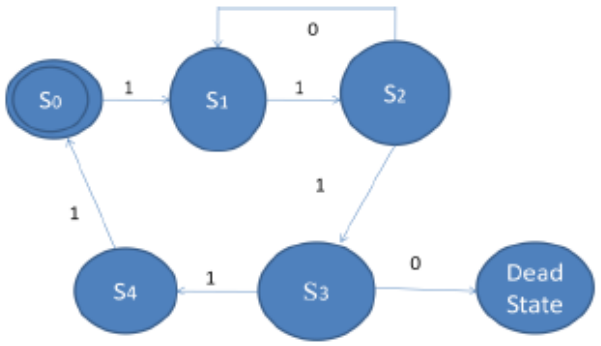

Fig 3: State Transition Diagram

Where,

0: Unsuccessful

1: Successful

Transitions are:

S0: Data Owner will send data to logger.

S1: Data Owner will create logger which is a jar file to store data and principles. .

S2: Authentication of CSP to JAR file.

S3: Authentication of user.

S4: owner can see merge log

C. Fog computing methodology

In this paper which proposes a different approach for securing data in the cloud using offensive decoy method. We supervise information access in the cloud and detect abnormal data access patterns. When unsecured access is identified and after checked by raising queries, we utilize a mislead attack by forwarding large amounts of decoy information to the attacker. This prevents the utilization of the user's own information. Hypothesis supervises in a local file context provides evidence that this approach may provide unprecedented levels of user data security in a Cloud environment.

We use this technology to launch disinformation attacks against harmful groups, protecting from noticeably the real sensitive customer data from fake worthless data. In this paper, which propose two ways of using Fog computing to prevent attacks such as the Twitter attack, by retrieving decoy data inside the Cloud by the Cloud service customer and within personal online social networking profiles by individual user. The basic idea is that we can limit the damage of stolen data if we decrease the value of that stolen information to the attacker. We can achieve this through a 'preventive' mislead attack. We assume that protected Cloud events may be implemented by given two additional security features:

1) User Behavior Profiling:

It is expected that access to a user's information in the Cloud will exhibit a normal access. User profiling is a popular method that can be applied here to design how, when, and how much a client utilizes their data in the Cloud. Such 'normal user' behavior can be continuously checked to determine whether abnormal access to a user's information is happening or not. This procedure of behavior-basis protection is commonly used in fraud detection applications. Such prominence usually consists of metered information, how many documents are commonly read. These user-distinguish features may serve to detect abnormal Cloud access based partially upon the scale and scope of data transferred [13].

2) Decoys:

Decoy information, such as decoy documents, honey files, honey pots, and various other bogus information can be generated on demand and serve as a means of detecting unauthorized access to information and to 'poison' the thief's ex-filtrated information. Serving decoys will confound and confuse an attacker into believing they have ex-filtrated useful information or not. This method may be integrated with user behavior profiling technology to secure a user's information in the Cloud. Whenever exceptional access to a cloud events are observed, decoy data can be get back by the Cloud and delivered in such a way as to appear completely lawful. The real user, who is the owner of the data, would readily identify when decoy information is being returned by the Cloud, and might alter the Cloud's responses through a variety of means, such as challenge questions, to inform the Cloud security system that it has inaccurately detected an unsecured access. In the situation where the access is accurately identified as an unauthorized access, the Cloud security system would deliver unbounded amounts of bogus information to the attacker, thus protecting the user's real information from unsecured disclosure. The decoys contributes two features: (1) validating whether data access is authorized when abnormal information access is detected, and (2) confusing the attacker with bogus information.

These posit that the combination of these two security features will provide unprecedented levels of security for the Cloud. No current Cloud security mechanism is available that provides this level of security.

V. PEER –TO-PEER AUDITING MODE

Let us describe, distributed auditing mechanism including the algorithms for data owners to query the logs regarding their data.

A. PULL AND PUSH MODE

To allow users to be timely and accurately informed about these data usage, the distributed logging mechanism is complemented by an innovative auditing mechanism. Support two complementary auditing modes: 1) Push mode; 2) pull mode.

Push mode. In that mode, the logs are periodically pushed to the data owner by the harmonizer. The push mode may be activated by the following two events: one is that the time elapses for a certain period according to the temporal timer inserted as part of the JAR file; the other is that the JAR file exceeds the size stipulated by the content owner at the time of generation. And then logs are forwarded to the data owner, the log files will be deleted to empty the space for further purpose. Including with the log files, the error accurate information for those logs is also dumped. The push mode is the basic mode which can be adopted by both the pure log and the access logs, instead of whether there is a request from the data owner for the log files. This action contributes two significant functions in the logging architecture:

(1) It assures the size of the log files does not explode and

(2) It enables timely detection and correction of any loss or damage to the log files.

Concerning the latter function, Notice that the auditor, upon receiving the log file, will check its cryptographic guarantees, by checking the record's integrity and validation. By building of the records, the data owner will be able to quickly detect fraudulence of entries, by utilizing the Checksum joined to all records.

Pull mode allows auditors to retrieve the logs anytime to check the recent access to these own data. The pull message consists simply of an FTP pull command, which will be turnout from the command line. For experienced users, a wizard consisting a batch file may be easily constructed. The request can be forwarded to the harmonizer, and the user may be known of the information's locations and obtain an integrated copy of the authentic and sealed log file.

Algorithm for pull and push pure Log mode

Require: size: log file size for maximum, **time:** maximum time allowed to before the log file is wasted, **tbeg: timestamp** at which the last dump happened, **log: current** log file, **Pull**; command is received from data owner.

```
Let TS (NTP) be the network time protocol timestamp
Pull=0
rec :=< UID, DOID, Access Type, Result, Time, Loc>
 lsize: =sizeof (log)
 If ((cuttimetbeg)<time)&&(lsize<size)&&(pull==0)then
   Log: =log+ENCRYPT (rec)
   PING to CJAR
   If PING-CJAR then
     PUSH RS (rec)
   Else
     EXIT (1)
   Endif
Endif
If ((cutime-tbeg)>time) || (lsize>=size)
  If PING-CJAR then
    PUSH log RS (LOG):=NULL
    Tbeg: =TS (NTP)
    PULL: =0
  Else
    EXIT (1)
  Endif
Endif
```

The algorithm presents logging and Synchronization processing with the harmonizer in case of PureLog. Check size and time of the log file. The size and time threshold for a dump are specified by the data owner at the time of creation of the JAR. Data owner requested to log files are checked. If none of these events are happened, it continues to conceal the record and write the error-correction information to the harmonizer. The interaction with the harmonizer starts with a simple handshake. If no reply gets back, then the log file registers an error. After the data owner is alerted through e-mails, and after the JAR is setup to forward error messages. Once the handshake is completed, the communications with the harmonizer proceed. In case of Access Log, the above algorithm is modified by adding an additional check after step 6.AccessLog check the CSP for satisfies condition specified in the policies. If the conditions are fulfilled then access will proceeds; otherwise, it will losts. Regardless of the access control result, they tried access to the information in the JAR file will be logged. Auditing mechanism has two main advantages. It guarantees a high level of availability of the logs and the use of the harmonizer minimizes the amount of workload for human users in going through long log files sent by different copies of JAR files.

VI. OVERVIEW OF ATTACKS

An attacker may intercept messages during the authentication of a service provider with the certificate authority, and respond back the messages in order to conceal as a legal service provider. The two points are that the attacker can replay the messages. The first point is after the actual service provider has completely disconnected and ended a session with the certificate authority. The other is when the actual service provider is disconnected but the session was not completed, then the attacker will go to renegotiate the connection. The first attack does not approach since the certificate typically has a time stamp which will become obsolete at the time point of reutilize. The second one may be unsuccessful since renegotiation is banned in the latest version of OpenSSL and cryptographic checks have been added.

VII. PERFORMANCE SURVEY

In this part, we initialize the context of the test environment and then present the performance study of our system.

A. EXPERIMENTAL ENVIRONMENT

We tested our CIA framework by setting up a small cloud, using the Emulab testbed [16]. In particular, the test environment consists of several OpenSSL-enabled servers: one head node which is the certificate authority, and distinct nodes. Each of the servers is installed with Eucalyptus [15]. Eucalyptus/Walrus is an open source cloud implementation for Linux systems which is loosely based on Amazon EC2, thus contributes the strong emerging functionalities of Amazon EC2 into the open source domain. We used Linux-based servers running Ubuntu 12.04 server OS. Each server has a 64-bit Core2Duo processor, 4 GB RAM, and a 500 GB HDD. Each server is fitted to execute the OpenJDK runtime environment with IcedTea6 2.3.9.

VIII. CONCLUSION AND FUTURE VISION

This paper presents effective mechanism that performs automatic authentication of users and make log records of every information access by the user. Data owner will audit his content on cloud, and he will get the confirmation that his information is safe on the cloud. Data owner additionally able to recognize the duplication data of information created while not his data. Data owner mustn't worry concerning his knowledge on cloud exploitation this mechanism and information usage is clear, exploitation this mechanism.

In future we would like to enhance a cloud, on which we will install JRE and JVM, to do the validation of JAR. Refine to enhance the protection of accumulated data and to reduce log record generation time.

REFERENCES

[1] Smitha Sundareswaran, Anna C. Squicciarini and Dan Lin, "Ensuring Distributed Accountability for Data Sharing in the Cloud," IEEE Transaction on dependable a secure computing, VOL. 9, NO. 4, pg 556-568, 2012.

[2] S. Pearson, Y. Shen, and M. Mowbray," A privacy Manager for Cloud Computing," Proc. Int'l Conf. Cloud Computing (cloudcom), pp.90-106, 2009.

[3] S. Pearson and A. Charlesworth, "Accountability as a Way Forward for Privacy Protection in the Cloud," *Proc First Int'l conf. Cloud* Computing, 2009.

[4] 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.

[5] A. Squicciarini, S. Sundareswaran and D. Lin, "Preventing Information Leakage from Indexing in the Cloud," *Proc. IEEE Int'l*

Conf. Cloud Computing, 2010.
[6] B. Chun and A. C. Bavier,"Decentralized Trust Management and Accountability in Federated System," *Proc. Ann. Hawaii Int'l Conf. System Science (HICSS), 2004.*
[7] B. Crispo and G. Ruffo, "Reasoning about Accountability within Delegation," Proc. Third Int'l Conf. Information and Comm. Security (ICICS), pp. 251-260, 2001.
[8] S. Sundareswaran, A. Squicciarini, D. Lin, and S. Huang, "Promoting Distributed Accountability in the Cloud," *Proc. IEEE* Int'l Conf. Cloud Computing, 2011.
[9] D.J. Weitzner, H. Abelson, T. Berners-Lee, J. Feigen-baum, J. Hendler, and G.J. Sussman, "Information Accountability," Comm. ACM, vol. 51, no. 6, pp. 82-87, 2008.
[10] B. Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C. John Wiley & Sons, 1993.
[11] Praveen Gauravaram, John Kelesy, Lars Knudsen, and Soren Thomsen, "On Hash function using Checksums"
[12] Ryan K L Ko, Peter Jagadpramana, Miranda Mowbray, Siani Pearson, Markus Kirchberg, Qianhui, "TrustCloud: A Framework for Accountability and Trust in Cloud Computing" HP Laboratories, pp 1 – 7, HPL-2011-38
[13] M. Ben-Salem and S. J. Stolfo, "Modeling user search-behavior for Masquerade detection," in Proceedings of the 14th International Symposium on Recent Advances in Intrusion Detection. Heidelberg: Springer, September 2011, pp. 1-20.
[14] R. Corin, S. Etalle, J.I. den Hartog, G. Lenzini, and I. Staicu, "A Logic for Auditing Accountability in Decentralized Systems,"
[15] Eucalyptus Systems, http://www.eucalyptus.com/, 2013.
[16] Emulab Network Emulation Testbed, www.emulab.net, 2013.

K.Nagendra received the bachelor's degree in Information technology in 2010 from Jntu Anantapur. He is currently pursuing the master's degree in CSE in the college of JNTUACEP.

Dr.A.Suresh Babu received the PhD degree in Information Extraction Systems in Data Mining from the University of Jntu Anantapur in 2013. He is an assistant professor at the Jntu college of Engineering, Pulivendula, Kadapa, Andhra Pradesh, India. His research interests include Data Mining and Cloud Computing.

.