ISSN 2278 - 3091

International Journal of Advanced Trends in Computer Science and Engineering (IJATCSE), Vol.2 , No.5, Pages : 09-14 (2013)
Special Issue of ICCECT 2013 - Held during September 20, 2013, Bangalore, India

Enhancing Distributed Accountability by Using Proxy Re-encryption Scheme

K.Nagendra[1], Dr. A.Suresh Babu[2]
[1]PG scholar, JNTUACEP, Andhra Pradesh, India, sandunagi@gmail.com
[2]Assisant Professor, JNTUACEP, Andhra Pradesh, India,asureshjntu@gmail.com

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 proxy re-encryption scheme to supports encoding operations over encrypted messages as well as forwarding operations over encoded and encrypted messages. Our method fully integrates encrypting, encoding, and forwarding. 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

INTRODUCTION

In simple terms, cloud computing are often softened to a browser primarily based application that's hosted on a remote server. To the common user, that's all he or she very has to understand cloud computing. However there's a lot additional to that than simply that. What cloud computing really represents is huge: it's the simplest way for small organizations to contend with abundant larger ones, it's the simplest way to save lots of lots of money and it's the simplest way to utilize energy efficiency in operations.

Cloud computing because it relates to web technology is all around United States. Once we access our email, after we search for data, we are utilizing the ability of processing technology that exists at a far off location without United States knowing regarding it. In fact, even the foremost basic computer applications need a network connection recently to try and do easy tasks.

In effect, the cloud provides networked users with an extension of their own machine. As long as a user is connected to the web, the power of cloud computing comes into play and lots of advantages are often reaped. One example would be processing power. Applications may be run on the fly from a terminal machine once processing power isn't a concern; the only issue that users ought to worry regarding would be their bandwidth affiliation and its reliability on the network.

Service models

In the deployment model, completely different cloud types are an expression of the way during which infrastructure is deployed. We will think about the cloud because the boundary between wherever a client's network, management, and responsibilities ends and also the cloud service providers begins. As cloud computing has developed, different vendors provide clouds that have different services related to them. The portfolio of services offered adds another set of definitions referred to as the service model.

There are many alternative service models represented within the literature, all of that take the subsequent form: XaaS, or "as a Service" Three service types are universally accepted:

• **Infrastructure as a Service** IaaS provides virtual machines, virtual storage, virtual infrastructure, and alternative hardware assets as resources that clients will provision. The IaaS service provider manages the complete infrastructure, whereas the client is responsible for all alternative aspects of the deployment. This may comprises the operating system, applications, and user interactions with the system.

Examples of IaaS service providers include:
- Amazon Elastic Compute Cloud (EC2)
- Eucalyptus
- GoGrid

• **Platform as a Service** PaaS provides virtual machines, operating systems, applications, services, development frameworks, transactions, and control structures. The client will deploy its applications on the cloud infrastructure or use applications that were programmed utilizing languages and tools that are supported by the PaaS service provider. The service provider manages the cloud infrastructure, the operating systems, and also the enabling software. The client is responsible for installing and managing the application that it is deploying. A PaaS service adds integration features, middleware, and alternative orchestration and choreography services to the IaaS model. Samples of PaaS services are:
- Force.com
- GoGrid Cloud Center
- Google AppEngine
- Windows Azure Platform

• **Software as a Service** SaaS may be a complete operating environment with applications, management, and also the user interface. Within the SaaS model, the application is provided to the client through a thin client interface, and also the customer's responsibility begins and ends with entering and managing its information and user interaction. Everything from the application all the way down to the infrastructure is that the vendor's responsibility.

.

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 stages 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.

These stages 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].

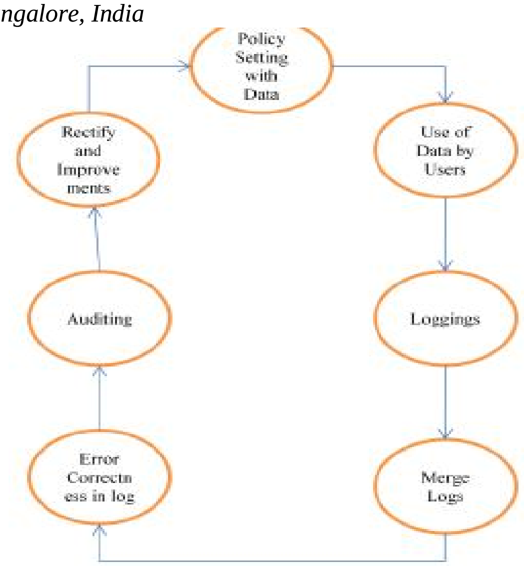

Fig 1 shows Stages of Accountability

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.

PROBLEM OVERVIEW

We begin this section by considering an informative example that aids as the basis of our problem statement to validate the main features of our system. Example 1. For an assumption, Mitrajith an expert photographer plans to sell his pictures by utilizing the Google Cloud Services. For his business within the cloud, he has the subsequent specifications.

• His images are accessed only by the users who have acquired his services.

• Dormant buyers are allowed to only scan his photos that are made the payment for getting 6 months membership.

• As a result of the nature of some of his works only users who had made the payment for getting 37 months membership will read and comment.

• For a few of his works users are allowed to read, write &amp; execute who had made the payment for getting premium membership

• In case any dispute arises with a client he needs to possess all the access information of that client.

• He needs to make sure that the CSP of Google doesn't share his information with different service providers, so the accountability furnished for individual users can even be expected from the CSP.

With the above outline in mind, we analyze the common requirements and develop several instructions to achieve data accountability in the cloud. A user, who enrolled to a certain cloud service, regularly needs to send his/her data as well as allied access control policies (if any) to the service provider. After the data are gotten by the cloud service provider, the service provider will be having the granted access rights, such as read, write, and execute, on the data. Using conventional access control systems, once the access rights are granted, the data will be fully accessible at the service provider.

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 Constituents

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. Process 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

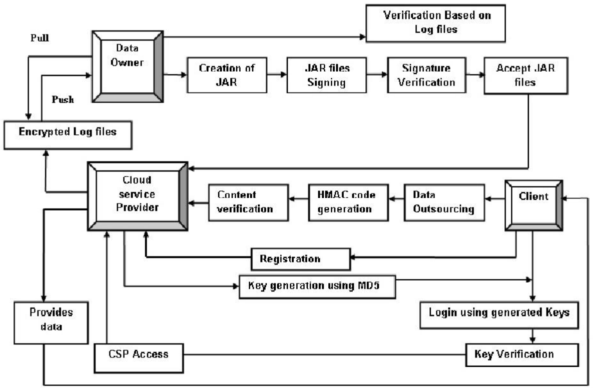

Fig 2 shows Accountability Mechanism in cloud

C. Proxy Re-Encryption Scheme

Proxy re-encryption schemes are proposed by Mambo and Okamoto [14] and Blaze et al. [15]. During a proxy re-encryption scheme, a proxy server will transfer a ciphertext under a public key PKA to a new one under another public key PKB by utilizing the re-encryption key RKA!B. The server doesn't know the plaintext throughout transformation. Ateniese et al. [16] proposed some proxy re-encryption schemes and applied them to the sharing function of secure storage systems. In their work, messages are initial encrypted by the owner and so keep in a storage server. Once a user needs to share his messages, he sends a re-encryption key to the storage server. The storage server re-encrypts the encrypted messages for the licensed user. Thus, their system has information confidentiality and supports the data forwarding function. Our work additional integrates encryption, re-encryption, and encoding such that storage robustness is strengthened.

Type-based proxy re-encryption schemes proposed by Tang [17] offer a much better granularity on the granted right of a re-encryption key. A user will decide which kind of messages and with whom he needs to share during this kind of proxy re- encryption schemes. Key-private proxy re-encryption schemes are proposed by Ateniese et al. [18]. During a key-private proxy re-encryption scheme, given a re-encryption key, a proxy server cannot verify the identity of the recipient. This sort of proxy re-encryption schemes provides higher privacy guarantee against proxy servers. Although most proxy re-encryption schemes use pairing operations, there exist proxy re-encryption schemes while not pairing [19].

An encryption scheme is multiplicative homomorphic if it supports a group operation on encrypted plaintexts without decryption. The multiplicative homomorphic encryption scheme supports the encoding operation over encrypted messages. We tend to then convert a proxy re-encryption scheme with multiplicative homomorphic property into a threshold version. A secret key is shared to key servers with a threshold value t. To decrypt for a group of k message symbols, each key server independently queries 2 storage servers and partially decrypts two encrypted codeword symbols. As long as t key servers are out there, k codeword symbols are obtained from the partially decrypted cipher texts.

So as to preserve privacy, the shoppers can encrypt their information once they out- source it to the cloud. However, the encrypted type of information greatly impedes the utilization because of its randomness. Several efforts are finished the purpose of data usage however without undermining the information privacy. Homomorphism: Given two cipher texts c1 and c2 on plaintexts m1 and m2 respectively, one will get the cipher text on the plaintext m1 +m2 and/or m1 •m2 by evaluating c1 and c2 while not decrypting cipher texts. Proxy re-encryption: Given a proxy re-encryption key, the proxy will transform a cipher text of 1 user to a cipher text of the target user. Threshold decryption: By dividing the non-public key into many pieces of secret shares, all clients will work along to decrypt the cipher text – the output of the function.

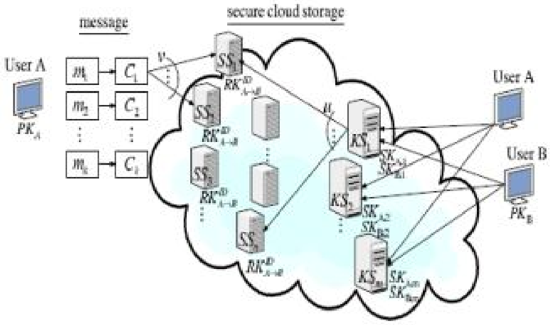

Fig 3 shows A General System Model of Work

PEER –TO-PEER AUDITING MECHANISM

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

A. PULL AND PUSH ACTION

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 action; 2) pull action.

Push action. In that mode, the logs are periodically pushed to the data owner by the harmonizer. The push action 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 action 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 action 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 action

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.

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.

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] B. Crispo and G. Ruffo, “Reasoning about Accountability within Delegation,” Proc. Third Int'l Conf. Information and

Comm. Security (ICICS), pp. 251-260, 2001.

[5] S. Sundareswaran, A. Squicciarini, D. Lin, and S. Huang, “Promoting Distributed Accountability in the Cloud,” *Proc. IEEE* Int’l Conf. Cloud Computing, 2011.

[6] 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.

[7] B. Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C. John Wiley & Sons, 1993.

[8] Praveen Gauravaram, John Kelesy, Lars Knudsen, and Soren Thomsen, “On Hash function using Checksums”

[9] 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

[10] 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.

[11] R. Corin, S. Etalle, J.I. den Hartog, G. Lenzini, and I. Staicu, "A Logic for Auditing Accountability in Decentralized Systems,"

[12] Eucalyptus Systems, http://www.eucalyptus.com/, 2013.

[13] Emulab Network Emulation Testbed, www.emulab.net, 2013.

[14] M. Mambo and E. Okamoto, "Proxy Cryptosystems: Delegation of the Power to Decrypt Ciphertexts," IEICE Trans. Fundamentals of Electronics, Comm. and Computer Sciences, vol. E80-A, no. 1, pp. 54- 63, 1997.

[15] M. Blaze, G. Bleumer, and M. Strauss, "Divertible Protocols and Atomic Proxy Cryptography," Proc. Int'l Conf. Theory and Application of Cryptographic Techniques (EUROCRYPT), pp. 127-144, 1998.

[16] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage," ACM Trans. Information and System Security, vol. 9, no. 1, pp. 1-30, 2006.

[17] Q. Tang, "Type-Based Proxy Re-Encryption and Its Construction," Proc. Ninth Int'l Conf. Cryptology in India: Progress in Cryptology (INDOCRYPT), pp. 130-144, 2008.

[18] G. Ateniese, K. Benson, and S. Hohenberger, "Key-Private Proxy Re-Encryption," Proc. Topics in Cryptology (CT-RSA), pp. 279-294, 2009.

[19] J. Shao and Z. Cao, "CCA-Secure Proxy Re-Encryption without Pairings," Proc. 12th Int'l Conf. Practice and Theory in Public Key Cryptography (PKC), pp. 357-376, 2009

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.