An Approach for Storage Security in Cloud Computing- A Survey

W. Sharon Inbarani, G. Shenbaga Moorthy, C. Kumar Charlie Paul

Abstract— **The many advantages of cloud computing are increasingly attracting individuals and organizations to outsource their data from local to remote cloud servers. In addition to cloud infrastructure and platform providers, such as Amazon, Google, and Microsoft, more and more cloud application providers are emerging which are dedicated to offering more accessible and user friendly data storage services to cloud customers. Storing data in a third party's cloud system causes serious concern over data confidentiality. General encryption schemes protect data confidentiality, but also limit the functionality of the storage system. We propose a threshold proxy re-encryption scheme and integrate it with decentralized erasure code such that a secure distributed storage system is formulated. The distributed storage system not only supports secure and robust data storage and retrieval, but also lets a user forward his data in the storage servers to another user without retrieving the data back.**

Key Terms—**Cloud Computing, Erasure Code, Homomorphism, Threshold Proxy Re-encryption.**

I. Introduction

The field of Information Technology is constantly evolving. There is no doubt about the big process of the internet, which is the main factor in IT world, especially with regard to speed in data transfer. Internet is becoming more and more important for nearly everybody as it is one of the newest and the medium of the future. Since the need for online services is increasing, the extent of services available through the internet, such as online software, storage, platform, file archives, on-demand self-service, ubiquitous network access, location independent resource pooling, etc., is also growing. This leads to formation of a structured provision of services, called cloud computing.

Cloud computing is the long dreamed vision of computing as a utility, where users can remotely store their data into the cloud so as to enjoy the on-demand high quality application and services from a shared pool of configurable computing resources. One of the main services in the cloud is the availability of online storage, called cloud storage. Data's are stored remotely in the cloud in a flexible on-demand manner. The benefits of cloud storage services are relief of the burden for storage management, universal data access with independent locations, and avoidance of capital expenditure on hardware, software, and personnel maintenance, etc [6]. By using the cloud storage services, users can share their data with each other, and perform their co-operative tasks together without the need of meeting each other so often.

Since the speed of data transfer over the internet is increasing, there is no problem in storing and sharing data in the cloud. The main concern in cloud computing is to provide a large amount of services in a virtualized manner in order to reduce the server sprawl, inefficiencies and high costs. So in cloud computing the servers that are used to provide services, among others cloud storage, are fully virtualized.

This virtualization mechanism makes it possible for cloud storage users to get the specific amount of storage that they need, and thus they are only required to pay for the used storage. Cloud services can be accessed via a web based user interface. One of the main advantages is elasticity. Users get the storage they need, and they only pay for their usage. By using cloud storages, small organizations save the complexity and cost of installing their own storage services.

A cloud storage system is considered as a large scale distributed storage system that consists of many independent storage servers [3]. In the process of storing data to the cloud, and retrieving data back from the cloud, there are mainly three elements that are involved, namely the user, the server and the communication between them. In order for the data to have the necessary security, all these elements must have a solid security.

For the user, it is mostly every user's responsibility to make sure that no unauthorized party can access his machine. On the server side, data must have confidentiality, data robustness, functionality, integrity and availability. The communication between client and server must be performed through a secure channel. Secure storage in cloud computing may be achieved through cryptographic access control.

I. Literature review

A. *Secure and dependable storage services in Cloud Computing*

Cloud storage enables users to remotely store their data and enjoy the on-demand high quality cloud applications without the burden of local hardware and software management. Though the benefits are clear, such a service is also relinquishing user's physical possession of their outsourced data, which inevitably poses new security risks towards the correctness of the data in cloud. This paper

W.Sharon Inbarani, *Department of Computer Science and Engineering, A.S.L Pauls College of Engineering and Technology, Coimbatore, India, 9952164567.*

G.Shenbaga Moorthy, *Department of Elecronics andCommunication Engineering, A.S.L Pauls College of Engineering and Technology, Coimbatore, India, 8012439223.*

Dr.Kumar Charlie Paul, *Principal, A.S.L Pauls College of Engineering and Technology, Coimbatore, India, 9443028493.*

proposes a distributed storage integrity auditing mechanism, utilizing the homomorphic token and distributed erasure-coded data [1]. The proposed design allows users to audit the cloud storage with very lightweight communication and computation cost. The auditing result not only ensures strong cloud storage correctness guarantee, but also simultaneously achieves fast data error localization i.e., the identification of misbehaving server.

Error localization is a key prerequisite for eliminating errors in storage systems. However, many previous schemes do not explicitly consider the problem of data error localization, thus only provide binary results for the storage verification.

A challenge-response protocol is achieved by integrating the correctness verification and error localization. The response values from servers for each challenge not only determine the correctness of the distributed storage, but also contain information to locate potential data error(s).

Advantages:

- Lightweight communication and computation cost.
- Data are stored redundantly across multiple physical servers.
- Erasure-Correcting code is used to provide redundancies and guarantees the data dependability against byzantine failures.
- Identifies misbehaving servers.

Disadvantages:

Light overhead

B. Privacy -Preserving Remote Data Integrity Checking Protocol with Data Dynamics and Public Verifiability

Storing data in the cloud has become a trend. An increasing number of clients store their important data in remote servers in the cloud, without leaving a copy in their local computers. Sometimes the data stored in the cloud is so important that the clients must ensure it is not lost or corrupted. While it is easy to check data integrity after completely downloading the data to be checked, downloading large amounts of data just for checking data integrity is a waste of communication bandwidth.

Hence, a lot of works have been done on designing remote data integrity checking protocols, which allow data integrity to be checked without completely downloading the data. In remote data integrity checking protocols, the client can challenge the server about the integrity of a certain data file, and the server generates responses proving that it has access to the complete and uncorrupted data [8].

The basic requirements are that the client does not need to access the complete original data file when performing the verification of data integrity, and that the client should be able to verify integrity for an unlimited number of times. Furthermore, the protocol needs to be secure against a malicious server that tries to pass the data integrity verification without access to the complete and uncorrupted data.

1. Data Dynamics

Data Dynamics means after clients store their data at the remote server, they can dynamically update their data at later times. At the block level, the main operations are block insertion, block modification and block deletion.

2. Public Verifiability

Each and every time the secret key sent to the client's email and can perform the integrity checking operation. This scheme consists of two entities: a challenger that stands for either the client or any third party verifier, and an adversary that stands for the untrusted server. Client doesn't ask any secret key from third party.

3. Privacy against Third Party Verifiers

Under the semi-honest model, a third party verifier cannot get any information about the client's data from the protocol execution. Hence, the protocol is private against third party verifiers. If the server modifies any part of the client's data, the client should be able to detect it; furthermore, any third Party verifier should also be able to detect it.

Advantages:

- Remote data integrity protocol is used for cloud storage which supports public verifiability and privacy against third party verifiers.
- Does not need third party auditor.
- Data level dynamics can be supported by using block level dynamics.
- In block level dynamics it has good efficiency in the aspect of communication, computation and storage cost.

Disadvantages:

- There is no clear mapping relationship between the data and the tags.
- Whenever a piece of data is modified, the corresponding blocks and tags are updated which leads to unnecessary communication and computation costs.

C. Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing

This paper proposed some services for data security and access control when users outsource sensitive data for sharing on cloud servers. This paper addresses this challenging open issue by, on one hand, defining and enforcing access policies based on data attributes, and on the other hand allowing the data owner to delegate most of the computation tasks involved in fine grained data access control to untrusted cloud servers without disclosing the underlying data contents.

This scheme enables the data owner to delegate tasks of data file re-encryption and user secret key update to cloud servers without disclosing data contents or user access privilege information. This goal can be achieved by exploiting and uniquely combining techniques of attribute-based encryption (ABE), proxy re-encryption, and lazy re-encryption [7].

This scheme has salient properties of user access privilege confidentiality and user secret key accountability and achieves fine – graininess, scalability and data confidentiality for data access control in cloud computing.

In order to achieve secure, scalable and fine-grained access control on outsourced data in the cloud three advanced cryptographic techniques are used.

- Key Policy Attribute-Based Encryption (KP-ABE).
- Proxy Re-Encryption (PRE).
- Lazy re-encryption.

1. Key Policy Attribute-Based Encryption (KP-ABE)

KP-ABE is a public key cryptography primitive for one-to-many communications. In KP-ABE, data are associated with attributes for each of which a public key component is defined. User secret key is defined to reflect the access structure so that the user is able to decrypt a cipher text if and only if the data attributes satisfy his access structure.

2. Proxy Re-Encryption (PRE)

Proxy Re-Encryption (PRE) is a cryptographic primitive in which a semi-trusted proxy is able to convert a cipher text encrypted under Alice's public key into another cipher text that can be opened by Bob's private key without seeing the underlying plaintext.

3. Lazy Re-Encryption

The lazy re-encryption technique and allow Cloud Servers to aggregate computation tasks of multiple operations. The operations such as

a. Update secret keys
b. Update user attributes

Advantages:

- Low initial capital investment.
- Shorter start-up time for new services.
- Lower maintenance and operation costs.
- Higher utilization through virtualization.
- Easier disaster recovery.

Disadvantages:

Heavy computation overhead if PRE is not combined with KP-ABE

D. Public Auditability and Data Dynamics for Storage Security in Cloud Computing

This paper proposes a TPA which eliminates the involvement of the client through the auditing of whether his data stored in the cloud is indeed intact. The support for data dynamics via the most general forms of data operation, such as block modification, insertion and deletion, is also a significant step toward practicality, since services in Cloud Computing are not limited to archive or backup data only.

While prior works on ensuring remote data integrity often lacks the support of either public auditability or dynamic data operations, this paper achieves both.

In particular to achieve efficient data dynamics, existing proof of storage models can be improved by manipulating block tag authentication by using Merkle Hash Tree construction for block tag authentication [9]. To support efficient handling of multiple auditing tasks, a multi-user setting is used in which TPA can perform multiple auditing tasks simultaneously.

1. Public auditability for storage correctness assurance

To allow anyone, the clients who originally stored the file on cloud servers, to have the capability to verify the correctness of the stored data on demand.

2. Dynamic data operation support

Dynamic data operations allow the clients to perform block-level operations on the data files while maintaining the same level of data correctness assurance. The design should be as efficient as possible so as to ensure the seamless integration of public auditability and dynamic data operation support.

3. Blockless Verification

No challenged file blocks should be retrieved by the verifier (*e.g.* TPA) during verification process for efficiency concern.

4. Batch Auditing for Multi-client Data

As cloud servers may concurrently handle multiple verification sessions from different clients, given K signatures on K distinct data files from K clients, it is more advantageous to aggregate all these signatures into a single short one and verify it at one time.

The signature scheme allows the creation of signatures on arbitrary distinct messages. Moreover, it supports the aggregation of multiple signatures by distinct signers on distinct messages into a single short signature, and thus greatly reduces the communication cost while providing efficient verification for the authenticity of all messages.

Advantages:

- Public auditing system of data storage security in cloud computing supports fully dynamic data operations.
- Block insertion is possible which is missing in existing schemes.
- Scalable and efficient.

Disadvantages:

Secure data forwarding is not possible.

E. Ensuring Data Storage Security in Cloud Computing

In this paper, a problem of data security in cloud data storage is investigated. To ensure the correctness of user's data in cloud data storage, an effective and flexible distributed scheme with explicit dynamic data support, including block update, delete, and append.

Erasure - correcting code is used in file distribution preparation to provide redundancy parity vectors and guarantee the data dependability. By utilizing the homomorphic token with distributed verification of erasure coded data, this scheme achieves the integration of storage correctness insurance and data error localization [2].

Challenge Response protocol is used to provide the localization of data error. In cloud data storage, a user stores the data through a CSP into a set of cloud servers, which are running in a simultaneous, cooperated and distributed manner. Data redundancy can be employed with technique of erasure-correcting code to tolerate faults or server crash.

Advantages:

- Localization of data error.
- Resilient against Byzantine failures.
- Scalable and efficient.

Disadvantages:

It is crucial to consider the dynamic case, where a user may wish to perform various block-level operations of update, delete and append to modify the data file while maintaining the storage correctness.

F. An Efficient Remote Data Possession Checking in Cloud Storage

As cloud storage can achieve the goal that getting all storage resources in a plug and play way, it becomes a focus of attention. When users store their data in cloud storage, they mostly concern about whether the data is intact. This is the goal of remote data possession checking (RDPC) schemes. This paper proposes an efficient RDPC scheme.

It has several advantages as follows. First, it is efficient in terms of computation and communication. Second, it allows verification without the need for the challenger to compare against the original data. Third, it uses only small challenges and responses, and users need to store only two secret keys and several random numbers. Finally a challenge updating method is proposed based on Euler's theorem [4].

Remote data possession checking is a topic that focuses on how to frequently, efficiently and securely verify that a storage server can faithfully store its client's original data without retrieving it. The storage server is assumed to be un-trusted in terms of both security and reliability.

There are two types of schemes, namely provable data possession (PDP) and proof of retrievability (POR). The difference between PDP and POR is that POR checks the possession of data and it can recover data in case of a failure. Usually a PDP can be transformed to a POR by adding erasure or error correcting codes.

Advantages:

- Efficient in terms of computation and communication.
- This scheme allows verification without the need for the challenger to compare against the original data.
- Challenging updating method is used based on Euler's theorem.

Disadvantages:

- Number of verification is limited.
- No data updating.

II. EXISTING SYSTEM

Data security, which has always been an important aspect of quality of service, Cloud Computing inevitably, poses new challenging security threats for number of reasons. Data are stored in a single server. The data's are directly stored in a cloud server in which no encryption and decryption technique is used for storing. Users can not access the cloud without the knowledge of cloud requester. For auditing purpose, cloud requesters give the local copy of data to auditor.

Traditional cryptographic primitives for the purpose of data security protection cannot be directly adopted due to the user's loss control of data under Cloud Computing. Therefore verification of correct data storage in the cloud must be conducted without explicit knowledge of the whole data. Considering various kinds of data for each user stored in the cloud and the demand of long term continuous assurance of their data safety, the problem of verifying correctness of data storage in the cloud becomes even more challenging.

There are three problems in the existing system.

1. The user has to do most computation and the communication traffic between the user and storage servers is high.
2. The user has to manage his cryptographic keys. If the user's device of storing the keys is lost or compromised, the security is broken.
3. Data storing and retrieving, it is hard for storage servers to directly support other functions. For example, storage servers cannot directly forward a user's messages to another one. The owner of messages has to retrieve, decode, decrypt and then forward them to another user.

III. PROPOSED SYSTEM

In proposed scheme a new threshold proxy re-encryption scheme and integrate it with a secure decentralized code to form a secure distributed storage system. The encryption scheme supports encoding operations over encrypted messages and forwarding operations over encrypted and encoded messages. The tight integration of encoding, encryption, and forwarding makes the storage system efficiently meet the requirements of data robustness, data confidentiality and data forwarding. Accomplishing the integration with consideration of a distributed structure is challenging.

This system meets the requirements that storage servers independently perform encoding and re-encryption and key servers independently perform partial decryption. The advantages of proposed system are:

1. Comparing the existing system it reduces the maintenance cost and investment.
2. Prevent errors due to systematic process.
3. All the data which is being stored in the cloud database in encrypt format.
4. Security is high.
5. Extensive security and performance analysis shows that the proposed scheme is highly efficient and resilient against Byzantine failure, malicious data modification attack, and even server colluding attacks.
6. Data's are forwarded directly to another user by storage server directly under the command of the data owner.

IV. ENSURING CLOUD DATA STORAGE

A secure cloud storage system is constructed, which supports the function of secure data forwarding by using a threshold proxy re-encryption scheme. Our system is highly distributed where storage servers independently encode and forward messages and key servers independently perform partial decryption.

A. System Model

As shown in Fig. 1, our system model consists of users, n distributed storage servers SS_1, SS_2, ….SS_n , and m key servers KS_1, KS_2, …. , KS_m. The purpose of distributed storage servers is to store data reliably over a long period of time by using a distributed accumulation of storage servers [5].

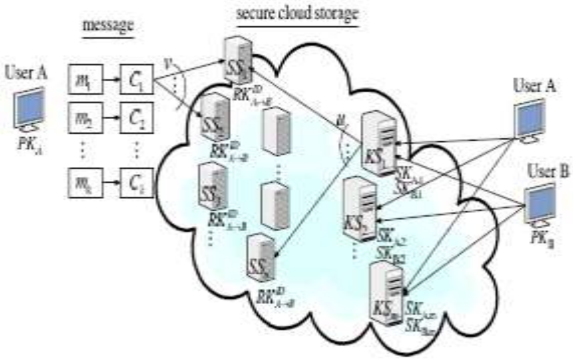

Fig. 1 A General System Model

A decentralized architecture for storage systems offers good scalability, because a storage server can join or leave without control of a central authority. Key servers provide key management services. In cryptography, the security of a system lies on protection of the secret key.

The storage servers and key servers are two different types of servers. The keys servers are much more secure and their number is much less. The storage servers provide large capacity of storage, while they are prone to attacks.

A user distributes the cryptographic key to key servers, which performs cryptographic functions on behalf of the user. These key servers are highly protected by security mechanisms.

A new threshold proxy re-encryption scheme has been proposed such that it will be integrated with a secure decentralized code to form a secure distributed storage system. The encryption scheme supports encoding operations over encrypted messages and forwarding operations over encrypted and encoded messages.

The tight integration of encoding, encryption, and forwarding makes the storage system efficiently meet the requirements of data robustness, data confidentiality, and data forwarding. A secure cloud storage system supports the function of secure data forwarding by using a threshold proxy re-encryption scheme. The encryption scheme supports decentralized erasure codes over encrypted and encoded messages.

The proposed system is highly distributed where storage servers independently encode and forward messages and key servers independently perform partial decryption.

B. *File Rifting*

File Rifting is a technique which is used to achieve data exchange for cloud computing. The prevention of file from intruders plays a vital role. The fragmentation makes a bit higher complexity to gain complete information of a file. Each fragmented file is sent over the network for Information Exchange. A source file is selected and number of output files to be splitted is selected. The content present in the source file is read and splitted according to the output files selected.

Data fragmentation can be achieved by erasure coding technique. To ensure data reliability in distributed storage systems, various data redundancy techniques can be employed, such as replication, erasure codes, and network coding. An erasure code provides redundancy by breaking objects or data files into smaller fragments and fragments are stored in distributed cloud storage system. The key is that, the data can be recovered from any combination of a smaller number of fragments.

An erasure code provides redundancy without the overhead of strict replication. Erasure codes divide an object into *m* fragments and recode them into *n* fragments, where *n*>m. The rate of encoding *r=m/n <1* .The key property of erasure codes is that the original object can be reconstructed from any *m* fragments. For example, using an *r* = ¼ encoding on a block divides the block into *m*=16 fragments and encodes the original *m* fragments into *n*=64 fragments.

A decentralized erasure code is suitable for use in a distributed storage system. After the message symbols are sent to storage servers, each storage server independently computes a code-word symbols and stores it. This finishes encoding and storing process.

C. *Homomorphic Encryption*

In order to preserve privacy, the clients will encrypt their data when they out-source it to the cloud. However, the encrypted form of data greatly impedes the utilization due to its randomness. Many efforts have been done for the purpose of data usage but without undermining the data privacy.

To overcome this security crypto process is implemented. Multiple data is chosen at this phase and numbers of files are counted. Then secret keys are created (Public Key, Private key) for Authentication. Then the keys are assigned to the files. Files are encrypted by means of corresponding keys. Public key is used to encrypting and private key is used for decrypting. The system manager chooses system parameters and publishes them. Each user A is assigned a public-secret key pair (PK_A, SK_A). User A distributes his secret key SK_A to key servers such that each key server KS_i holds a key share SK_A, i, $1 \leq i \leq m$. The key is shared with a threshold t.

The main aim of this scheme is to encrypt data before sending it to the cloud provider, but to execute the calculations the data should be decrypted every time we need to work on it. Until now it was impossible to encrypt data and to trust a third party to keep them safe and able to perform distant calculations on them. So to allow the Cloud provider to perform the operations on encrypted data without decrypting them requires using the cryptosystems based on homomorphic encryption.

Homomorphic Encryption systems are used to perform operations on encrypted data without knowing the private key (without decryption), the client is the only holder of the secret key. When the result of any operation is decrypted the result of any operation is the same as we had carried out the calculation on the raw data.

D. *Threshold Proxy Re-encryption*

A fundamental approach of threshold PRE scheme is for secure computation. This scheme performs arbitrary computations on encrypted data without decrypting it. Threshold PRE technique has multiplicative homomorphic property. A multiplicative homomorphic encryption scheme supports the encoding operation over encrypted messages and forwarding operations over encrypted and encoded messages.

The three properties exhibited by Threshold PRE scheme are :

Homomorphism : Given two ciphertexts c1 and c2 on plaintexts p1 and p2 respectively, one can obtain the ciphertext on the plaintext p1+p2 and/or p1.p2 by evaluating c1 and c2 without decrypting ciphertexts.

Proxy re-encryption: Transforming encrypted data of one user to encrypted data of target user.

Threshold decryption: By dividing the private key into several pieces of secret shares, all clients can work together to decrypt the ciphertext – the output of the function.

E. *Cloud Data Storage*

Data outsourcing to cloud storage servers is raising trend among many firms and users owing to its economic advantage. This essentially means the owner of the data (client) moves its data to a third party cloud storage server. Cloud storage is an online virtual distributed storage provided by cloud computing vendors.

Cloud storage services can be accessed via a web service interface, or a web based user-interface. One of the advantages is its elasticity. Customers get the storage they need, and they only pay for their usage. By using cloud storages, small organizations save the complexity and cost of installing their own storage devices. The same as cloud computing, cloud storage has also the properties of being agile, scalable, elastic and multi-tenant.

User A encrypts his message M and dispatches it to storage servers. A message M is decomposed in to *k* blocks $m_1, m_2, ...m_k$ and has an identifier ID. User A encrypts each block m_i into a cipher text C_i and sends it to *v* randomly chosen storage servers. Upon receiving cipher texts from a user, each storage server linearly combines them with randomly chosen coefficients into a codeword symbol and stores it. The user stores the encrypted files in a different location in a cloud server. The requester has the corresponding keys. The numbers of storage servers are determined by the user's requirement.

F. Data Retrieval

When the user needs back the file which is in cloud server the user can retrieve the re-encrypted files from multiple servers. The user decrypts the files individually by referring the keys in key servers. Decrypted individual files are then merged to get the whole file.

User A request to retrieve a message from storage servers. The message is either stored by him or forwarded to him. User A sends retrieval request to key servers. Upon receiving the retrieval request and executing a proper authentication process with user A , each key sever KS_i request *u* randomly chosen storage servers to get codeword symbols and does partial decryption on the received codeword symbols by using the key share $SK_{A,i}$. Finally, user A combines the partially decrypted codeword symbols to obtain the original message M.

G. Data Forwarding

A user forward the data in the storage servers to another user without retrieving the data back. User A forwards his encrypted message with an identifier ID stored in storage servers to user B such that B can decrypt the forwarded message by his secret key. User A uses his secret key SK_A and B's public key PK_B to compute a re-encryption key $RK^{ID}_{A\text{->}B}$ and then sends $RK^{ID}_{A\text{->}B}$ to all storage servers.

Each storage server uses the re-encryption to re-encrypt its codeword symbol for later retrieval request by B. The re-encrypted codeword symbol is the combination of cipher texts under B's public key.

V. CONCLUSION

In this paper, we consider a cloud storage system consists of cloud servers and key servers. We integrate a new threshold proxy re-encryption scheme. The threshold proxy re-encryption scheme supports encoding, encryption, and secures data forwarding operation in a distributed way. We present a secure cloud storage system that provides secure data storage and secure data forwarding functionality. In future, we can develop more efficient encryption techniques which reduce the time needed for encryption and decryption.

REFERENCES

[1] Cong Wang, Kui Ren, Qian Wang and Wenjing Lou "Towards Secure and Dependable Storage Services in Cloud Computing," IEEE transactions on Services Computing, vol 5, no. 3, pp 220-232, 2011.

[2] C.Wang, Qian Wang, Kui Ren, Wenjing Lou, "Ensuring Data Storage Security in Cloud Computing," Proc. IWQoS „09, pp. 1–9, July 2009.

[3] Hsiao-Ying Lin and Wen-Guey Tzeng "A Secure Erasure Code-Based Cloud Storage System with Secure Data Forwarding," IEEE Trans.Parallel and Distributed Systems, vol 23, no. 6, pp. 995-1003, June 2012.

[4] L. Chen, Gongde Guo "An Efficient Remote Data Possession Checking in Cloud Storage", Fujian Normal University, vol. 5, no. 4, April 2011.

[5] Lin H.Y. and Tzeng W.G. "A secure decentralized erasure code for distributed network storage," IEEE Transactions on Parallel and Distributed Systems, vol. 21, pp. 1586–1594, November 2010.

[6] M.Armbrust, A.Fox, R.Griffith, A.D. Joseph, R.H.Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, and M.Zaharia, " A Berkeley view of Cloud Computing, " University of California, Berkeley, Tech. Rep. UCB-EECS-2009-28, Feb 2009.

[7] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in IEEE INFOCOM'10, 2010.

[8] Sheng Zhong and Zhuo Hao. "A Privacy-Preserving Remote Data Integrity Checking Protocol with Data Dynamics and Public Verifiability," IEEE Internet Computing, 2010

[9] Qian Wang, Cong Wang, Kui Ren, Wenjing Lou, and Jin Li, "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing" , IEEE Transactions on Parallel & Distributed Systems, Volume: 22, Issue: 5, pages: 847-859.

W.Sharon Inbarani ., is pursuing her M.E (CSE) in A.S.L Pauls College of Engineering and Technology, Coimbatore, India. Her research area is Cloud Computing, Network Security.

G. Shenbagamoorthy., is working as a Assistant Professor in A.S.L Pauls College of Engineering and Technology. He has completed his M.Tech degree in CSE. His research interests include Cloud Computing, Network Security.

Dr. Kumar Charlie Paul., is working as a principal in A.S.L Pauls college of Engineering. He has published various papers in international conferences and reputed journals. His research interest includes Networking.