An Inclusive Perspective for Threshold Proxy Re-Encryption Scheme to Ensure Data Confidentiality

Shruthi R
Department of Computer Science and Engineering
Channabasaveshwara Institute of Technology
Gubbi, Karnataka, India.

Prof Nagaraja P
Department of Information Science and Engineering
Channabasaveshwara Institute of Technology
Gubbi, Karnataka, India.

*Abstract:***A cloud storage system is composed of a collection of data storage servers, which helps in providing durable storage services across the Internet. Data Confidentiality, being the serious concern for storing the data in distributed cloud storage servers. Generally Encryption schemes support data confidentiality but also restrict some functionality of the storage system because only few operations are supported over encrypted data. Thus building a secure storage system that supports multiple functions is a major challenge since the storage servers are distributed and lack with the central power. Therefore proposed a threshold proxy re-encryption (PRE) scheme and combine it with a decentralized erasure coding method that helps in forming a secure distributed storage system. This distributed storage system guarantees a secure cloud and supports robust data storage and retrieval, and also lets a user in forwarding his data in the distributed storage servers to another user without regaining the data back. The main technological contribution is that the PRE scheme supports encoding operations over encrypted messages as well as forwarding operations over encoded and encrypted messages. Our technique entirely integrates encrypting, encoding, and forwarding. We explore and propose appropriate parameters for the number of copies of a message sent to the storage servers and key server queries the number of storage servers. These parameters let more bendy adjustment between the number of storage servers and robustness.**

*Keywords:***Secure storage system, Encoding, Encryption, Threshold Cryptography, Proxy Re-Encryption, Decentralized erasure code.**

I. INTRODUCTION

Nowadays cloud computing is extremely popular and it is still developing standard. It is strong rising area in a computing study and industry in present days. Cloud computing is on demand provision in which information, shared resources, software and all other devices are agreed accordingly to client needs for specific time. Cloud is a word that is used usually in case of internet. In this location users need not to possess the infrastructure for different computing services, rather users can access their data from any computers and also from any part of world.

Cloud computing are pretty advantageous and are accessible to the end user because of the services and models. The two available models are: Deployment model and service model. Further the deployment model has three divisions: public, private, and hybrid cloud. Private cloud is also known as internal cloud or corporate cloud which represents the business network. This cloud is accessed in an organization that requires more control over their data compared to the data that is provided by the third party's cloud. Public cloud makes the services available to anyone across the internet. In this public cloud industry pays for the capability and also the users rent for what they access. Hybrid cloud is the union of private and public cloud.

Cloud service has three categorised models for provisioning service. They are: Infrastructure-as-a-service (Iaas), Platform-as-a-service (Paas), and Software-as-a-service (Saas). In Iaas where there is only the network provided and in Paas both the network and the operating system is provided and in Saas operating System the necessary software and network is provided by cloud service and is popular as it decreases the difficulty of network, users do not need to buy software licenses and information in cloud.

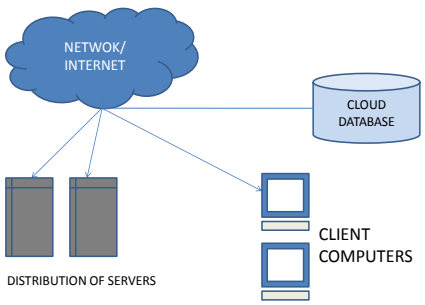


Fig 1: Cloud Computing Architecture

Topologically, cloud computing refers to the combination of several elements: Clients, the cloud database and the distributed servers. As shown above these three elements form a cloud computing solution. Each element has its own specific functional applications. In general clients are the devices that help for the end user to access the data across internet from cloud and are based on how they interact and manage with the cloud for information. Cloud database is house information which is regarded as a collection of servers that is meant as a datacentre in which the application is provisioned when subscribed by the end

ISSN: 2278 – 1323

user. Distributed Servers don't need to be located in the same location, when subscribed for information by end user as the name specified there is no central authority presence and are geographically distributed in which each server plays an important role in security as well as being robust.

Cloud computing is an idea that considers the resources on the Internet as a collective unit, a cloud. The high-speed networks and Internet access become accessible in current years, several services are available on the Internet so that users can access them from anywhere at any point of time. For instance, the most accepted one is an email service. Users just use the services provided, without the aware of how the storage of data is managed and how far it is secure. Thus this paper mainly marks on the design of a cloud storage system for maintaining robustness, confidentiality, and functionality.

A cloud storage system consists of many independent storage servers that are measured as a large-scale distributed storage system. Data robustness is an important requirement for storage systems. There have been many approaches for data storage in distributed storage servers [1] [2]. One way to offer data robustness is to duplicate a message so that each storage server stores a copy of the message. This maintains robustness of data as the message can be regained as long as one storage server survives [3]. The other way is to adopt erasure coding method which encodes a message of k symbols to an automatically generated code word of n symbols to ensure data confidentiality. For a message to store, each of its generated code word symbols is to be stored across different storage server. An erasure error of the code word symbol occurs due to the storage server breakdown. The number of crashed servers below the tolerance threshold level of the erasure code can be recovered as long as the message with the generated code word symbols are stored in the accessible storage servers through the decoding procedure. Thus provides an exchange between the storage size and the tolerance threshold of crashed servers. A decentralized erasure code is an erasure code method that independently evaluates each code word symbol per message. Therefore, the encoding process for each message is split into n parallel tasks of generating code word symbols. This decentralized erasure coding method is appropriate for use in a distributed storage system. Each message symbols sent to the distributed storage server is received and independently computes a code word symbol and stores in it. This completes the encoding process and storing process. Recovery process is also the same as above. Data confidentiality is a serious issue when the data is stored in third party storage server. For a strong confidentiality of messages in the storage servers, user encrypts messages using cryptography method before the erasure method is used. When the user wishes to use a message, he needs to regain the code word symbols from storage servers, decode the message, and then decrypt the message using cryptographic keys. This raises three issues in the mentioned integration of encryption and encoding. First, the user needs to do the majority calculation and thus the communication traffic between the user and storage servers becomes high. Second, the user needs to manage his cryptographic keys. If there is any lost of user's device key for storing data, the security is broken down. Finally, data storage and retrieving in further becomes hard for storage servers to support for other functions. For instance, storage servers cannot straightforward a user's messages to another storage server. The owner of the messages has to retrieve message, decode them, decrypt the message and then forward them to another user. Thus in this paper, we deal with the difficulty of forwarding the data to another user by storage servers directly based on the command passed by the data owner. The system model has distributed storage servers and key servers built. As storage of cryptographic keys within a single device is quite risky, a user distributes his cryptographic key to key servers in a cloud that perform cryptographic functions on account of the user. These key servers are greatly secured by high security technique. To fine fit the distributed construction of systems, we need the servers that independently perform all operations. Through this consideration, we put forward a new threshold proxy re-encryption scheme and combine it with a protected decentralized code to outline a secure distributed storage system. The encryption scheme supports encoding operations over encrypted messages and forwarding operations over encrypted and encoded messages. The firm integration of encoding, encryption, and forwarding makes the storage system competently meet the requirements of data robustness, data confidentiality, and data forwarding. Accomplishment of integration with the consideration of a distributed formation is challenging. Our scheme meets the necessities that storage servers independently execute encoding and re-encryption and key servers in parallel carry out partial decryption. In addition, we regard the system in a more general setting than the previous mechanism. This setting allows more bendy adjustment between the number of storage servers and robustness.

II.REALTED WORK

A. What is Proxy Re-Encryption

As discussed in the previous section, the proxy re-encryption refers to the process of re-encrypting the message. The owner first encrypts the message using his cryptographic keysand stores in the distributed cloud servers. When a user receives a request or need to share his messages, he sends his cryptographic key to re-encrypt to the key servers in the cloud storage system. Key servers re-encrypt the message on behalf of the authorized user and forwards to the other user.

B. Need ofProxy Re-Encryption

The user or the owner who wants to receive the message cannot wait until the user forwards the message. This leads to the delay and increase in cost. Thus a proxy server in the cloud re-encrypts and forwards the data to the other user. This makes the data forwarding easy and needful for the user.

C. Need of Data Confidentiality

For an end user to trust on the cloud is an important issue for the purpose security. Consequently the third party or a cloud service provider needs to ensure the security for the data stored by the user. This is guaranteed using the Decentralized Erasure Code for encoding and Proxy Re-Encryption for cryptographic function. Therefore maintaining the data is a major concern and preventing the hacking of data from an unauthorized user.

ISSN: 2278 – 1323

III.PROPOSED SYSTEM

To deal with the problem of forwarding the data to the other user, we have a proposed scheme representation that provides a solution for efficient forwarding of data, storage of data without regaining the data back and also ensuring the data confidentiality. We build a secure storage system and represent the model with distributed servers and key servers to handle cryptographic functions. Storing the keys in single device is quite risky and thus the user manages to distribute his keys to the key servers to act on behalf of him. Keys handled by key servers are highly maintained through an effective security mechanism. For this purpose, an effective threshold proxy re-encryption scheme is proposed and integrated with a decentralized erasure coding method and a Homomorphic Encryption. This encryption allows the encoding method over the encrypted messages and also to forward the data across both the encoded and encrypted messages.

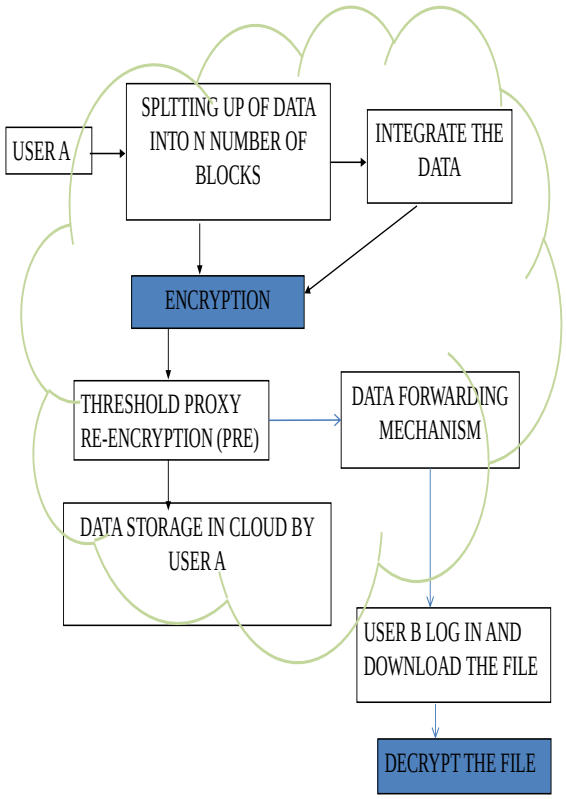


Figure2: Proposed System Architecture

IV. IMPLEMENTATION

A. System Model

Consists of the following and explained in detail.

- **Decentralized Erasure Code:** This phase is mainly used for the splitting reason and encoding the data. The method splits the data or the messages into blocks. Suppose there are n number of blocks say $n=ak^c$, where the number of storage servers are greater than the number of divided blocks. This is considered as the initiation of the project. This computes code word symbol for each message independently. Therefore encoding task is parallel done for generating symbols. The message is later stored for integration procedure.
- **Integration:**In this process the split message is joined into s number of blocks and is stored in the huge storage server.
- **Encryption:** in this, User A encrypts his message M, ie., he encrypts each message from plain text to cipher text with an unique identifier mi to ci and sends for storage. For encrypting, random key generation method is used in which every session random keys are generated. As data confidentiality is seriously concerned, the data is encrypted twice using proxy re-encryption method and secured because of random key generation technique. For more than 10,000 key is generated per session.
- **Data Forwarding:** In order to forward the data by User A, user sends his encrypted message with an unique identifier (ID) to the storage servers. In order to decrypt User B has secret key. To carry out this, A has its secret key SA_k and B's Public key PB_k to compute for re-encryption key as $RK^{ID}_{A->B}$ and this key $RK^{ID}_{A->B}$ is being sent to all the key servers by user.
- **Login:** It's a login page where a user can login and can access the data and services. If he has a account he can directly go to login process as an authorized user or else he can create his new account.
- **Uploading File:** User can upload his/her data through his account using Id and IP address. User logins and stores the encrypted data and sends a key for re-encryption to all key servers which act as proxy in the scheme.
- **Data Retrieval:** The data is retrieved by either of the users. If user A needs to retrieve the data through the recovery process it is made possible after the authentication confirmation. In case of user B he receives a forwarded message by downloading and decrypting as an authorized user.

B. Methodology

MULTIPLICATIVE HOMOMORPHISM:This encryption allows the encoding method over the encrypted messages. Then it is converted into proxy re-encryption scheme with Homomorphic property into the threshold version. The threshold value t is known by the key servers.For decrypting a set of c message symbols, every key server separately inquiry 2 storage servers and partly decrypts for two encrypted code word symbol. k code word symbols are gained from the partly decrypted cipher texts as long as t key servers are accessible.

C. Algorithms:

The combination of algorithms used for data security is:

a) **DES Algorithm:** Following are the steps:

1. Fragmentation of the text into 64-bit i.e., 8 into octet blocks;
2. Beginning with permutation of blocks;
3. Division of the blocks into two parts: left and right, as *L* and *R*;
4. Permutation and substitution of steps repeated for 16 times as **rounds**;

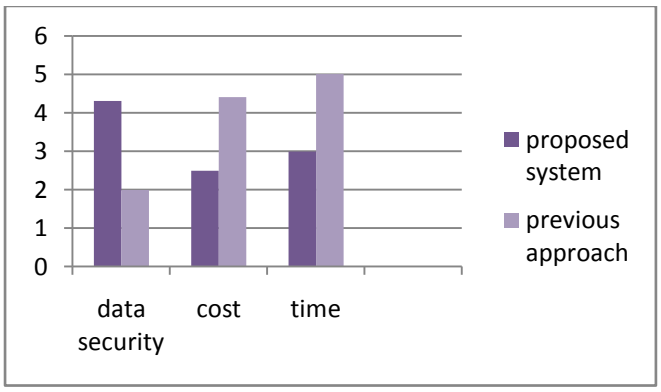
ISSN: 2278 – 1323

5. Again Re-join the left and right parts and then inverse initial permutation.

b) **RSA Algorithm:**

1. Prefer two distinctive prime numbers *p* and *q*.
   - For safety purposes, the integers *p* and *q*chosen should be random, and should be of comparable bit-length. Prime integers can be capably found by a primality test.
2. Calculate*n* = *pq*.
   - *n* is hold as the modulus for both the public and keys private keys. The key length is usually expressed in termsof bits, is the key length.
3. Calculate $\varphi(n) = \varphi(p)\varphi(q) = (p-1)(q-1) = n - (p+q-1)$, where φ refers to Euler's totient function. This obtained value is kept secret.
4. Select an integer *e* such that $1 < e < \varphi(n)$ and $\gcd(e, \varphi(n)) = 1$; i.e., *e* and φ(*n*) be co prime.
   - *e* is out as the public key exponent.
5. Find out *d*, $d \equiv e^{-1} \pmod{\varphi(n)}$; where *d* is the modular multiplicative inverse of *e* (modulo φ(*n*)).
   - This is more evidently stated as: explain for *d* known $d \cdot e \equiv 1 \pmod{\varphi(n)}$
   - This is repeatedly computed using the extended Euclidean algorithm. By this obtainedpseudo code in the *Modular integers* part, inputs *a* and *n* in turn correspond to *e* and *φ(*n*)*, respectively.
   - *d* is kept as the private key exponent.
6. Encryption
   $c = m^e \pmod{n}$
   Where c is computed cipher text
   m is a integer turned out from a message M
7. Decryption
   $m = c^d \pmod{n}$
   The original message M is recovered by decrypting using the above formula using the private key.

V. RESULT ANALYSIS



The above experimental result shows that practically the data can be secured and data forwarding is made easy. The Empirical result depicts that the data security is more comparatively and also there is reduction in cost and time parameters.

VI. EXPERIMENTAL RESULTS

Table 1: Analyzed Result

PARAMETERS	PROPOSED SYSTEM	PREVIOUS APPROACH
DATA CONFIDENTIALITY	4.3	2
COST	2.5	4.4
TIME	3	5

VII. CONCLUSION AND FUTURE WOK

In this paper, we have considered the secure cloud storage system that consists of both the storage servers and the key servers. We integrate this with a newly proposed threshold proxy re-encryption scheme (PRE) and decentralized erasure coding method. The threshold PRE scheme supports encoding the data, forwarding the data, and partial decryption of data in a distributed method. To carry out decryption of a message consisting of k blocks, which are encrypted and encoded to n codeword symbols, each key server only has to partially decrypt two codeword symbols in our system. By using this threshold proxy re-encryption scheme, we present a secure cloud storage system that provides secure data storage and secure data forwarding functionality in a decentralized structure. Furthermore, every storage server separately performs encoding of data and re-encryption of message and each key server alone perform partial decryption. Our proposed storage servers operate as storage nodes in a content addressable storage system for storing content addressable blocks. Our key servers perform as access nodes to provide a front-end layer like a traditional file system interface. In addition, study on thorough collaboration is required.

VIII. REFERENCES

[1]. S.Amritha, Mrs. Sarvana Kumar, "Threshold Proxy Re-Encryption Scheme and Decentralised code in cloud storage with secure data forwarding" IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 9, Issue 5 (Mar. - Apr. 2013), PP 27-31.

[2]. N.Jenefa, J. Jayalakshmi, "A Cloud Storage System with Data Confidentiality and Data Forwarding" International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-3, Issue-1, March-2013.

[3]. Giuseppe Ateniese† Kevin Fu‡ Matthew Green† Susan Hohenberger, "Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage".

[4]. C. Kavinilavu, "A Protected Cloud Storage System with Safe Information Forwarding", International Journal Of Engineering Research & Technology (IJERT) ISSN: 2278-0181 Vol. 2 Issue 1, January- 2013.

[5] Hsiao-Ying Lin, Member, IEEE, and Wen-Guey Tzeng, Member "A Secure Erasure Code-Based Cloud Storage System with Secure Data Forwarding"IEEE Transactions On Parallel And Distributed Systems, VOL. 23, NO. X, XXX 2012.

ISSN: 2278 – 1323

[6] J. Kubiatowicz, D. Bindel, Y. Chen, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao, "Oceanstore: An Architecture for Global-Scale Persistent Storage," Proc. Ninth Int'l Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS), pp. 190- 201, 2000.

[7] Adya, W.J. Bolosky, M. Castro, G. Cermak, R. Chaiken, J.R.Douceur, J. Howell, J.R. Lorch, M. Theimer, and R. Wattenhofer, "Farsite: Federated, Available, and Reliable Storage for an Incompletely Trusted Environment," Proc. Fifth Symp. Operating System Design and Implementation (OSDI), pp. 1-14, 2002.

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

[9] Blaze.M, 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.

[10] J. Kubiatowicz, D. Bindel, Y. Chen, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao, "Oceanstore: An Architecture for Global-Scale Persistent Storage," Proc. Ninth Int'l Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS), pp. 190- 201, 2000.

[11] P. Druschel and A. Rowstron, "PAST: A Large-Scale, Persistent Peer-to-Peer Storage Utility," Proc. Eighth Workshop Hot Topics in Operating System (HotOS VIII), pp. 75-80, 2001.

[12] A. Adya, W.J. Bolosky, M. Castro, G. Cermak, R. Chaiken, J.R. Douceur, J. Howell, J.R. Lorch, M. Theimer, and R. Wattenhofer, "Farsite: Federated, Available, and Reliable Storage for an Incompletely Trusted Environment," Proc. Fifth Symp. Operating System Design and Implementation (OSDI), pp. 1-14, 2002.

[13] A. Haeberlen, A. Mislove, and P. Druschel, "Glacier: Highly Durable, Decentralized Storage Despite Massive Correlated Failures," Proc. Second Symp. Networked Systems Design and Implementation (NSDI), pp. 143-158, 2005.

[14] R. Perlman, "File System Design with Assured Delete," Proc. Network and Distributed System Security Symp. ISOC (NDSS), 2007.

[15] R. Geambasu, T. Kohno, A. Levy, and H.M. Levy, "Vanish: Increasing Data Privacy with Self-Destructing Data," Proc. 18th Conf. USENIX Security Symp, Aug. 2009.

First Author: Shruthi R was born in Karnataka, India. She received the B.E Degree in Information Science and Engineering from Visvesvaraya Technological University Belgaum Dist., India in 2013 and pursuing Mtech Degree in Computer Science and Engineering from the same University. Her research interests are in the area of cloud computing.

Second Author: Nagaraja P M-tech, was born in Karnataka, India. He is working as Associate Prof. and HOD in Information Science Engineering Dept., CIT, Gubbi, Tumkur (district), and Karnataka, INDIA.

ISSN: 2278 – 1323