© 2016 IJSRSET | Volume 2 | Issue 1 | Print ISSN : 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

Custom Based Protocol for Information Retrieval in Cloud Computing

P. Gowri , K. Sangeetha
Department of Information Technology, SNS College of Technology, Coimbatore, Tamil Nadu, India

ABSTRACT

Prominent issue in cloud computing is security in which storing and retrieving the data in a third party's cloud system and public auditing scheme which causes serious problems and conflict over data confidentiality during the transaction of data. Though there are number of methods are available to overcome this problem like cryptography, key encryption etc. The general encryption schemes protect data confidentiality during the transaction; along with this process but the main drawback is its limits the functionality of the storage system because of only few operations are supported over encrypted data. This project proposes a secured threshold proxy re-encryption server and integrates it along with a decentralized erasure code which provides a secure distributed storage system for processing big data in which multiple users can interact with the storage system. In this environment users can upload their data in to the distributed storage system which will be stored in database in the encrypted format. User in the cloud will authorize the sender request to generate the key with the authorized one time key sender can access the encrypted file in decrypted format at once. The key becomes invalid after one use. The proxy server provides security for big data in cloud with proxy re encryption method which process effective way.

Keywords : Proxy Server, Re-Encryption Key, Cloud Storage.

I. INTRODUCTION

Cloud systems [2, 3] are used to enable data sharing capabilities but secured data forwarding of data is an important issue in cloud environment. Many privacy and security attacks occur within the Cloud provider themselves [4, 5] as they usually have direct access to stored data and steal the data and sell it to third parties in order to gain profit. This can be avoided by creating a proxy re-encryption code which provides secure data transaction. Proxy re-encryption scheme [6] provides security improvements over other approaches that are used earlier. The main advantage is that proxy servers are unidirectional and will not reveal the secret. So, in the can be using cloud environment through proxy re encryption method with virtual proxy server for secured data forwarding from cloud server. The main objective of this project is to develop a cloud architecture using privacy preservation protocol, Shared authority based data forwarding can be done through proxy server with Proxy re- encryption method.

During the data accessing process different users may be in a collaborative relationship, thus data sharing becomes significant to achieve productive benefits. A cloud storage system contains collection of storage servers which provides long-term storage services over the internet. Storing data in a third party's cloud system causes serious concern over data confidentiality with general encryption schemes only limited functionality can be processed. We propose a threshold proxy re-encryption scheme and integrate it along with a decentralized erasure code thus a secure distributed storage system is formulated. The distributed storage system not only supports secure and robust data storage and retrieval, it also lets a user to forward the data from the storage servers to another user without retrieving the data back. To furnish a cloud storage server for long term storage over the internet, the storage server will also act as a data base server. We can upload data stored in the cloud server through proxy re encryption method. To generate proxy re encryption key for one time data access proxy server will be created virtually for one time data access.

II. METHODS AND MATERIAL

A. Literature Survey

The most prominent issue in cloud environment which affects cloud performance is security. Secured data transfer can be done in cloud environment with proxy re encryption method. Many researchers have proposed various methods [9, 10] and studies [7, 8] to transfer the data confidentially in a cloud network.

Cloud computing System provide a convenient mechanism for users to manage their files or data with the notion called database as a service as a scheme. A user can outsource his encrypted data to un-trusted proxy servers [11]. The Proxy servers can perform functions on the outsourced cipher texts by not knowing anything about the original data or files. Re-encryption scheme [12] use key sharing concept which will store the secret key of each user as shares. Secret sharing [13, 14] in cloud is typically employed with a distributed service to split the private key among the set of servers.

Comparison of Algorithms:

S. no	Title	Methods/ algorithms	Advantages	disadvantages
1	Decentralize-d Erasure Codes for Distributed Networked Storage[16]	Randomized algorithm	Distributed network storage can be useful for peer-to-peer network or raid systems	Limited memory for storage space
2	Ubiquitous Access to Distributed Data in Large-Scale Sensor Networks through Decentralized Erasure Codes[15]	Decentralized erasure code	Reliable distributed storage	Enabling ubiquitous access to the distributed data packet is difficult here
3	A Secure Decentralized Erasure Code for Distributed Networked Storage[1]	Public key encryption	each storage server perform the encoding process in a decentralized way	privacy issue of the distributed networked storage system
4	System Support for Automated Availability Management [17]	Resolutionbased algorithm	find (and fix) some unknown attacks, and clarify some design details that may be relevant for other storage protocols	Improve the protocol's performance but complicate its security properties

B. Problem Definition

The existing system drawbacks are

- General encryption schemes protect data confidentiality, but to a limited functional of the storage system because only few operations are supported over encrypted data.
- Constructing a secure storage system supports multiple functions which is challenging when the storage system is distributed and has no central authority.
- Data robustness is one of the major requirements for storage systems.

C. Proposed System

A protocol based threshold proxy re-encryption scheme is proposed which integrate it with a decentralized erasure code such that a secure distributed storage system is formulated. The threshold proxy re-encryption scheme will provide a secure cloud storage system with secure data storage and secure data forwarding functionality in a decentralized structure.

Storing data in a third party's cloud system may cause serious concern on data confidentiality. In order to provide strong confidentiality for messages in storage servers, the user encrypt messages by a cryptographic method before applying an erasure code method to encode and store messages. When the user wants to view the message, he needs to retrieve the codeword symbols from storage servers, decode them, and then decrypt them by using the cryptographic keys.

Advantages of proposed system:

- A cloud storage system is designed with robustness, confidentiality, and functionality.
- When a cloud storage system is considered as a large scale distributed storage system that consists of many independent storage servers the proposed method works efficiently.

Architecture diagram:

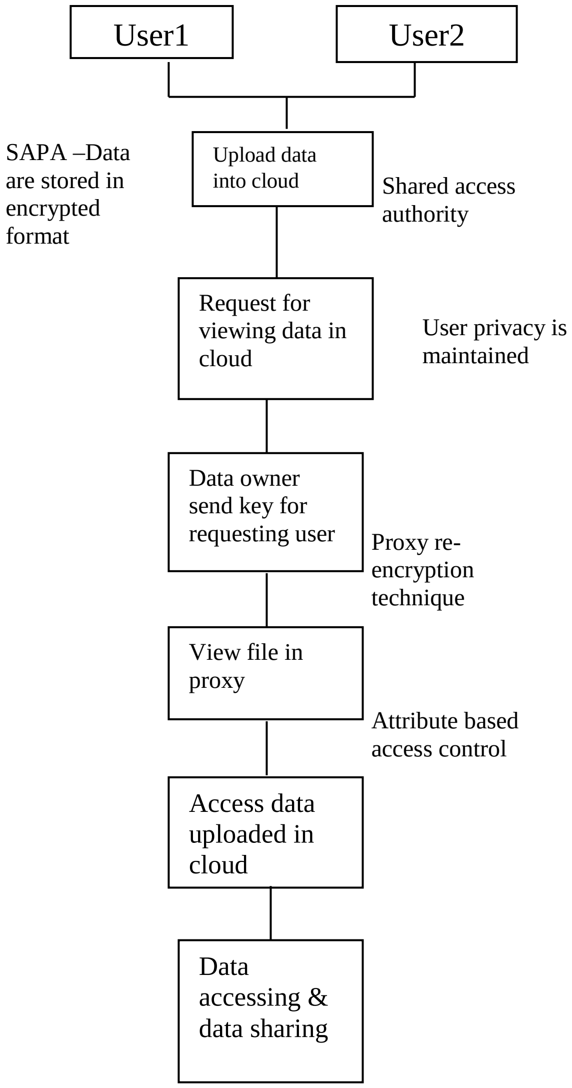

User1
User2
SAPA –Data are stored in encrypted format
Upload data into cloud
Shared access authority
Request for viewing data in cloud
User privacy is maintained
Data owner send key for requesting user
Proxy re-encryption technique
View file in proxy
Attribute based access control
Access data uploaded in cloud
Data accessing & data sharing

Figure 1 : Architecture diagram of secured threshold proxy re-encryption server with a decentralized erasure code.

D. METHODOLOGIES:

Distributed Erasure Code:

We consider the problem of constructing an erasure code for storage over a network when the data sources are distributed in the cloud server. Specifically, we assume that there are n storage nodes with limited memory and k < n sources generating the data. We want a data collector, who can appear anywhere in the network for accessing the data, to query any k storage nodes and be able to retrieve the data. We introduce Decentralized Erasure Codes, which are linear codes with a specific randomized structure inspired by network coding on random bipartite graphs with encrypted format. We show that decentralized erasure codes are optimally sparse, and lead to reduced communication, storage and computation cost over random linear coding over the cloud server.

Erasure code over a cloud Network:

Decentralized erasure codes are random linear codes over a finite field Fq with a specific randomized structure on their generator matrix. Each data packet Di is seen as a vector of elements of a finite field fi . We denote the set of data nodes by V1 with |V1| = k and storage nodes by V2, |V2| = n. We will now give a description of a randomized construction of a bipartite graph that corresponds to the creation of a decentralized erasure code. Every data node i ∈ V1 is assigned a random set of storage nodes N (i). This set is created as follows: a storage node is selected uniformly and independently from V2 and added in N(i) and this procedure is repeated d(k) times. Therefore N (i) will be smaller than d(k) if the same storage node is selected twice. In fact, the size of the set N (i) is exactly the number of coupons a coupon collector would have after purchasing d(k) coupons from a set of n coupons. It is not hard to see that when d (k) ≪ n, N(i) will be approximately equal to d(k) with high probability. Denote by N (j) = {i ∈ V1 : j ∈ N(i)} the set of data nodes that connect to a storage node. Each storage node will create a random linear combination of the data nodes it is connected with:

Sj = X ∀i:∈N(j) fij Di where the coefficients fij are selected uniformly and independently from a finite field Fq. Each storage node also stores the fij coefficients, which requires an overhead storage of N(j)(log2 (q) + log2 (k)) bits. This construction can be summarized into s = mG where s is a 1 × n vector of stored data, m is 1 × k data vector and G is a k × n matrix with non-zero entries corresponding to the adjacency matrix of the random bipartite graph we described. The key property that allows the decentralized construction of the code is that each data node is choosing its neighbors independently and uniformly or equivalently, every row of the generator matrix is created independently and has N(i) = O(d(k)) nonzero elements. This row independence, which we call “decentralized property. We compare our results with random linear coding for distributed networked storage. A data collector querying k storage nodes will gain access to k encoded packets. To reconstruct, the data collector must invert a k × k sub matrix G′ of G. Therefore, the key property required for successful decoding is that any selection of G′ forms a full rank matrix with high probability. Clearly d(k) is measuring the sparsely of G. Making d(k) as small as possible is very important since it is directly related with overhead storage, decoding complexity and communication cost. Our main contribution is identifying how small d(k) can be made for matrices that have the decentralized property to ensure that their sub matrices are full rank with high probability. The following theorems are the main results of this correspondence:

Theorem 1:

Let G be a random matrix with independent rows constructed as described. Then, d(k) = c ln (k) is sufficient for a random k ×k sub matrix G′ of G to be non-singular with high probability. More specifically, P r det(G′) = 0] ≤ k q + o(1) for any c > 5 n k.

Theorem 2:

(Converse) If each row of G is generated independently (Decentralized property), at least d(k) = **Ω**(ln(k)) is necessary to have G′ invertible with high probability. From the two theorems it follows that d(k) = c ln(k) is (order) optimal, therefore, decentralized erasure codes have minimal data node degree and logarithmically many nonzero elements in every row. Decentralized erasure codes can be decoded using Maximum Likelihood (ML) decoding, which corresponds to solving a linear system of k equations in GF(q). This has a decoding complexity of O(k 3). Note however that one can use the sparsely of the linear equations and have faster decoding. Using the Wiedemann algorithm one can decode in O(k 2 log(k)) time on average with negligible extra memory requirements.

Comparison Table : Authentication protocol and Proxy key protocol.

Authentication Protocol	**Proxy Key Protocol**
It is a type of cryptographic protocol specifically designed for the transfer of authentication data between two entries	PKP is an example of a hash based message authentication code (HMAC) .It contains a secret key with the current timestamp using a cryptographic hash function to generate a onetime password
It involves two or more parties and everyone involved in the protocol must know the protocol in advance	a user will enter username and password into a website or other server, generate a one-time password for the server using PKP
Pap-password authentication protocol: it is highly insecure because the highly credentials are being transmitted over the network in plain ASCII text	**One time:** the user can view the data only one time in his life time
Chap- challenges handshake ap: it is not useful for large installations, every secrets are maintained in both ends	**Time based protocol**: user can view the data by using the key within a particular time has set
Eap- extensible authentication protocol: It does not provide security which is basically depends on implementation	**Date based protocol:** user can view the data only in a given date

III. RESULTS AND DISCUSSION

The Result obtained according to the committed abstract. The user uploaded data in the cloud server can be viewed in the encrypted format which is done with re-encryption method with the prior authentication (i.e.) with username and password.

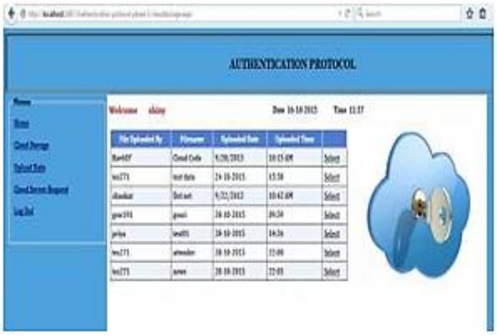
Encryption type = 64 bit key
Key Length = 23 char
Key type = Alpha numerical with special characters.
Key character = Encryption and decryption
Key Limitation = Caps alphabets = 26, Small alphabets = 26, 0 -9 numbers = 10, Special Characters = 10:
Result = **3.848329407410064e+135** of key combinations can be produced. It is equivalent to 30000 trillion and above combination.

Acknowledgement for cloud server request sent: here the proxy key is sent along with this request to user for accessing the file. The re encrypted key will be received by the user. Process:

After data owner authorized the sender request.

1. Key will be generate from the cloud side and sent to the user.
2. Proxy server will be created on user request.
3. Data in the cloud storage will be decrypted.
4. Decrypted data will be sent to the proxy server.

The index page allows the new user to sign up. New user can generate the user name by entering name, age, gender and the user name and password will be generated. The user login's to the cloud server with their username and password. Then in user home page, the user can see the request accepted by another user.

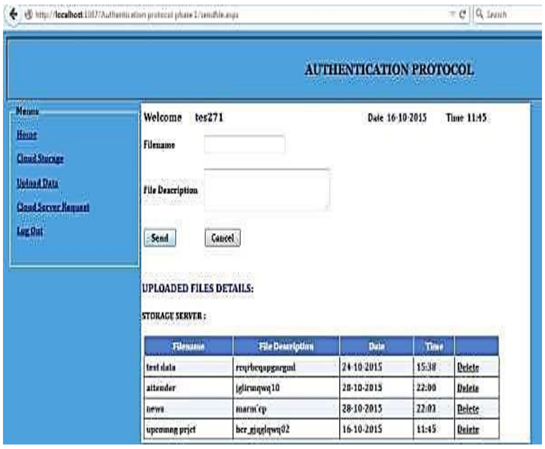


Welcome shiny Date 16-10-2015 Time 11:37

File Uploaded By	Filename	Uploaded Date	Uploaded Time	
Harsh07	Cloud Code	9/28/2015	10:15 AM	Select
tes271	test data	24-10-2015	15:38	Select
shankar	Dot net	9/22/2015	10:42 AM	Select
gow191	gowli	28-10-2015	09:54	Select
priya	test01	28-10-2015	14:26	Select
tes271	attender	28-10-2015	22:00	Select
tes271	news	28-10-2015	22:03	Select

Figure : User's cloud server

The user can now view the files uploaded by the other users in the cloud server. To view the other user's uploaded file, select the file and send request. Now the request will be uploaded.



Welcome tes271 Date 16-10-2015 Time 11:45

Filename

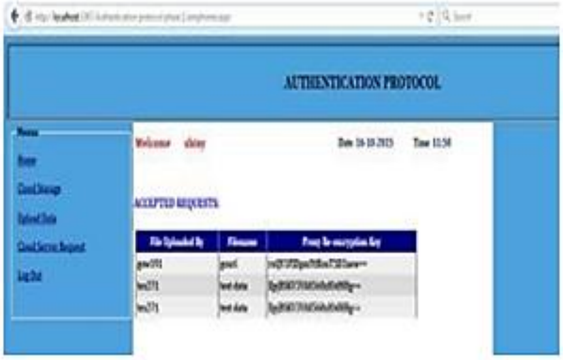
File Description

UPLOADED FILES DETAILS:

STORAGE SERVER :

Filename	File Description	Date	Time	
test data	reqrbcqapgargml	24-10-2015	15:38	Delete
attender	jglirmqwq10	28-10-2015	22:00	Delete
news	marm'cp	28-10-2015	22:03	Delete
upcomng prjct	ber_gjqglqwq02	16-10-2015	11:45	Delete

To upload the file to the cloud, add the file and send it. Similarly other users in the cloud also send the receive files.



Welcome shiny Date 16-10-2015 Time 11:38

ACCEPTED REQUESTS

File Uploaded By	Filename	Proxy Re-encryption Key
gow191	gowli	[illegible]
tes271	test data	[illegible]
tes271	test data	[illegible]

The user gets the accepted response by the other user and now the user can view the file.

IV. CONCLUSION

Thus we are concluding that all the results are obtained according to the committed abstract. In this paper, a cloud storage system is considered which consists of storage servers and key servers. We integrate a newly proposed threshold proxy re-encryption scheme and erasure codes over exponents. The threshold proxy re-encryption scheme supports encoding, forwarding, and partial decryption operations in a distributed way. To decrypt a message of k blocks that are encrypted and encoded ton code word symbols, each key server only has to partially decrypt two codeword symbols in our system. By using the 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. Moreover, each storage server independently performs encoding and re-encryption and each key server independently perform partial decryption. Our storage system and some newly proposed content addressable file systems and storage systems are highly compatible. Our storage servers act as storage nodes in a content addressable storage system for storing content addressable blocks. Our key servers act as access nodes for providing a front-end layer such as a traditional file system interface. Further study on detailed cooperation is required.

V. REFERENCES

[1] Hsiao-Ying Lin ; Dept. of Comput. Sci., Nat. Chiao Tung Univ., Hsinchu, Taiwan ; Wen-Guey Tzeng," A Secure Decentralized Erasure Code for Distributed Network Storage", Parallel and Distributed Systems, IEEE Transactions on

(Volume:21 , Issue: 11), February 2010, pp. 1586 - 1594.

[2] Mell P, Grance T (2012) The NIST definition of cloud computing. NIST Spec Publ 800:145. National Institute of Standards and Technology, U.S. Department of Commerce. Source: http:// csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf. Accessed on Oct 2012

[3] Wikipedia definition of Cloud computing (2012). Source: http://en.wikipedia.org/wiki/Cloud_ computing. Accessed on Oct 2012.

[4] Rocha F, Abreu S, Correia M (2011) The final Frontier: confidentiality and privacy in the cloud, pp 44–50.

[5] Huang R, Gui X, Yu S, Zhuang W (2011) Research on privacy-preserving cloud storage framework supporting ciphertext retrieval. International conference on network computing and information security 2011:93–97.

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

[7] Renjith P , Sabitha S,"Survey on Data Sharing and Re-Encryption in Cloud", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 2, Issue 2, February 2013.

[8] Tushar A. Rane , Shrishail T. Patil , Anita H. Khade,"Survey on Security Challenge for Data forwarding in Cloud", International Journal of Science and Research (IJSR), India Online ISSN: 2319-7064.

[9] Anurag Porwal, Rohit Maheshwari, B.L.Pal, Gaurav Kakhani, "An Approach for Secure Data Transmission in Private Cloud", International Journal of Soft Computing and Engineering (IJSCE), Volume-2, Issue-1, March 2012

[10] Varsha S.Agme , Prof. Archana C.Lomte, " Cloud Data Storage Security Enhancement Using Identity Based Encryption", International Journal of Application or Innovation in Engineering & Management (IJAIEM), Volume 3, Issue 4, April 2014.

[11] Ivan and Y. Dodis, "Proxy cryptography revisited," in Proc.Network and Distributed System Security Symposium - NDSS'03, (San Diego, California, USA), pp. 1–20, The Internet Society, Feb. 2003 .

[12] H. Y. Lin and W. Tzeng, "A secure erasure code-based cloud storage system with secure data forwarding," IEEE Transactions on Parallel and Distributed Systems, vol. 23, pp. 995 – 1003, June 2012.

[13] G. R. Blakley. Safeguarding cryptographic keys. In Proceedings of the National Computer Conference, 48, pages 313–317. American Federation of Information Processing Societies Pro- ceedings, 1979.

[14] A. Shamir. How to share a secret. Communications of the ACM, 22(11):612–613, November 1979.

[15] Dimakis, A.G. , Prabhakaran, V. Ramchandran, K., "Ubiquitous access to distributed data in large-scale sensor networks through decentralized erasure codes" Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium ,pp. 111 – 117

[16] Dimakis, A.G. Prabhakaran, V. Ramchandran, K. Decentralized erasure codes for distributed networked storage", Information Theory, IEEE Transactions on (Volume:52 , Issue: 6), pp.2809 - 2816