PROXY RE-ENCRYPTION SCHEME FOR SECURE MULTI-DATA FORWARDING USING CLOUD STORAGE SYSTEM

KR.NAGA SINDHU*

ABSTRACT:

Cloud Computing has been envisioned as the next-generation architecture of IT Enterprise. A cloud storage infrastructure differs from a traditional data storage infrastructure in that it accesses files remotely over a internet and is usually built on an object-based storage platform. It moves the application software and databases to the centralized large data centers, it reduce user burden. Where the management of the data and services may not be fully trustworthy. We propose a proxy re-encryption scheme for providing secure cloud storage system service. It supports user secure forward his data to the storage servers to another user without getting the data back. Finally we simulate the cloud storage system which provide service to multiple users are request for same service, at a time the owner not getting data back and provide secure service over a cloud storage system

keywords: Decentralized erasure code, proxy re-encryption, threshold cryptography, secure storage system.

* M.TECH., INFORMATION TECHNOLOGY, PSN COLLEGE OF ENGINEERING AND TECHNOLOGY, TIRUNELVELI

1 INTRODUCTION:

Cloud computing is only a different way to deliver computer resources, rather than a new technology, it has sparked a revolution in the way organizations provide information and service. Computers in the cloud are configured to work together and the various applications use the collective computing power as if they are running on a single system. In this paper, we focus on designing a cloud storage system for robustness, confidentiality, and functionality. A cloud storage system is considered as a large-scale distributed storage system that consists of many independent storage servers. Data robustness is a major requirement for storage. It contain many proposals of storing data over storage servers [1], [2], [3], [4], [5]. One way to provide data robustness is to replicate a message such that each storage server stores a copy of the message. It is very robust because the message can be retrieved as long as one storage server survives. Another way is to encode a message of m symbols into a codeword of n symbols by erasure coding. Storing data in a third party's cloud system causes serious concern on data confidentiality.

In order to provide strong confidentiality for messages in storage servers, a user can encrypt messages by a cryptographic method before applying an erasure code method to encode and store messages.

User forwarding data to another user by storage servers directly under the command of data owner. System model that consists of distributed storage servers and key servers. A user distributes his cryptographic key to key servers that shall perform cryptographic functions on behalf of the user. These key servers are highly protected by security mechanisms. To well fit the distributed structure of systems, we require that servers independently perform all operations. With this consideration, 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. There are three problems in the above straightforward integration of encryption and encoding. First, the cloud storage system forward one service at a time.Second, this setting not allows flexible adjustment between the storage server. Third the speed of responding is low, at a time it provides one service. So other request are waiting in a queue for getting a response.

Our Motivation: In this paper, we address the problem of forwarding data to multiple user by storage servers directly under the command of the data owner.These key servers are highly protected by security mechanisms.To well fit the distributed structure of systems, we require that servers independently perform all operations.With this consideration, we propose a new threshold proxy re-encryption scheme and integrate it with a secure decentralized code to form a secure distributed storage system. It efficiently meet the requirements of data robustness, data confidentiality, and data forwarding. Accomplishing the integration with consideration of a distributed structure is challenging. Our system meets the requirements that storage servers independently perform encoding and re-encryption and key servers independently perform partial decryption. Moreover, we consider the system in a more general setting than previous works. This setting allows more flexible adjustment between the number of storage servers and robustness.

Our contributions: Assume that there are n distributed storage servers and m key servers in the cloud storage system. A message is divided into k blocks and represented as a vector of k symbols.

We construct a secure cloud storage system that supports the multiple function of secure data forwarding by using a threshold proxy re-encryption scheme. The encryption scheme supports decentralized erasure codes over encrypted messages and forwarding operations over encrypted and encoded messages. Our system is highly distributed where storage servers independently encode and forward messages and key servers independently perform partial decryption.

2.RELATED WORKS:-

We briefly review decentralized erasure code, distributed storage systems, proxy re encryption schemes, and integrity checking mechanisms.

2.1 Decentralized Erasure Code

The problem of reconstructing the k packets from any k out of n storage nodes is essentially an erasure channel coding problem. If we assume the existence of a centralized super node that can gather all the data, we could use any (n; k) erasure code. More specially, the centralized node would gather the k data packets, use an erasure code to generate n encoded packets, and assign and send one encoded packet to each storage node. If we use a good erasure code, we will be guaranteed to reconstruct the original packets by asking any k encoded storage nodes. In fact, any erasure code could be used even

without gathering the data in one location if there was a mechanism to create the code and coordinate the data nodes. Essentially each data node corresponds to one row in the generator matrix of the code.

2.2 Distributed Storage Systems

At the early years, the Network-Attached Storage (NAS) [7]and the Network File System (NFS) [8] provide extra

storage devices over the network such that a user can access the storage devices via network connection. Afterward, many improvements on scalability, robustness, efficiency,and security were proposed [1], [2], [9].A decentralized architecture for storage systems offers good scalability, because a storage server can join or leave without control of a central authority. To provide robustness against server failures, a simple method is to make replicas of each message and store them in different servers. However, this method is expensive as z replicas result in z times of expansion. One way to reduce the expansion rate is to use erasure codes to encode messages [10], [11], [12], [13], [5]. A message is encoded as a codeword, which is a vector of symbols, and each storage server stores a codeword symbol. A storage server failure is modeled as an erasure error of the stored codeword symbol. Random linear codes support distributed encoding, that is, each codeword symbol is independently computed. To store a message of k blocks, each storage server linearly combines the blocks with randomly chosen coefficients and stores the codeword symbol and coefficients.

2.3 Proxy Re-Encryption Schemes

Proxy re-encryption schemes are proposed by Mambo and Okamoto [14] and Blaze et al. [15]. In a proxy re-encryption scheme, a proxy server can transfer a cipher text under a public key PKA to a new one under another public key PKB by using the re-encryption key RKA!B. The server does not know the plaintext during transformation. Ateniese et al. [6] proposed some proxy re-encryption schemes and applied them to the sharing function of secure storage systems. In their work, messages are first encrypted by the owner and then stored in a storage server. When a user wants to share his messages, he sends a re-encryption key to the storage server. The storage server re-encrypts the encrypted messages for the authorized user. Thus, their system has data confidentiality and supports the data forwarding function. Our work further integrates encryption, re-encryption, and encoding such that storage robustness is strengthened. Type-based proxy re-encryption schemes proposed by Tang [7]

provide a better granularity on the granted right of a re-encryption key. A user can decide which type of messages and with whom he wants to share in this kind of proxy re-encryption schemes.

2.3 Integrity Checking Functionality

Another important functionality about cloud storage is the function of integrity checking. After a user stores data into the storage system, he no longer possesses the data at hand. The user may want to check whether the data are properly stored in storage servers. The concept of provable data Possession [2], [1] and the notion of proof of storage [2],[3], [4] are proposed. Later, public audit ability of stored data is addressed in [5]. Nevertheless all of them consider the messages in the clear text form.

3 SYSTEM ARCHITECTURE

We represent a system model that consists of four phases. As shown in Fig. 1, our system model consists of users, n storage servers SS1; SS2; . . . ; SSn, and m key servers KS1;KS2;…; KSm. Storage servers provide storage services and key servers provide key management services. They work independently. Our distributed storage system consists of five phases: system setup, data storage, proxy re-encryption, data forwarding, and data retrieval. These five phases are described as follows.

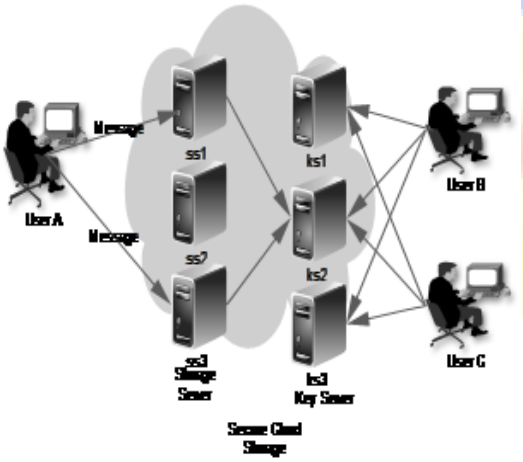

Fig 1.A General System Model for Our Work

In the *System Setup* phase, Our system model consists of users, n storage servers SS1; SS2; . . . ; SSn, and m key servers KS1; KS2; . . . ; KSm. Storage servers provide storage services and key servers provide key management services. In the system setup phase, the system manager chooses system parameters and publishes them. Each user A is assigned a public-secret key pair (PKA; SKA) and ShareKeyGen (). User A distributes his secret key SKA to key servers such that each key server KSi holds a key share SKA; and the public key is kept by user.

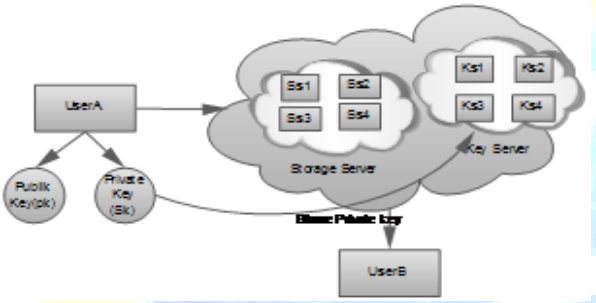

Fig 2.A System Setup Phase

In the *data storage* phase, user A encrypts his message M and dispatches it to storage servers. A message M is decomposed into k blocks m1; m2; . . .; mk and has an identifier1 ID. One way to provide data robustness is to replicate a message such that each storage server stores a copy of the message. User A encrypts each block mi into a cipher text Ci and sends it to v randomly chosen storage servers. A storage server receives a set of original cipher texts with the same identity from A. Upon receiving cipher texts from a user, the storage server performs Encode on the set of k cipher texts and stores the encoded result (codeword symbol).

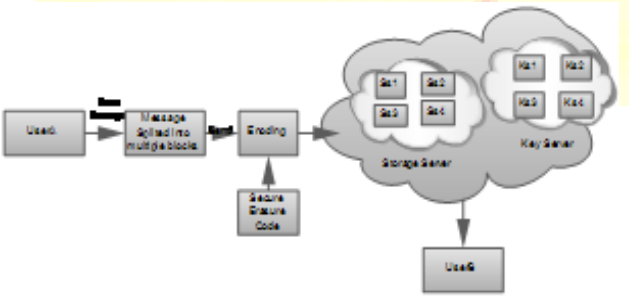

Fig 3 A Data Storage Phase Setup

The *proxy re-encryption* scheme phase supports encoding, forwarding, and partial decryption operations in a distributed way. 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. Messages are first encrypted by the owner and then stored in a storage server. When a user wants to share his messages, he sends a re-encryption key to the storage server. The storage server re-encrypts the encrypted messages for the authorized user. Proxy re-encryption schemes can significantly decrease communication and computation cost of the owner. Proxy re-encryption schemes significantly reduce the overhead of the data forwarding function in a secure storage system.

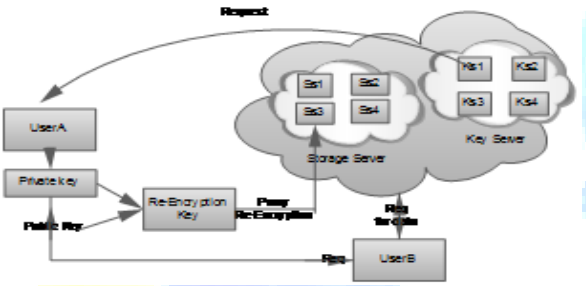

Fig 4 A Proxy Re-Encryption scheme Setup

In *Data forwarding* phase, we construct a secure cloud storage system that supports the function of secure data forwarding by using a proxy re-encryption scheme. In the data forwarding phase, a user runs Key Recover() and Re Key Gen() send to storage server, and each storage server performs Re Enc().In the data forwarding phase, 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. To do so, A uses his secret key SKA and B's public key PKB to compute a re-encryption key RKID A B and then sends RKID A B to all storage servers. Each storage server uses the re-encryption key to re-encrypt its codeword symbol for later retrieval requests by B. The re-encrypted codeword symbol is the combination of cipher texts under B's public key. In order to distinguish re-encrypted codeword symbols from intact ones, we call them original codeword symbols and re encrypted codeword symbols, respectively.

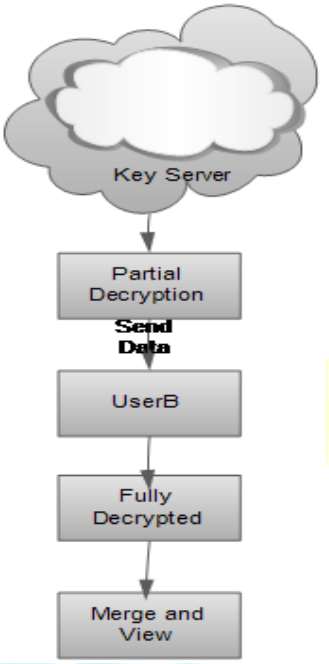

Fig 5 A Data Forwarding Phase

In the *Data retrieval* phase, user A requests to retrieve a message from storage servers. The message is either stored by him or forwarded to him. User A sends a retrieval request to key servers. Upon receiving the retrieval request and executing a proper authentication process with user A, each key server KSi requests u randomly chosen storage servers to get codeword symbols and does partial decryption on the received codeword symbols by using the key share SKA ; I. Finally, user A combines the partially decrypted codeword symbols to obtain the original message M.

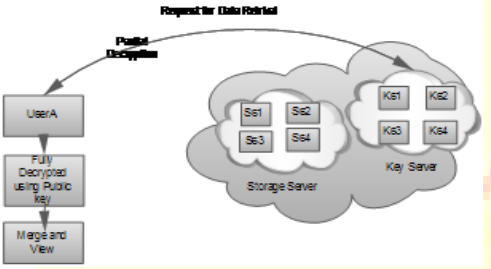

Fig 6 A Data Retrieval Phases

4. Construction of secure cloud storage system security

Our approach. We use a threshold proxy re-encryption scheme with multiplicative homomorphic property. An encryption scheme is multiplicative homomorphic if it supports a group operation on encrypted plaintexts without decryption

$D(SK,E(PK,m1)*E(PK,m2))=m1.m2$

where E is the encryption function, D is the decryption

function, and (PK,SK) is a pair of public key and secret key. Given two coefficients g1 and g2, two message symbols m1 and m2 can be encoded to a codeword symbol m^g ,m^{g2} in the encrypted form

$$C=E(PK,m1)^{g1}*E(PK,m2)^{g2}=E(PK,m1^{g1}.m2^{g2})$$

Thus, a multiplicative homomorphic encryption scheme supports the encoding operation over encrypted messages.We 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 via the Shamir secret sharing scheme [6], where $t \geq k$. In our system, to decrypt for a set 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 available, k codeword symbols are obtained from the partially decrypted ciphertexts.

.

5. Analysis

Storage cost. To store a message of k blocks, a storage Server SSj stores a codeword symbol (b,α,β,Ÿ) and the Coefficient vector (g1,j; g2,j . . . gk,j). They are total of (1 +2l1+l2+kl3) bits,The average cost for a message bit stored in a storage server is (1 + 2l1 +l2 + kl3)/kl2 bits, which is dominated by l3=l2 for a sufficiently large k. In practice, small coefficients, l3 / l2, reduce the storage cost in each storage server.

Correctness. There are two cases for correctness. The Owner A correctly retrieves his message and user B correctly retrieves a message forwarded to him. The correctness of encryption and decryption for A. The correctness of re-encryption and decryption for B. As long as at least k storage servers are available, a user can retrieve data with an overwhelming probability. Thus, our storage system tolerates n -k server failures.

Security. The data confidentiality of our cloud storage system is guaranteed even if all storage servers, nontarget users, and up to (t -1) key servers are compromised by the attacker.

The probability of a successful retrieval. A successful retrieval is an event that a user successfully retrieves all k blocks of a message no matter whether the message is owned by him or forwarded to him. The randomness comes from the random selection of storage servers in the data storage phase, the random coefficients chosen by storage servers, and the random selection of key servers in the data retrieval phase. The probability of a successful retrieval depends on (n, k, u,v) and all randomness.

5 DISCUSSIONS AND CONCLUSION

In this paper, we consider a cloud storage system 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 reencryption scheme supports encoding, forwarding, and partial decryption operations in a distributed way. To decrypt a message of k blocks that are encrypted and encoded to n codeword 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 performs partial decryption. Our storage system and some newly proposed content addressable file systems and storage system [13], [14], [15] 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.

ACKNOWLEDGEMENT

The author would like to thank the anonymous reviewers for their many valuable comments and suggestions that help to improve both the technical content and the presentation quality of this paper

REFERENCES

[1] .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.

[2] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," Proc. 14th ACM Conf. Computer and Comm. Security (CCS), pp. 598-609, 2007.

[3] K.D. Bowers, A. Juels, and A. Oprea, "HAIL: A High-Availability and Integrity Layer for Cloud Storage," Proc. 16th ACM Conf.Computer and Comm. Security (CCS), pp. 187-198, 2009.

[4] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing," Proc. IEEE 29th Int'l Conf. Computer Comm. (INFOCOM), pp. 525-533, 2010.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

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

[10] W. Dong, F. Douglis, K. Li, H. Patterson, S. Reddy, and P. Shilane,"Tradeoffs in Scalable Data Routing for Deduplication Clusters,"Proc. Ninth USENIX Conf. File and Storage Technologies (FAST), p. 2,2011.

[11] 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.

[12] 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.

[13] G. Ateniese, S. Kamara, and J. Katz, "Proofs of Storage from Homomorphic Identification Protocols," Proc. 15th Int'l Conf.Theory and Application of Cryptology and Information Security (ASIACRYPT), pp. 319-333, 2009.

[14] A.G. Dimakis, V. Prabhakaran, and K. Ramchandran, "Decentralized Erasure Codes for Distributed Networked Storage," IEEE Trans. Information Theory, vol. 52, no. 6 pp. 2809-2816, June 2006.

[15] H.-Y. Lin and W.-G. Tzeng, "A Secure Decentralized Erasure Code for Distributed Network Storage," IEEE Trans. Parallel and Distributed Systems, vol. 21, no. 11, pp. 1586-1594, Nov. 2010.

Author's Profile:

KR. Naga Sindhu received the B. Tech degree in Information Technology from the Mookambigai College of Engineering,Trichy,India,in 2011 and is currently purchasing the M.Tech degree in Information Technology in the Department of Information Technology at the University of Chennai, college PSNCET.

Her Research interest in Cloud Computing and network Security.