IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 9, Issue 5 (Mar. - Apr. 2013), PP 27-31
www.iosrjournals.org

Threshold Proxy Re-Encryption Scheme and Decentralized Erasure Code in Cloud Storage With Secure Data Forwarding

S.Amritha[1], Mr.S.Saravana Kumar[2]
[1]*M.E.(Cse), Srinivasan Engg College,Perambalur,Tamilnadu,India.*
[2]*Ap/Cse, Srinivasan Engg College, Perambalur,Tamilnadu, India.*

Abstract : *A cloud storage system, used to store large number of data in storage server. Cloud system is used to provide large number storage servers, which provide long-term storage service over the Internet. Third party's cloud system does not provide data confidentiality. Constructing centralized storage system for the cloud system makes hackers stole data easily. General encryption schemes protect data confidentiality. In the proposed system a secure distributed storage system is formulated by integrating a threshold proxy re-encryption scheme with a decentralized erasure code. The distributed storage system not only supports secure and robust data storage and retrieval, but also lets a user forward data from one user to another without retrieving the data back. The main technical involvement is that the proxy re-encryption scheme supports encoding operations over encrypted messages as well as forwarding operations over encoded and encrypted messages. The method fully integrates encrypting, encoding, and forwarding. The proposed system is applied for military and hospital applications, then other secret data transmission.*

Keywords *- Decentralized erasure code, proxy re-encryption, threshold cryptography, secure storage system.*

I. INTRODUCTION

Cloud computing is a model that treats the resources on the Internet as a unified entity, a cloud. Users use a service without being concerned about how computation is done and storage is managed. This method used to focus on designing a cloud storage system for robustness, privacy, and functionality. A cloud storage system is considered as a large scale distributed storage system that consists of many self-governing storage servers. Data robustness is a major obligation for storage systems. There are many proposals of storing data over storage servers. One way to present data robustness is to duplicate a message such that each storage server stores a copy of the message. It is robust because the message can be retrieved as long as one storage server survives. Another way is to encrypt a message of k symbols into a codeword of n symbols by erasure coding. To store a message, each of its encoded messages is stored in a various storage server.A storage server failure corresponds to an erasure error of the encode symbol. As long as the number of failure servers is under the acceptance threshold of the erasure code, the message can be recovered from the encode symbols stored in the available storage servers by the decoding process. This provides a trade off between the storage size and the acceptance threshold of failure servers. A decentralized erasure code is an erasure code that independently computes each codeword symbol for an encrypted message. Thus, the encoding process for a message used to split up message into n parallel tasks of generating codeword symbols. A distributed storage system is constructed a decentralized erasure code. After the messages are sent to storage servers, each storage server separately computes a codeword symbol for the received message and stores it. This finishes the encoding operation and storing process. The recovery process is also the same process like a encoding process. Storing data in a third party's cloud system will not provide a data confidentiality. In order to provide well-built confidentiality for messages in storage servers, a user encrypt messages by a threshold cryptography method before applying an erasure code method to encode and store messages. When user wants to use a message, user needs to recover the codeword symbols from storage servers, decode, and decrypt them by using cryptography keys. There are three problems in the above simple integration of encryption and encoding methods. First, the user has to do computation and the communication traffic between the user and storage servers is far above the ground. Second, the user has to manage his cryptography keys. If the user's tool of storing the keys is vanished or compromise, the security is broken. Finally, in addition data storing and retrieving, it is inflexible for storage servers to directly support other functions. For example, storage servers cannot frankly forward a user's messages to another user. The owner of messages has to retrieve message, decode, decrypt and then forward them to another user.

II. RELATED WORK

2.1 Ocean Store

Ocean Store is a utility infrastructure designed to span the globe and provide continuous access to persistent information. Since this infrastructure is comprised of untrusted servers, data is protected through

redundancy and cryptographic techniques. To improve performance, data is allowed to be cached anywhere, anytime. Additionally, monitoring of usage patterns allows adaptation to regional outages and denial of service attacks; monitoring also enhances performance through pro-active movement of data. A prototype implementation is currently under development. In the past decade it has seen astounding growth in the performance of computing devices. Even more significant has been the rapid pace of miniaturization and related reduction in power consumption of these devices. Before such a revolution can occur, however, computing devices must become so reliable and resilient that they are completely transparent to the user.

2.2 PAST

This technique sketches the design of PAST, a large-scale, Internet-based, global storage utility that provides scalability, high availability, persistence and security. PAST is a peer-to-peer Internet application and is entirely self organizing. PAST nodes serve as access points for clients, participate in the routing of client requests, and contribute storage to the system. Nodes are not trusted, they may join the system at any time and may silently leave the system Without warning. Yet, the system is able to provide strong assurances, efficient storage access, load balancing and scalability. Among the most interesting aspects of PAST's design are(1) the Pastry location and routing scheme, which reliably and efficiently routes client requests among the PAST nodes, has good network locality properties and automatically resolves node failures and node additions; (2) the use of randomization to ensure diversity in the set of nodes that store a file's replicas and to provide load balancing; and (3) the optional use of smartcards, which are held by each PAST user and issued by a third party called a broker. The smartcards support a quota system that balances supply and demand of storage in the system.There are currently many projects aimed at constructing peer-to-peer applications and understanding more of the issues and requirements of such applications and systems. Peer-to-peer systems can be characterized as distributed systems in which all nodes have identical capabilities and responsibilities and all communication is symmetric. We are developing PAST, an Internet-based, peer-to-peer global storage utility, which aims to provide strong persistence, high availability, scalability and security. The PAST system is composed of nodes connected to the Internet, where each node is capable of initiating and routing client requests to insert or retrieve files. Optionally, nodes may also contribute storage to the system. The PAST nodes form a self-organizing overlay network. Inserted files are replicated on multiple nodes to ensure persistence and availability.

III. SYSTEM MODEL

3.1 DECENTRALIZED ERASURE CODE

In decentralized erasure code which is used to split up the messages or text data into n number of blocks. This is used for splitting purpose. Our result $n=ak^c$ allows that number of storage server be greater than the number of blocks of a text data's. Decentralized erasure code is a first phase of this project. This has been initiated.

In the decentralized erasure code is an erasure code that independently computes each codeword symbol for a message. Thus, the encoding method for a message can be split into n parallel tasks of generating codeword symbols. A decentralized erasure code is used in a distributed storage system. The n blocked message is stored in for the integration process.

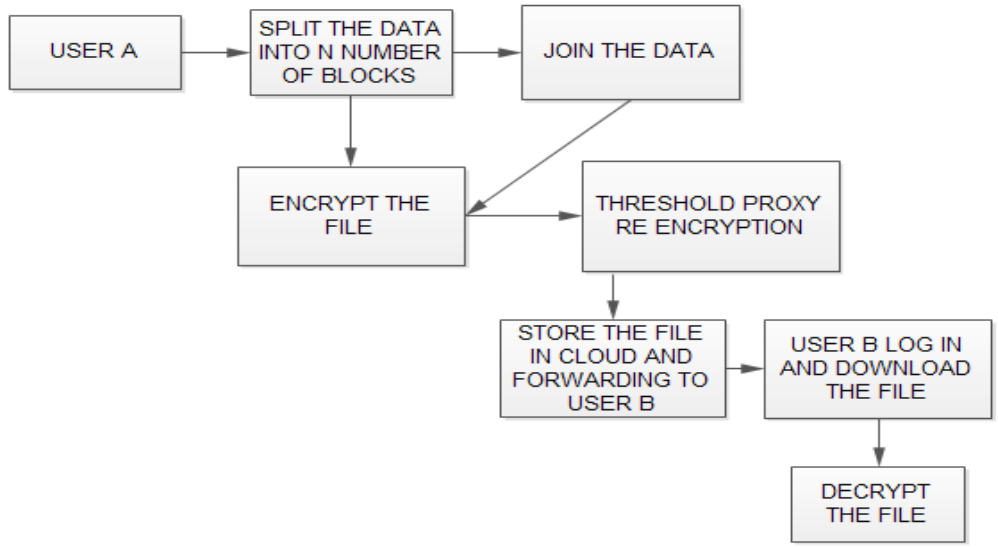

Fig 1: Overview architecture

3.2 INTEGRATION

In an integration processes, the splinted message is joined into an m number of blocks, and stored into lager storage server. User A encrypts his message M is decomposed into k number of blocks m1,m2, . . .,mk and which has an identifier ID. User A encrypts each block mi into a cipher text Ci 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. Note that a storage server may receive fewer than k message blocks and we assume that all storage servers know the value k in advance.

Integration is used to combine messages into m number of block, which is encrypted and stored into a large number storage server. Then forward to user B. Data which is encrypted by using single key. This is produced by using hash key algorithm. In the data storage phase, user A encrypts his message M and dispatches it to storage servers. A message M is decomposed into k number of blocks m1, m2, . . .,mk and which has an identifier ID. User A encrypts each block mi into a cipher text Ci 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. Note that a storage server may receive fewer than k message blocks and it assumes that all storage servers know the value k in advance.

3.3 ENCRYPTION

This is used to encrypt a plain text into a cipher text. Cipher text is produced along with a single key. This key is used to convert the cipher text again into a plain text. The integrated data is encrypted with a single key using random key generation method. Hash key algorithm using random key generation. This is used to generate, the random key. Whenever users encrypt the text, each season time a new key is generated.

Storing data in a third party cloud does not provide Confidentiality in cloud storage. Data confidentiality is provided by threshold proxy re-encryption scheme. Using this technique the data is encrypted twice and key is generated by a random key generation methods using hash algorithm. This is used to generate the more than 10,000 key at the session time.

3.4 DATA FORWARDING

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 SK_A and B's public key PK_B to compute a re-encryption key $RK^{ID}_{A->B}$ and then sends $RK^{ID}_{A->B}$ to all storage servers. Every storage server uses the re encryption key to re-encrypt its codeword symbol for later retrieval needs by B. The re-encrypted codeword symbol is the grouping of cipher texts under B's public key. In order to differentiate re-encrypted codeword symbols from intact ones, we call them unique codeword symbols and re-encrypted codeword symbols, correspondingly.

3.5 LOGIN

Log in page make user to access an account in a cloud server. When user has an account in the cloud server for accessing data and provides other services. User can sign up the page directly else users needed to create new account using create account option.

3.6 UPLOADING FILE

User after sing up his/her account. User forward data to another user using his/her account. Using id of an user and IP address. Upload encrypted files and forward to a user. User upload files along with a key which is used to encrypt the text.

3.7 DATA RETRIEVAL

Date retrieval is the final module of this project. User download data and using proxy re-encryption method text decoded and partial decrypted. A proxy server can transfer a cipher text under a public key PK_A to a new one under another public key PK_B by using the re-encryption key $RK_{A->B}$.

In the data retrieval phase, user A retrieves a message from storage servers. The message is either stored by user A or forwarded to user A. User A sends a recovery request to key servers. Upon receiving the recovery request and execute a proper verification process with user A, each key server KS_i needs u randomly chosen storage servers to get code symbols and does partial decryption on the received code symbols by using the key share $SK_{A,i}$. Finally, user A combine the partially decrypted codeword symbols to obtain the original message M.

There are two suitcases for the data recovery phase. The first case is that a user A retrieves his own message from cloud. When user A needs to retrieve the message with an identifier ID, he informs all key servers with the individuality token A key server first retrieves original code symbols from u randomly chosen storage

servers and then performs partial decryption Share Dec on every retrieved original codeword symbol. The result of partial decryption is called a partially decrypted code symbol. The key server sends the moderately decrypted codeword symbols and the coefficients to user A. After user A collects replies from at smallest amount t key servers and at least k of them are originally from distinct storage servers, he executes Combine on the t partially decrypted codeword symbols to recover the blocks m1,m2, . . .,mk. The second case is that a user B retrieves a message forwarded to user B. User B informs all key servers straight. The collection and combining parts are the same as the first case except that key servers retrieve re-encrypted codeword symbols and perform partial decryption Share-Decrypted on re-encrypted codeword symbols.

IV. Experimental Result

This experiment shows that our approach is practical and could be used in secure data forwarding in distributed environment using cloud storage system. The empirical results show that cost reduction, time consuming, provide more security.

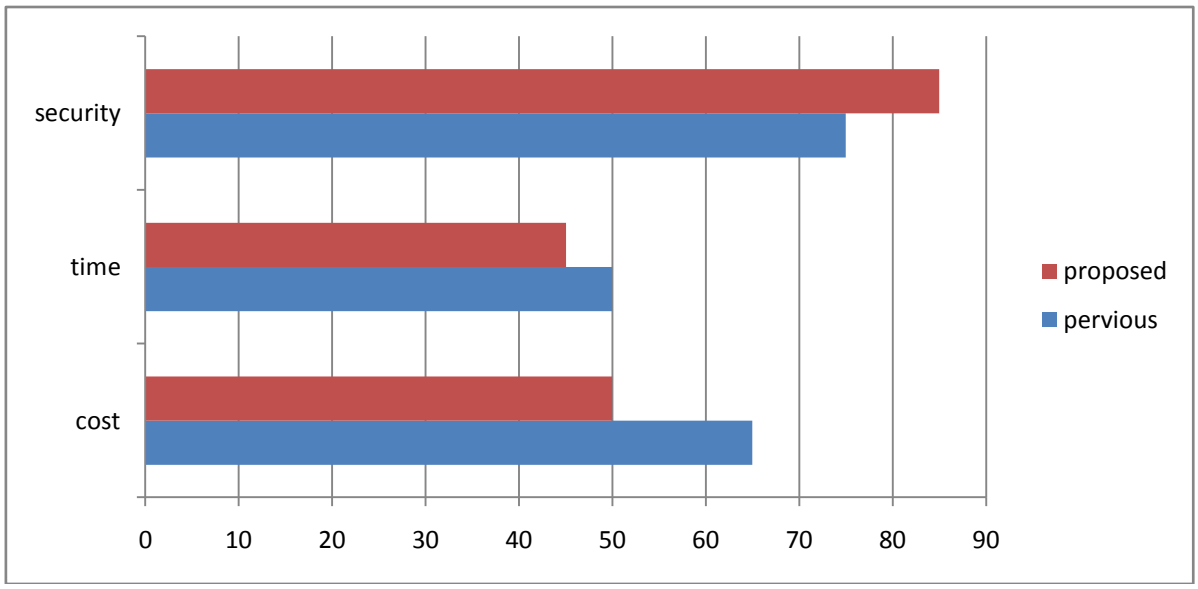

Fig 2: This result shows that compare to the previous system ,the proposed system is provide more security, low cost, time consuming.

V. CONCLUSION

The study of existing system has revealed the use of centralized server, micro bench mark and Third Party Auditor (TPA). The implementations of the traditional systems have resulted in crashes, DOS attacks and unavailability due to regional network outages. In the proposed system a secure distributed storage system is formulated by integrating a threshold proxy re-encryption scheme with a decentralized erasure code. The proxy re-encryption scheme supports not only the expected encoding operations over encrypted messages but also the forwarding operations over encoded and encrypted messages

Acknowledgements

This work was presented in part at the IEEE International Conference on Communications (ICC), 2012.
This work was can be done in part of in our institution and support of all staff members.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] Brownbridge.D.R., L.F. Marshall, and B. Randell, "The Newcastle Connection or Unixes of the World Unite!," Software Practice and Experience, vol. 12, no. 12, pp. 1147-1162, 1982.

[5] Dimakis. A.G, V. Prabhakaran, and K. Ramchandran, "Ubiquitous Access to Distributed Data in Large-Scale Sensor Networks through Decentralized Erasure Codes," Proc. Fourth Int'l Symp. Information Processing in Sensor Networks (IPSN), pp. 111- 117, 2005.

[6] Dimakis.A.G., 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.
[7] Druschel. P 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.
[8] Haeberlen.A, 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.
[9] Hsiao-Ying Lin, Member, IEEE, and Wen-Guey Tzeng, Member "A Secure Erasure Code-Based Cloud Storage System with Secure Data Forwarding"vol. 23, no. 6, june 2012.

Authors Profile

S.Amritha received the B.E Degree computer science and engineering and now she is an M.E student in the Department of Computer Science & Engineering, Srinivasan Engineering College – Dhanalakshmi Srinivasan Group of Institutions, Perambalur, TN, India.

Her research interest includes Network Security and Mobile Computing.

S.Saravana kumar is working as Assistant Professor/CSE, Srinivasan Engineering College – Dhanalakshmi Srinivasan Group of Institutions, Perambalur, TN, India.

His research interest includes pervasive computing, Wireless Networks and Image Processing.