International Journal of Research in Engineering, Science and Management
Volume-3, Issue-1, January-2020
www.ijresm.com | ISSN (Online): 2581-5792

Design and Implementation of Hidden Identity Mechanism for File Storage Server

Pradeep Kumar Patel[1], Chandu Vaidya[2]
[1]*M.Tech., Department of Computer Science and Engineering, Rajiv Gandhi College of Engineering & Research, Nagpur, India*
[2]*Professor, Department of Computer Science and Engineering, Rajiv Gandhi College of Engineering & Research, Nagpur, India*

***Abstract*: Proposed system is hidden identity System. Where, no entity including file storage server having full information about files stored on the storage media. It simply means system will store all users' files on same drive or same folder. Now this will protect owner identity and file access threats. But for this user need to replace complete file storage services on the entire network. For the proof of concept, we can rebuild file storage service and demonstrate it over intranet. Here, first client will pass the file to file storage service along with some security Key; file storage service will then rename the file with unique identification number or code which will be generated by user key. Once this new renamed file stored on the storage owner identity will be lost, hence system will be having no record or information about file owner detail along with file name. Every time while reading or accessing the file or file part system will calculate or generate the name of the file and access it.**

***Keywords*: Hidden identity mechanism, File storage server.**

1. Introduction

The concept 'server' is designed to serve information or services to multiple users over the network, which makes it, enable user profile management. Taken an example of storage server; system need to manage data or files from multiple users or belonging to multiple owners. Managing this ownership, systems normally separates the user files in different directory and records this directory information in index table. By the time system separated the files in different directory for its own management it unknowingly reveals the file owner information to system user who has direct access to the server directory structure. Main threat lies here only, that malicious user has access to every user's files. Malicious user attacks are not controllable or having no direct protection over it but it can be made difficult for the malicious user to get the ownership information about files by hiding the files original name and the ownership information by designing novel index table which has no record about files real identity and its location information. So by the time of accessing the files its location and name information will be generated temperately using secure key.

A. Background

Record stockpiling comprises of document frameworks that are provisioned to at least one servers in a NAS situation. A record framework is connected to servers with conventions, for example, Network File Systems (NFS) or Common Internet File System (CIFS) over ethernet. I/O access on a capacity framework is cultivated through perusing or composing singular records of information. A File stockpiling server is a kind of server that is utilized to store, get to, secure and oversee advanced information, records and administrations. It is a reason constructed server utilized for putting away and getting to little to substantial measure of information over a mutual system or through the Internet. has more storage room, stockpiling access interfaces and specific information recovery and the executive's utilities. A capacity server by and large fills in as a main issue of access for information stockpiling and access. Nearby customer hubs and remote PCs get to the capacity server through a GUI control board and FTP or through automatic API access by programming and applications. File stockpiling comprises of record frameworks that are provisioned to at least one servers in a NAS situation. A record framework is appended to servers with conventions, for example, Network File Systems (NFS) or Common Internet File System (CIFS) over ethernet. I/O access on a capacity framework is practiced through perusing or composing singular records of information. A File stockpiling server is a kind of server that is utilized to store, get to, secure and oversee computerized information, documents and administrations. It is a reason constructed server utilized for putting away and getting to little to vast measure of information over a mutual system or through the Internet. has more storage room, stockpiling access interfaces and specific information recovery and the executive's utilities. A capacity server for the most part fills in as a main issue of access for information stockpiling and access. Neighbourhood customer hubs and remote PCs get to the capacity server through a GUI control board and FTP or through automatic API access by programming and applications. Major aim of the proposed system is to restructure the user file ownership information and make it invisible from

all system entities. Here system will hide file index table information and manage all users file under same directory structure. File will not be searchable nor will the index table be having original file name references.

2. Literature survey

Till now 80 percent of work was carried in single cloud environment and much attention is not paid to multi-cloud environment. Our proposed system is based on secured cloud storage in multi-cloud environment using DFS. In Multi-Cloud trust, reliability, security is improved as it is distributed among various clouds which are placed globally and at any instance of time none of user will get complete set of data. This section specifies some work carried by researchers in similar area for secured cloud storage.

Revised Blakely's secret sharing mechanism is proposed to improve security and reliability of DFS without affecting scalability. Author has studied various secret sharing schemes which are used for sharing data in distributed environment. Each scheme focused on 2 parts as share creation and restoration of data. Assuming data is divided into S chunks and stored separately, it will require only T chunks (where T < S) to recover data and every data node will get less than T chunks. This scheme does not require key management. To reduce computation overhead in this scheme, Graphical processing unit is used. Even though cryptographic key management is not required in this approach but cost of implementation is increased due to GPU. Authors have tested this scheme in simple environment with 8 CPU core and actual testing is not done in cloud. Authors have not mentioned how DFS will communicate with other proprietary DFS in multi cloud environment.

Fan-Hsun et al. proposed secure and reliable cloud DFS using replacement of Hadoop DFS with open source based Tahoe least-authority file system. This system improves fault tolerance by recovering data even though some storage nodes are faulty. It is more secure as Tahoe-LAFS incorporates AES encryption.Tahoe–LAFS is very good open source system which is based on service provider independent security means security is fully managed by user at Saas level by using web interface .This system divides data chunks on 7 nodes and require minimum 3 nodes to recover data when some data nodes are in failure condition. This system is secure and reliable but authors have used already implemented Tahoe-LAFS and new DFS is not developed. This DFS is tested only on 4 simple storage nodes and no discussion was carried out by authors about use of system in actual single or multi-cloud environment and operations supported by File system.

In some research cloud storage service model for inter and intra cloud is proposed at Iaas level. Different data chunks of file are stored in various VMs of single or multi cloud. User can store and retrieve data from multiple cloud. System uses user authentication followed by file splitting / file retrieval by cloud manager interface and then it will be handover to multiple clouds/users. System supports both inter and intra cloud operations but security is not enough as encryption methodology is not used. Authors have not discussed different threats at IaaS level to virtual machine like cross-VM attack and side channel attack.

KhengKok Mar has introduced multiple cloud based secure virtual diffused file system by hosting it on exiting setup of public cloud. This system used information dispersal algorithm to divide data into multiple parts and diffused them in various clouds. It used registry server for managing metadata and data distribution. This DFS scheme supports random read/ write and streaming I/O operations. Even though system is good for operating at multi-cloud environment, it is using only data splitting architecture, one of four architecture proposed in research. Author has proposed optional encryption hence security issues may exist.

Our proposed system is similar but more efficient than system proposed research in which full authority will be given to system user / client at SaaS level using web interface. Various users can upload data to cloud based DFS at PaaS level which will be developed by us. We will develop our own DFS which is capable of handling user's data. User will have various options like splitting followed by encryption or vice-versa. We will be providing various encryption options to user for security of data. Our DFS will use proprietary based DFS on existing public cloud infrastructure to upload various encrypted chunks on multi-clouds. Since user data is available partially on each cloud, neither system administrator nor adversary will get full access to data thus resolving major security issues in secured data storage for single cloud environment.

3. Proposed system

This present invention discloses system and method for secure file storage using hidden owner identity mechanism. Currently with the increase in the creation and use of digital data, enormous amount of data is created on file storage server which is maintained by third party vendor hence there is need to provide secure system which will store all user's data securely on system where only authorized user knows complete information.

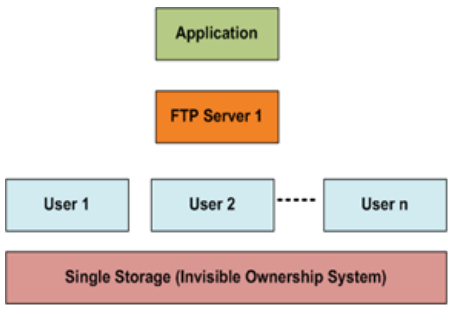

Fig. 1. System protocol architecture

Since file storage server stores all users data in various user directories, creates threat of user ownership recognition where

system administrator attack is possible. We have developed secure file storage system where file owner information user is hidden hence nobody including user, administrator and file storage server has complete information about stored data. Our system reduces indexing overheads and cumbersome security key management.

4. Conclusion

This paper proposed an overview on design and implementation of hidden identity mechanism for file storage server.

References

[1] Jens-Matthias Bohli, Nils Gruschka, Meiko Jensen, Luigi Lo Iacono, and Ninja Marnau, "Security and Privacy Enhancing Multi-Cloud Architectures", IEEE Transaction on Dependable and Secure Computing, Jan 2013.
[2] Zhifeng Xiao and Yang Xiao, "Security and Privacy in Cloud Computing", IEEE Communication Survey & Tutorials, Accepted for Publication, March 2012.
[3] Ayesha Malik, Muhammad Mohsin Nazir, "Security Framework for Cloud Computing Environment", Journal of Emerging Trends in Computing and Information Sciences, Vol. 3, No. 3, March 2012.
[4] Mukesh Singhal and Santosh Chandrasekhar, "Collaboration in Multicloud Computing Environments: Framework and Security Issues", Published by the IEEE Computer Society, 2013.
[5] Mohammed A. AlZain, Eric Pardede, Ben Soh, James A. Thom' "Cloud Computing Security: From Single to Multi-Clouds", International Conference on System Sciences, 2012.
[6] Kan Yang, Ren, Xiaohua Jia, Bo Zhang, and Ruitao Xie, "DAC-MACS: Effective Data Access Control for Multi-Authority Cloud Storage Systems," IEEE 2013.
[7] P. Mell and T. Grance, "The NIST definition of cloud computing," National Institute of Standards and Technology, Tech. Rep., Sept 2011.
[8] Jing-Jang Hwang and Hung-Kai Chuang, "A Business Model for Cloud Computing Based on a Separate Encryption and Decryption Service," National Science Council of Taiwan Government, IEEE 2012.
[9] J.-M. Bohli, M. Jensen, N. Gruschka, J. Schwenk, and L.L.L. Iacono, "Security Prospects through Cloud Computing by Adopting Multiple Clouds," Proc. IEEE Fourth Int'l Conf. Cloud Computing (CLOUD), 2011.
[10] M. Jensen, J. Schwenk, N. Gruschka, and L. Lo Iacono, "On Technical Security Issues in Cloud Computing," in Proceeding of IEEE Int'l Conf. Cloud Computing (CLOUD-II), 2009.3
[11] Kan Yang, Xiaohua Jia, "Attributed-based Access Control for Multi-Authority Systems in Cloud Storage," in Proceeding of 2012 32nd IEEE International Conference on Distributed Computing Systems, IEEE 2012.
[12] M. A. AlZain, B. Soh and E. Pardede," MCDB: Using Multi-Clouds to Ensure Security in Cloud Computing," in Proceeding of 2011 Ninth IEEE International Conference on Dependable, Autonomic and Secure Computing, IEEE 2011.
[13] Selvakumar G. Jeeva Rathanam M. R. Sumalatha, "PDDS - Improving Cloud Data Storage Security Using Data Partitioning Technique," IEEE 2012.
[14] Akash Kumar Mandal, Mrs. Archana Tiwari, "Performance Evaluation of Cryptographic Algorithms: DES and AES," in Proceeding of 2012 IEEE Students' Conference on Electrical, Electronics and Computer Science, IEEE 2012
[15] J. D Assistant Professor, Ramkumar P Systems Engineer, Kadhirvelu D," Preserving Privacy through Data Control in a Cloud Computing Architecture using Discretion Algorithm," in Proceeding of Third International Conference on Emerging Trends in Engineering and Technology, IEEE 2010.
[16] Prashant Kumar, Lokesh Kumar," Security Threats to Cloud Computing", International Journal of IT, Engineering and Applied Sciences Research, Volume 2, No. 1, December 2013.