Advances and Applications in Mathematical Sciences
Volume 20, Issue , December 2020, Pages 203-210

RESULT ANALYSIS OF HIDDEN IDENTITY MECHANISM FOR FILE STORAGE SERVER

PRADEEP KUMAR PATEL, CHANDU VAIDYA and PARWANI DHOTE

RGCER, Nagpur
E-mail: pryadeep@gmail.com
chandu.nyss@gmail.com
parwanidhote14@gmail.com

Abstract

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.

Introduction

Almost in every well know file storage server there is arrangement of separating the user files in user specific folder or directory which is normally named with user identification or user name. This makes folders vulnerable and searchable if hacker is looking for specific user's files. Almost in every file storage server, it has a user management by allotting different folders to every user, but which creates a threat of user ownership identification,

2010 Mathematics Subject Classification: 03F60, 60K30, 68P20.
Keywords: PEEK composite, aluminum alloy, un-sprung mass and wheel rim.
Received May 20, 2020; Accepted July 31, 2020

where by looking at server configuration one can identify the owner of the files. Proposed system is Invisible Ownership 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.

Background

Looking toward the limitations, described in the literature review, it is necessary to have a novel universal procedure to overcome those limitations. As almost every big organization having web identity and runs on hosting or cloud environment, the organizational information becomes important aspect. Even after cloud provides industry a wide range of security and system free from malicious software virus attacks, still the data is not safe from malicious users having administrative rights. One having super user rights can access the data from cloud storage with wrong intensions. After going through different solutions to reduce malicious user attack we found keeping the data out of user reach will make data more secure than by any other way, but we can't forget the fact we have to keep the data somewhere. Hence the proposed system is designed by keeping these scenarios in consideration where we will distribute and secure the data at different location in order to hide original data directly from user. Here system will take smart decision on the basis of user request and split the data after successful encryption in to different blocks and store it on the multiple cloud storage, now malicious user can access the data but of no use. Another point of consideration is the safety of data in case of critical condition. The

proposed system is less with capability to re calculate the data from rest of the storage if any one of the storage gets failure

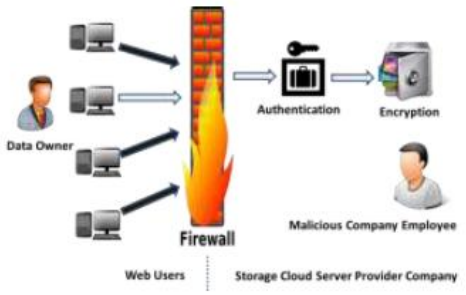

Figure 1.0. Security breaches.

Different mechanism including authentication using credentials, defining file ownership in order to restrict the file access, firewalls, encryption mechanism are available to protect information. It is unfortunate that none of the stated method guaranteed the complete security of the data and even after providing high security the major threat to the data on public cloud is malicious user figure 2.0 state the same issue where user can encrypt the information over cloud in secure environment but at the same time these storage locations are always accessible to the malicious user who is part of the public cloud service provider infrastructure. This thread is not easily traceable and no direct solution can be provided to this issue. *D*-HOM system mainly aims at hiding data and its ownership identity to make data inaccessible to malicious user which leads to securing the confidential data. Basic idea is to split file *F* in multiple parts *F*1, *F*2 and *F*3 then encrypting in to *F*1e, *F*2e and *F*3e respectively finally "PUSH" it to public cloud storage. Almost in every well know file storage server there is arrangement of separating the user files in user specific folder or directory which is normally named with user identification or user name. This makes folders vulnerable and searchable if hacker is looking for specific user's files. Take an example of file storage server, file storage service has a user management by allotting different folders to every user, but which creates a biggest threat of user ownership identification where by looking at service configuration one can identify the owner of the files. Part of the proposed system is Invisible Ownership System. Here no entity including file storage service has information about files stored on the storage. 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 we need to replace complete file storage service access on the entire cloud service provider. For the proof of concept, we can rebuild file storage service and demonstrate it over intranet. Now how it will work, first client will pass the file to file storage service along with some Key, file storage service will then rename the file with unique identification number or key which is generated by user key and private key or something like that. Once this new renamed file stored on the storage owner identity will be lost, no system will be there to identify, which file contains what and who is the owner of the file. See how useful it will be if we split then hide the identity of file. Every time while reading or accessing the file or file part system will calculate or generate the name of the file and access it.

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 users' data securely on system where only authorized user knows complete information.

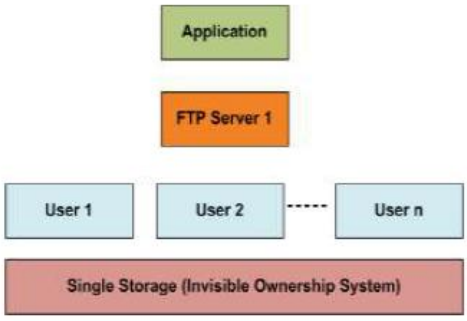

Figure 1.0. System Protocol Architecture.

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.

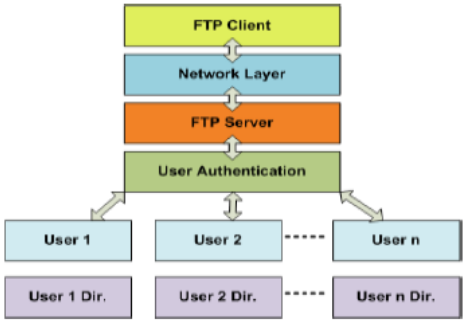

Figure 2.0. Existing FTP process.

Invisible Ownership System

- Proposed New Method /Algorithm for improving FTP services
- Header Deletion function will remove ownership details of file chunks to provide OS level security
- Proposed system will dynamically create naming convention ambiguity by renaming each file
- New name is generated using Filename, Server name, user key and cloud private key

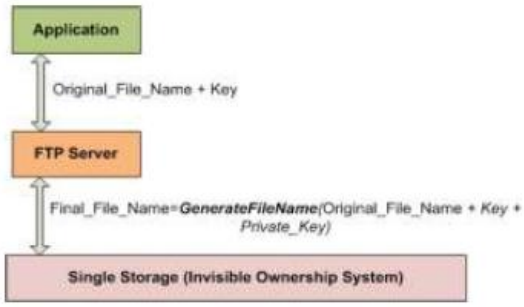

Figure 3.0. System process.

This present disclosure relates to the novel work carried for implementation of a Hidden Owner Identity mechanism for securely storing file on storage server. Moreover, it relates to unique, efficient and secure method for storing a file on any type of storage server either local or cloud-based server. Furthermore, present disclosure relates to computer program product consisting of computer -readable instructions stored on computer -readable storage media. These instructions being executed by computing system consisting of processing hardware to execute programs. File storage server is a server which stores various types of critical user data files and privacy-sensitive information using file systems therefore they are main targets for various types of security attacks.

Implementation

The sequence diagram represents timeline over the file processing in our implementation. The first step is the registration of client over the application, generate the login credentials to make the request for the verification and authentication of client to start file processing over the developed environment. After getting all the credentials for file processing client is allow to start working over the environment. Client selects a file to be uploaded and generate a secure key to make a request to upload a file into the available cloud. The next step calls the Encryption and splitting API's to perform the secure distribution of file over the cloud. The timeline process call also be called in reverse to get file receive request made satisfaction of the client.

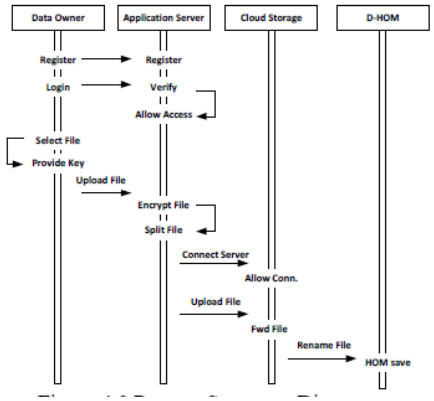

Figure 4.0. Process Sequence Diagram.

Algorithm: Save file (HOM)

1. Get File → FL
2. Get Key → K
3. Generate GUID → G
4. Rename FL=Enc (Gk)
5. CreateFile (FP)
6. Write FL→ FP
7. Save FP → CentralStorage

Conclusion

In presented proposed system we analysed different aspects of cloud computing and different file storage mechanism including local and distributed file system. Across the development we studied different available solutions and tried to find out what parameters can be considered and re-designed to enhance the performance or minimize the cost factor. We carried out the development of *D*-HOM (Distributed Hidden Ownership Mechanism) system which will enhance the file security by encrypting and distributing the file parts over multiple cloud and re-join when accessing. With simulation and implementation, we generated the performance result and resulting parameters are enhanced.

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 3(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 and 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 and 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 and M. R. Sumalatha, PDDS- Improving Cloud Data Storage Security Using Data Partitioning Technique, IEEE, 2012.

[14] Akash Kumar Mandal and 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. Ramkumar P. Systems Engineer and 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 and Lokesh Kumar, Security Threats to Cloud Computing, International Journal of IT, Engineering and Applied Sciences Research (IJIEASR), Volume 2, No. 1, December 2013.