ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 2, pp: (485-488), Month: April - June 2015, Available at: www.researchpublish.com

Formation of Data Storage with Identity Based Secure Distributed Plan of Action Schemes

Bouramma.Sadashiv.Varad

Department of Computer Science and Engineering, Visvesvararya Technological University, Belagavi, Belgavi, India

Abstract: **We are using cloud computing for data storage then secure distributed data storage can shift the burden of maintaining a larger number of files from proxy server. The proxy server are one which convert encrypted files for the owner to encrypted files for the receiver without the necessity of knowing content of the original file. In some situation original file will be removed but the owner for the sake of efficiency hence we are concentrate confidentiality and integrity of the outsourced data will be addressed carefully. It contain following properties**

1. **The file owner can decide the access permission independendently without the help of the private key generator.**
2. **For one query, a receiver can only access one file, instead of all files of the owner.**
3. **We are providing secure against the collusion attacks.**
4. **Maintaining chosen plaintext attacks and chosen cipher text attacks.**

Keywords: **data storage, server, encrypted files, original file, integrity.**

1. INTRODUCTION

Cloud computing provide different types of services such as database as a service ,platform as service, software as a service, infrastructure as a sevice,desktop as a service, database as a service is one which that manages owner personal file with convenient mechanism. In database as a service schemes, owner can send his encrypted file files to proxy server. A proxy server is a computer or usually set of computer that acts as an intermediate between a client computer and web server, it enables client computer to make indirect requests services such as web page, video, pdf files, etc.

The working of proxy server is a client contact the proxy server, requests a file, web page or other resource from a different server. If the proxy server has the requested file or web page in cache, the proxy server returns the file or web page to the client .otherwise, it connects to the requested server, provide the client with the file or web pages, and then save it in the cache for use later.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 2, pp: (485-488), Month: April - June 2015, Available at: www.researchpublish.com

Example of proxy server:

The Alice acts as a client computer, proxy server, and Bob acts as a web server then Alice can send request to bob using proxy server as mediator without having to contact him directly. Bob responds to client request send required data to proxy server, proxy server transferred data to the Alice then finally Alice will get required data.

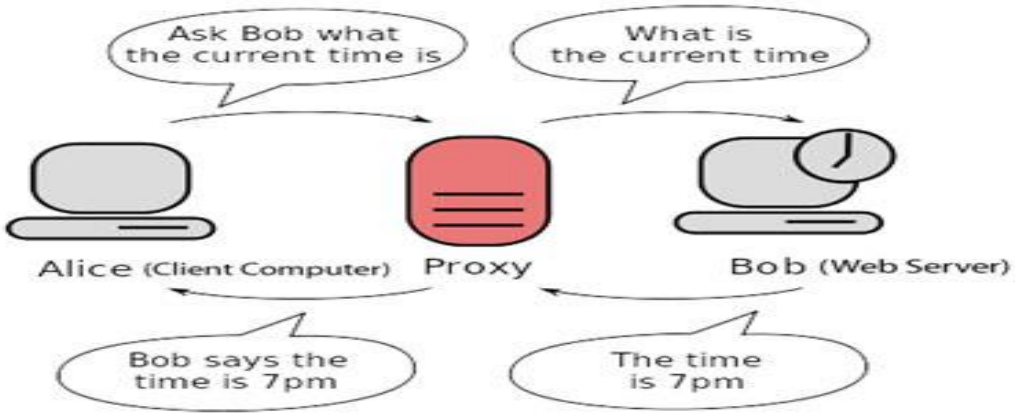


Figure: two computer are connected through a third computer (shown in red) acting as a proxy, communicate with each other (shown in grey).

The user are concern on confidentiality, integrity, sending query file to cloud computing .confidentiality is one which that is used for prevent unauthorized users from accessing sensitive data. Integrity is one which that is used for prevent modified and replacing of data .query files are stored in data storage is executed in between a proxy server and receiver.

Problem statement: Users are especially concerned on the confidentiality, integrity and query of the outsourced files as cloud computing is a lot more complicated than the local data storage systems, as the cloud is managed by an untrusted third party.

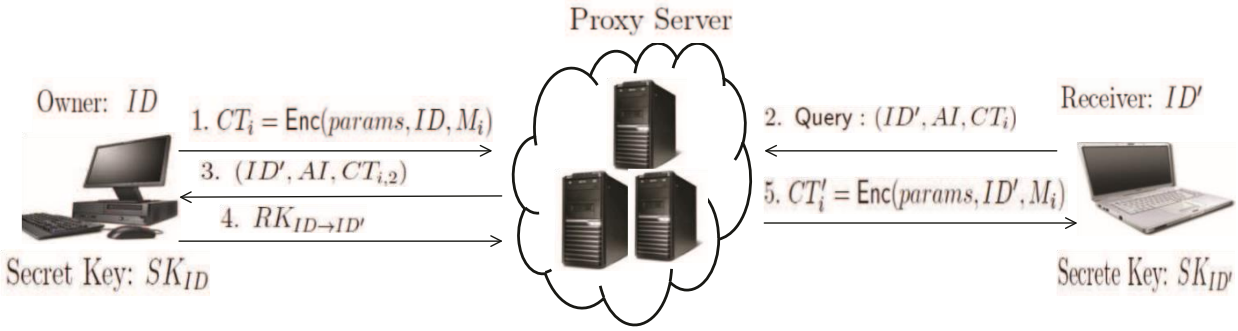
2. OBJECTIVES AND MOTIVATION

Secure distributed data storage can shift the burden of maintaining a large number of files from the owner to proxy servers. Proxy servers can convert encrypted files for the owner to encrypted files for the receiver *without* the necessity of knowing the content of the original files.

3. SYSTEM ARCHICTURE

It contains the following parts

1. DATA _OWNER
2. PROXY SERVER
3. USER
4. DATA STORAGE SYSTEMS



Data Owner: in this module, first the new data owner registers and get a valid login credentials. After login section, the data owner has permission to upload their file to proxy server. Data owner encrypts his data and outsource the cipher text to the proxy servers.

proxy server: in this module, proxy server store the encrypted data and transfer the cipher text for the owner to cipher text for receiver when they obtain an access permission from owner .they authenticate receivers and validate access permissions.

User: user authenticates himself to the owner and decrypts the encrypted cipher text to obtain data. In these systems, an end-to-end security is provided by cryptographic protocols. These systems are divided into two types namely shared file system and non-shared file system.

Data storage systems: data storage systems enable user to store their data to external proxy servers to enhance the access and availability and reduce maintenance cost.

4. SYSTEM DISCRIPTION

1 Data Storage Systems

2 Networked File Systems.

3 Storage-based Intrusion Detection Systems.

4 Cryptographic File System.

Data Storage Systems:

Data storage systems enable users to store their data to external proxy servers to enhance the access and availability, and reduce the maintenance cost. The privacy issues in data utility, and pointed out the main research directions in the protection of the externally stored data.

Networked File Systems:

In these systems, proxy servers are assumed to be trusted. They authenticate receivers and validate access permissions. The interactions between the proxy servers and receivers are executed in a secure channel. Therefore, these systems cannot provide an end-to-end data security, namely they cannot ensure the confidentiality of the data stored at the proxy server in these schemes, and a receiver authenticates himself to the proxy server using his password. Then, the proxy sever passes the authentication result to the file owner. The owner will make an access permission according to the received information.

Storage-based Intrusion Detection Systems:

In these systems, an intrusion detection scheme is embedded in proxy servers or the file owner to detect the intruder's behaviours, such as adding backdoors, inserting Trojan horses and tampering with audit logs. These schemes can be classified into two types: host-based system and network-based system. In the host-based systems, an intrusion detection scheme is embedded in the host to detect the local intrusion actions. On the contrary, in network-based systems, an intrusion detection scheme is embedded in the proxy servers to detect the external intruder's actions. The main advantage of these systems is that proxy servers can still detect the intrusion action seven if the host is compromised as the proxy server are independent from the host.

Cryptographic File System:

In these systems, an end to-end security is provided by cryptographic protocols which are executed by the file owner to prevent proxy servers and unauthorized users from modifying and accessing the sensitive files. These systems can be divided into two types: shared file system and non-shared system. In shared file systems the owner can share his files with a group of users. Cryptographic techniques deployed in these systems are key sharing, key agreement and key revocation. In non-shared file systems in order to share a file with another user, the owner can compute an access key for the user using his secret key. In these two systems, the integrity of the sensitive files is provided by digital signature schemes and message authentication codes (MAC).

5. RELATAED WORK

In this standard model where for one query, the receiver can only access one of the owner files he will not take all the file without the owner permission. We can also protect against collusion attacks.to achieve stronger security and access control.

6. CONCLUSION

Formation of data storage with identity based secure distributed plan of action in standard model where, for one query, the receiver can only access one file, instead of all files. Furthermore, the access permission can be made by the owner, instead of the trusted party. Our schemes are secure against the collusion attacks.

7. ACKNOWLEDGMENT

I feel great pleasure to acknowledge the guidance and assistance of all those people who have made work on this project endeavor and I thankful anonymous references for helpful suggestion.

REFERENCES

[1] H. Hacig¨um¨us, B. R. Iyer, C. Li, and S. Mehrotra, “Executing SQL over encrypted data in the database-service-provider model,” in Proceedings: SIGMOD Conference - SIGMOD’02 (M. J. Franklin,B. Moon, and A. Ailamaki, eds.), vol. 2002, (Madison, Wisconsin,USA), pp. 216–227, ACM, Jun. 2002.

[2] L. Bouganim and P. Pucheral, “Chip-secured data access: Confidentialdata on untrusted servers,” in Proc. International Conference on Very Large Data Bases - VLDB’02, (Hong Kong, China), pp. 131–142, Morgan Kaufmann, Aug. 2002.

[3] U. Maheshwari, R. Vingralek, and W. Shapiro, “How to build a trusted database system on untrusted storage,” in Proc. Symposium on Operating System Design and Implementation- OSDI’00, (SanDiego, California, USA), pp. 135–150, USENIX, Oct. 2000.

[4] A. Ivan and Y. Dodis, “Proxy cryptography revisited,” in Proc.Network and Distributed System Security Symposium - NDSS’03, (SanDiego, California, USA), pp. 1–20, The Internet Society, Feb. 2003.

[5] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, “Improved proxy re-encryption schemes with applications to secure distributed storage,” in Proc. Network and Distributed System SecuritySymposium - NDSS’05, (San Diego, California, USA), pp. 1–15, The Internet Society, Feb. 2005.

[6] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, “Improved proxy re-encryption schemes with applications to secure distributed storage,” ACM Transactions on Information and SystemSecurity, vol. 9, no. 1, pp. 1–30, 2006.

[7] S. D. C. di Vimercati, S. Foresti, S. Paraboschi, G. Pelosi, and P. Samarati, “Effficient and private access to outsourced data,” in Proc. International Conference on Distributed Computing Systems- ICDCS’11, (Minneapolis, Minnesota, USA), pp. 710–719, IEEE, Jun. 2011.

[8] H.-Y. Lin and W.-G. Tzeng, “A secure erasure code-based cloud storage system with secure data forwarding,” IEEE Transactionson Parallel and Distributed Systems, Digital Object Indentifier 10.1109/TPDS.2011.252 2012.