Proficient Public Verification of Data Reliability for Cloud Storage with Two Phase Protection

Anand Krupa Harishbhai[1], P. Nagadevi[2], R. Uma Maheswari[3], X.Nancy[4]
UG Student, Department of Computer Science and Engineering, Arjun College of Technology, Coimbatore, Tamilnadu, India. Email-id:
UG Student, Department of Computer Science and Engineering, Arjun College of Technology, Coimbatore, Tamilnadu, India. Email-id:
UG Student, Department of Computer Science and Engineering, Arjun College of Technology, Coimbatore, Tamilnadu, India. Email-id:
Assistant Professor, Department of Computer Science and Engineering, Arjun College of Technology, Coimbatore, Tamilnadu, India.

Abstract— **The cloud security is one of the important roles in cloud, here we can preserve our data into cloud storage. More and more clients would like to store their data to PCS (public cloud servers) along with the rapid development of cloud computing. Cloud storage services allow users to outsource their data to cloud servers to save local data storage costs. Multiple verification tasks from different users can be performed efficiently by the auditor and the cloud-stored data can be updated dynamically. It makes the clients check whether their outsourced data is kept intact without downloading the whole data. In our system we are using the own auditing based on the token generation. Using this key generation technique compare the key values from original keys we can find out the changes about the file. A novel public verification scheme for cloud storage using in distinguishability obfuscation, which requires a lightweight computation on the auditor and delegate most computation to the cloud. Not only stored also the content will be encrypted in the cloud server. If anyone try to hack at the cloud end is not possible to break the two different blocks. The security of our scheme under the strongest security model. They need first decrypt the files and also combine the splitted files from three different locations. This is not possible by anyone. Anyone can download the files from the server with file owner permission. At the time of download key generated (code based key generation) and it will send to the file owner. We can download the file need to use the key for verification and some other users want to download file owner permission is necessary.**

I. INTRODUCTION

Distributed computing has been imagined as the following creation data innovation (IT) design for undertakings, because of its extensive rundown of unparalleled preferences in the IT history: on-request self-benefit, omnipresent system get to, area self-deciding asset pooling, fast asset versatility, utilization based estimating and transference of hazard. As a disturbing innovation with significant ramifications, distributed computing is changing the very way of how organizations utilize data innovation. One essential part of this outlook changing is that information are being brought together or outsourced to the cloud. From clients' view, including together people and IT endeavors, putting away information remotely to the cloud in an adaptable on-request technique bring appealing advantages: arrival of the weight for storage room administration, boundless information access with place autonomy, and evasion of assets expenses on equipment, programming, and staff systems of support, and so on While distributed computing make these remuneration more engaging than any other time in recent memory, it additionally conveys new and testing security dangers to clients' outsourced information. As cloud administration suppliers (CSP) are part regulatory elements, information outsourcing is really surrendering client's last control more than the destiny of their information. As a matter of first importance, despite the fact that the frameworks beneath the cloud are significantly more effective and dependable than individual registering gadgets, they are still before the extensive variety of dangers for information respectability. As high-speed networks and ubiquitous Internet access become available in recent years, many services are provided on the Internet such that users can use them from anywhere at any time. Data robustness is a major requirement for storage systems. There have been many proposals of storing data over. As a disruptive technology with profound implications, cloud computing is transforming the very nature of how businesses use information technology. One fundamental aspect of this paradigm shifting is that data Are being centralized or outsourced to the cloud. From users' perspective, including both individuals and IT enterprises, storing data remotely to the cloud in a flexible on-demand manner brings appealing benefits: relief of the burden for storage management, universal data access with location independence, and avoidance of capital expenditure on hardware, software, and personnel maintenances, etc.

Cloud service providers (CSP) are separate administrative entities, data outsourcing is actually relinquishing user's ultimate control over the fate of their data. As a result, the correctness of the data in the cloud is being put at risk due to the following reasons. First of all, although the infrastructures under the cloud are much more powerful and reliable than personal computing devices, they are still facing the broad range of both internal and external threats for data integrity.

II. Related Work

In public cloud environment, most clients upload their data to PCS and check their remote data's integrity by Internet. When the client is an individual manager, some practical problems will happen. The computation overhead of verification by the auditor linearly increases with the size of the verified data set. Here third party public auditing scheme for the regenerating-code-based cloud storage. To solve the regeneration problem of failed authenticators in the absence of data owners, if these data cannot be processed just in time, the manager will face the loss of economic interest. In order to prevent the case happening, the manager has to delegate the proxy to process its data. In PKI (public key infrastructure), remote data integrity checking protocol will perform the certificate management. When the manager delegates some entities to perform the remote data integrity checking, it will incur considerable overheads since the verifier will check the certificate when it checks the remote data integrity. The cryptographic techniques for the purpose of data security protection cannot be directly user's control. Therefore, verification of correct data storage in the cloud must be conducted without explicit knowledge of the whole data. Considering various kinds of data for each user stored in the cloud and the demand of long term continuous assurance of their data safety, the problem of verifying correctness of data storage in the cloud becomes even more challenging. This is not just a third party data warehouse. The data stored in the cloud may be often updated by the users.

III. Proposed Method

An efficient distributed scheme with data in the cloud is been made. Here we are using the erasure code technique for distribute the data to cloud locations and access the data from cloud. User can register and login into their account. Provided an option to store, share and access the data from cloud storage. Here we are using the double ensured scheme for storing data into the cloud. First is your data or file splitted into multiple parts and it will store into different cloud server locations. Each and every file generates the key-code for auditing. Then Second is each and every splitted file will encrypt before store into different locations. The shared users can edit the file in the cloud with file owner's permission. That file eligible of own public auditing. Search and download the files, at the time of download user should use the security key. As an authentication success it will be decrypt and combine to get the original data from cloud. Moreover, we design a novel public verifiable authenticator, which is generated by a couple of keys and can be regenerated using partial keys. Thus, our scheme can completely release data owners from online burden. In addition, we randomize the encode coefficients with a pseudorandom function to preserve data privacy. Extensive security analysis shows that our scheme is provable secure under random oracle model and experimental evaluation indicates that our scheme is highly efficient and can be feasibly integrated into the regenerating code- based cloud storage. Compared to a lot of its predecessors, which only provide binary results about the storage state across the distributed servers, the challenge-response protocol in our work more provides the localization of data error. Unlike most prior works used for ensuring remote data integrity, the new scheme supports secure and efficient dynamic operations on data blocks, including: update, delete and append.Extensive protection and act analysis demonstrate that the proposed scheme is extremely efficient and resilient beside Byzantine failure, malicious data modification attack, and even server colluding attacks.

The overall flow of our proposed human activity analysis framework is depicted in the figure 1.

1. DFD diagram for user registration and login.

   While getting registered user have to submit only basic information which includes mainly unique mail-id, which should be remembered by user. Because that is going to play an important role in means of communication as well as getting accessed to their profile interface, which is created at the time of user getting registered.

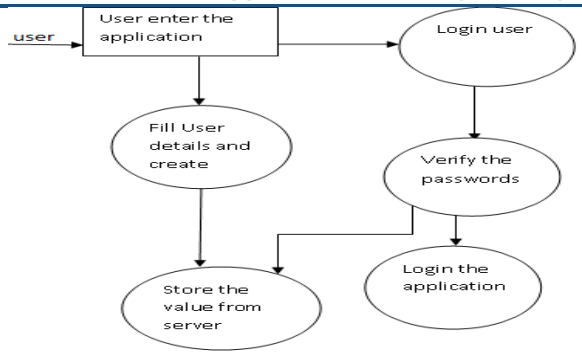

Figure 1 Registration and Login

The Uploading process flow is depicted in the figure 2.

The key components of our proposed framework are,

• User Interface
• File Uploading Process
• Secret Key Generation
• File Sharing Process
• File Auditing Process
• Mail Alert Process
• File Downloading Process

A. User interface:

In our Secure System we have a user friendly user interface to interact with our System. Every Act dual role as a data owner and data consumer while uploading file they are the owner of that file if they search other's file than they are the consumer. Users can create the account them self for that we have new pages, in that page we will get the details from the user and we generate the account for the user's. We have authentication system; we only allow authorized users to access our System. In our System we providing the easy file searching user's don't want to keep remember all uploaded file's exact name, for that we have given the keywords while uploading the files it will help to search the file easily.

B. File uploading process:

Storing data over storage servers one way to provide data robustness is to replicate a message such that each storage server stores a message. Another way is to encode a message of k symbols into a code-word of n symbols by erasure coding. To store a message, each of its code-word symbols is stored in a different storage server. A storage server corresponds to an erasure error of the code-word symbol. As long as the number of servers is under the tolerance threshold of the erasure code, the message can be recovered from the code-word symbols stored in the available storage servers by the decoding process. While uploading the file, it is splitted into sub-blogs by using erasure code technique. After getting splitted into sub-blogs it is encrypted using MD5 Algorithm and then stored at different locations into a cloud server. The block diagram of encryption algorithm is shown below.

2. DFD diagram for user uploading process:

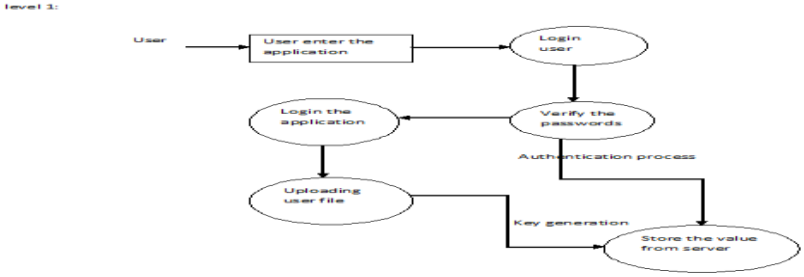

Figure 2 Uploading Process

3. MD5 Block Diagram:

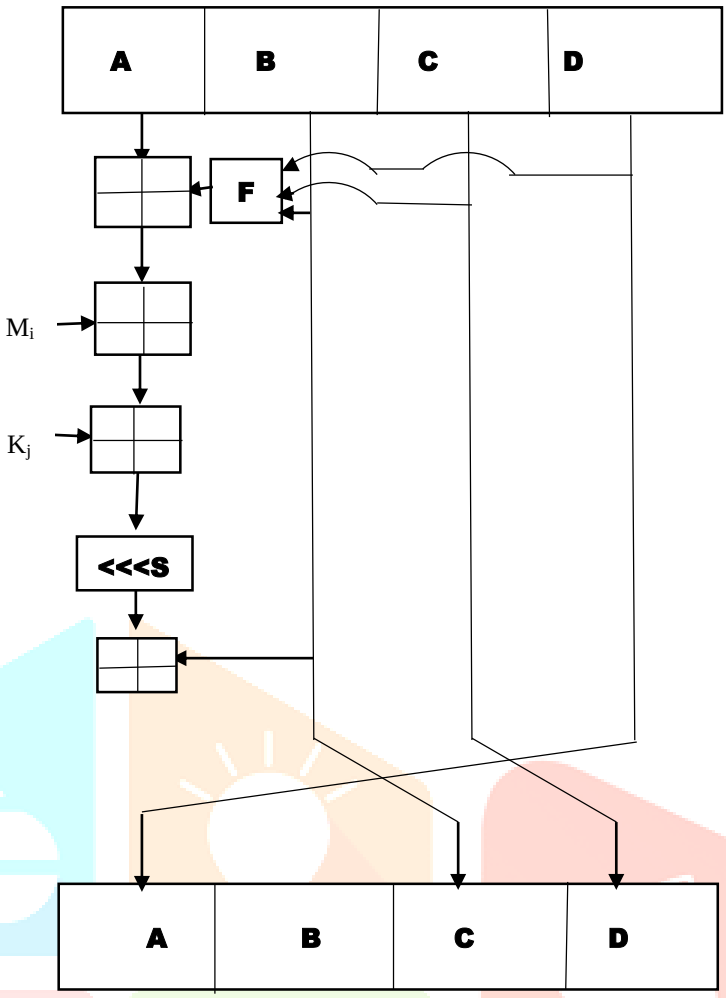

Figure 3 MD5 Algorithm

F (B, C, D) = (B ^ C) V (¬ B ^ D)

G (B, C, D) = (B ^ D) V (C ^ ¬D)

H (B, C, D) = B + C+ D

I (B, C, D) = C+ (B V ¬D)

C. Secret key generation:

Firstly the secret key will be generated as the initial step while uploading the file, every file which is uploaded, will have unique secret key. This key will be taken as an identification of every file. The secret key which we are using is a number we will make it use for both uploading and downloading. If the user want to download some file and if he gives the download request the secret key of that file will be sent to the file owner of the file maybe he or she can share it with the user who has requested for that password. Further file owner can send the specific secret key to the person who has requested through any means of communication.

D. File Sharing:

In our application we can share a file to a registered user by providing basic credentials, with the sharing option it is necessary to provide authority to the shared user whether to view or edit the file. A user can view the shared file within the application without downloading it and the same is possible with the edit option. If the file owner wants to share the file with any other registered user then the owner can give the unique id of that user while uploading the file, which will enable that user to view the file and even edit the file if he or she wants without downloading that particular file. If the user thinks to suggest some content the he or she can edit and upload the file. The uploaded file will get

uploaded temporary and in the main database file will not get modified. It will get modified only when the file owner will approve it if needed.

E. File Auditing:

Auditing is the process of checking the file whether the original contents of the file is changed. This module provides the file owner auditing, this we achieve by generating tokens. The tokens are generated with the ASCII values of the characters in the file and these characters are stored in the DB while uploading the file. If a shared user edit's the file and saves it, again a new token will be generated and stored in the DB. If the initial token and the current token aren't same then a notification will be sent to the file owner. In this session, file owner can checks whether the other registered users had viewed or edited the file or not. Even file owner can know about the particular user who has accessed the file by their unique id.

F. Mail alert process:

The uploading and downloading process of the user is first get the secret key in the corresponding user email id and then apply the secret key to encrypted data to send the server storage and decrypts it by using his secret key to download the corresponding data file in the server storage system's the secret key conversion using the Share Key Gen (SKA, t, m). This algorithm shares the secret key SKA of a user to a set of key servers. The user who has registered their mail id at the time of getting registered through that email id only user will be able to exchange the information through means of communication, Otherwise user has to get registered again and go through the verification process once more as before.

G. File Downloading process:

File downloading process is to get the corresponding secret key to the corresponding file to the user mail id and then decrypt the file data. The file downloading process re-encryption key to storage servers such that storage servers perform the re-encryption Operation. The length of forwarded message and the computation of re-encryption is taken care of by storage servers. Proxy re-encryption Schemes significantly reduce the overhead of the data Forwarding function in a secure storage system.

V. CONCLUSION AND FUTURE WORK

A privacy-preserving public auditing system for data storage security in cloud computing. We utilize the homomorphism linear authenticator and random masking to guarantee that the TPA would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process, which not only eliminates the burden of cloud user from the tedious and possibly expensive auditing task, but also alleviates the users' fear of their outsourced data leakage. Considering TPA may concurrently handle multiple audit sessions from different users for their outsourced data files, we further extend our privacy-preserving public auditing protocol into a multiuser setting, where the TPA can perform multiple auditing tasks in a batch manner for better efficiency.

We further extend our privacy-preserving public auditing protocol into a multi user setting, where the TPA can perform multiple auditing tasks in a batch manner for better efficiency. In prospective we will improving the performance. In this system we used only text files, In future we will include the image, audio, video files. In our system the OTP sent to owner mail id, coming up the client will receive the OTP on mobile by using the mobile number.

REFERENCES

[1] .Y. Cui, Z. Lai, X. Wang, N. Dai, and C. Miao, "Quicksync: Improving synchronization efficiency for mobile cloud storage services," in Proceedings of MobiCom. ACM, 2015, pp. 592–603.

[2] .H. Li, Y. Yang, T. H. Luan, X. Liang, L. Zhou, and X. Shen, "Enabling fine-grained multi-keyword search supporting classified sub-dictionaries over encrypted cloud data," IEEE Transactions on Dependable and Secure Computing, vol. 13, no. 3, pp. 312–325, 2016.

[3] .Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling public auditability and data dynamics for storage security in cloud computing," IEEE Transactions on Parallel and Distributed Systems, vol. 22, no. 5, pp. 847–859, 2011.

[4] .M. Sookhak, A. Gani, H. Talebian, A. Akhunzada, S. U. Khan, R. Buyya, and A. Y. Zomaya, "Remote data auditing in cloud computing environments: A survey, taxonomy, and open issues," ACM Computing Surveys, vol. 47, no. 4, 2015.

[5] .G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in Proceedings of CCS. ACM, 2007, pp. 598–609.

[6] H. Li, D. Liu, Y. Dai, and T. H. Luan, "Engineering searchable encryption of mobile cloud networks: When qoe meets qop," IEEE Wireless Communications, vol. 22, no. 4, pp. 74–80, 2015.

[7] K. Yang and X. Jia, "An efficient and secure dynamic auditing protocol for data storage in cloud computing," IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 9, pp. 1717–1726, 2013.

[8] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in Proceedings of INFOCOM. IEEE, 2010, pp. 19.

[9] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in Proceedings of SecureComm. ACM, 2008.

[10] S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters, "Hiding secrets in software: A cryptographic approach to program obfuscation," Communications of The ACM, vol. 59, no. 5, pp. 113– 120, 2016.

[11] .A. Juels and B. S. K. Jr, "Pors: Proofs of retrievability for large files," in *Proceedings of CCS*. ACM, 2007, pp. 583–597.

[12] J. Ni, Y. Yu, Y. Mu, and Q. Xia, "On the security of an efficient dynamic auditing protocol in cloud storage," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 10, pp. 2760–2761, 2013.