Journal of **N**onlinear **A**nalysis and **O**ptimization
Vol. 13, No. 1, (2022),
ISSN : **1906-9685**



ENABLING IDENTITY-BASED INTEGRITY AUDITING AND DATA SHARING WITH SENSITIVE INFORMATION HIDING FOR SECURE CLOUD STORAGE

Prof. C.P Lachake Department of Computer Engineering (SKN Sinhgad Institute of Technology and Science) (Affiliate to S.P.P.U.) Pune, India Cplachake.sknsits@sinhgsd.edu
Shinde Balaji Department of Computer Engineering (SKN Sinhgad Institute of Technology and Science) (Affiliate to S.P.P.U.) Pune, India shindebalaji0203@gmail.com
Ingle Shrikant Department of Computer Engineering (SKN Sinhgad Institute of Technology and Science) (Affiliate to S.P.P.U.) Pune, India shrikantingle188@gmail.com
Vedant Barapatre Department of Computer Engineering (SKN Sinhgad Institute of Technology and Science) (Affiliate to S.P.P.U.) Pune, India Vedantbarapatre23@gmail.com
Shinde Shradha Department of Computer Engineering (SKN Sinhgad Institute of Technology and Science) (Affiliate to S.P.P.U.) Pune, India Shradhas380@gmail.om

Abstract
A data storage server, such as a cloud server, can demonstrate to a verifier that it is honestly storing the data of a data owner by using remote data integrity checking (RDIC). Numerous RDIC protocols have been put out in the literature to this point, however the most of these designs have a sophisticated key management problem, meaning they depend on pricy public key infrastructure (PKI), which could make it difficult to implement RDIC in practise. In order to simplify the system and lower the cost of setting up and maintaining the public key authentication framework in PKI based RDIC schemes, we suggest a novel architecture of the identity-based (ID-based) RDIC protocol in this study. We formalise ID-based RDIC, along with its security model, which includes protection from rogue cloud servers and zero knowledge privacy from a third-party verification.
During the RDIC procedure, the proposed ID-based RDIC protocol does not reveal any information about the stored data to the verifier. The new design achieves zero knowledge privacy against a verifier and is demonstrated to be secure against the malicious server in the general group model. Extensive security research and implementation results show that the suggested protocol is practicable in real-world applications and provably secure. We Extend This Work with Group Management, Forward and Backward Secrecy by Time Duration, and File Recovery When Data Integrity Checking Fault Occurs.
Cloud computing, which has received considerable attention from research communities in academia as well as industry, is a distributed computation model over a large pool of shared-virtualized computing resources, such as storage, processing power, applications and services.
Cloud users are provisioned and release recourses as they want in cloud computing environment. This kind of new computation model represents a new vision of providing computing services as public utilities like water and electricity. Cloud computing brings a number of benefits for cloud users. However, there is a vast variety of barriers before cloud computing can be widely deployed. A recent survey by Oracle referred the data source from international data corporation enterprise panel, showing that security represents 87% of cloud users' fears1. One of the major security concerns of cloud users is the integrity of their outsourced files since they no longer physically possess their data and thus lose the control over their data.
Moreover, the cloud server is not fully trusted and it is not mandatory for the cloud server to report data loss incidents. Indeed, to ascertain cloud computing reliability, the cloud security alliance (CSA) published an analysis of cloud vulnerability incidents.

Objectives
There are primarily four goals that summaries how this project will work in its entirety. To develop an automated model that is both affordable and capable of monitoring the moisture content of a soil sample, primarily to meet the needs of farmers who do not have access to technology. Determine the impact of fertilizers and soil nutrients on agriculture. Improve the effectiveness of soil conservation

methods and raise public awareness of soil degradation. Test the viability of indigenous sensors (resistance blocks) as opposed to commercially available sensors.

I. Work related to the project

The Far site distributed file system provides availability by replicating each file onto multiple desktop computers. Since this replication consumes significant storage space, it is important to reclaim used space where possible. Measurement of over 500 desk top file systems shows that nearly half of all consumed space is occupied by duplicate files. 2 We present a mechanism to reclaim space from this incidental duplication to make it available for controlled file replication. Our mechanism includes

1) convert gent encryption, which enables duplicate files to coalesced into the space of a single file, even if the files are

encrypted with different users' keys, and 2) SALAD, a Self Arranging, Lossy, Associative Database for aggregating file content and location in formation in a decentralized, scalable, fault-tolerant manner. Large-scale simulation experiments show that the duplicate-file coalescing system is scalable, highly effective, and fault-tolerant.

II. System Information

Reclaiming Space from Duplicate Files in a Serverless Distributed File SystemThe Far site distributed file system provides availability by replicating each file onto multiple desktop computers. Since this replication consumes significant storage space, it is important to reclaim used space where possible.

Measurement of over 500 desktop file systems shows that nearly half of all consumed space is occupied by duplicate files. We present a mechanism to reclaim space from this incidental duplication to make it available for controlled file replication. Our mechanism includes

1) convergent encryption, which enables duplicate files to coalesced into the space of a single file, even if the files are encrypted with different users' keys, a

2) SALAD, a Self Arranging, Lossy, Associative Database for aggregating file content and location information in a decentralized, scalable, fault- tolerant manner. Large-scale simulation experiments show that the duplicate-file coalescing system is scalable, highly effective, and fault-tolerant.

A. Existing Systems and their Advantages or

Disadvantages

Advantages:

- By using block level deduplication reduce the storage space
- Reliability issue is overcome
- Distributed Deduplication System is performed

Disadvantages: -

- To solve reliability problem by distributed deduplication system the storage space increase lightly.

A. SOFTWARE QUALITY ATTRIBUTES

The application software gives justice to important quality attributes such as:

- Flexibility: Input related to various

domains accepted by the system.

- Reliability: System generates crime report data which includes the expected output as well as the criminal profiling.
- Usability: Provides simple user interface

easily accessible by the concerned user.

- Scalability: System can be used for variable data as well as is scalable on multiple systems over same network.
- Security: Secure as the system asks for user's credentials to provide access to system.

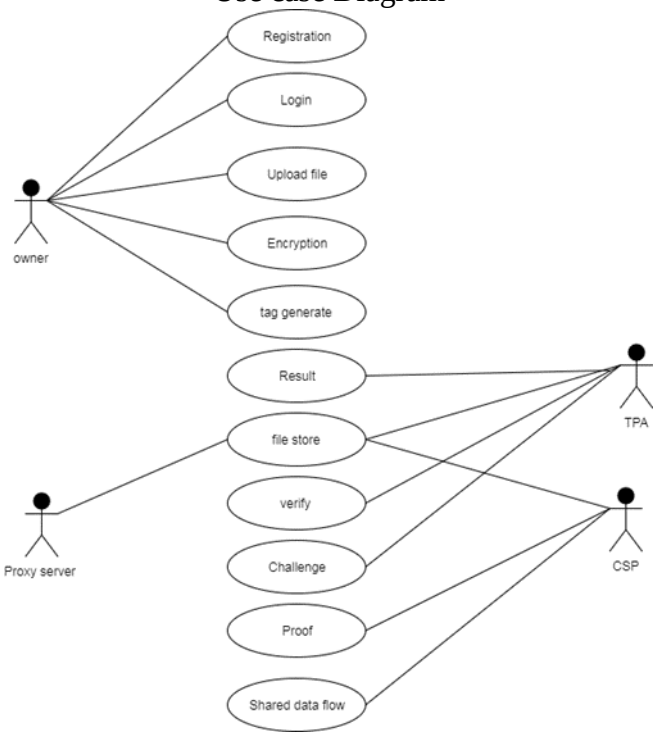
IV .IMPLEMENTATION OF PROPOSED SYSTEM

A. System Model

Static (or structural) view: Emphasizes the static structure of the system using objects, attributes, operations and relationships. The structural view includes class diagrams and composite structure diagrams.

Dynamic (or behavioral) view: Emphasizes the dynamic behavior of the system by showing collaborations among objects and changes to the internal states of objects. This view includes sequence diagrams, activity diagrams and

Use case Diagram



B. System Functionalities

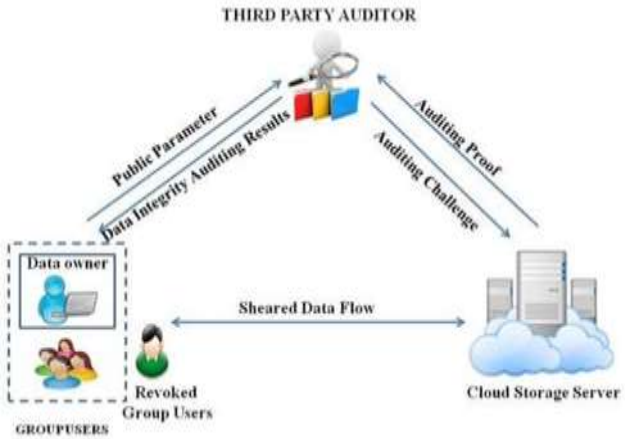
The primary design constraint is the Desktop platform. Since the application is designated for Desktop Systems, effective GUI and well user friendliness will be the major design considerations. Creating a user interface which is both effective and easily navigable is important. Also as we are utilizing the database for our each of the four major steps based on different algorithms so storage space need to be considered for smooth functioning of system.Other constraints such as memory and processing power are also worth considering.The analysis and auditing system is meant to be quick and responsive even when dealing with large amount of data so each feature of the software must be designed and implemented considering efficiency.As our system involves algorithms the system must consider the requirements of all techniques for the format of input and output generated and their individual working efficiency and its contribution to overall software applications efficiency.The software will give the desired results only if the specified software requirements are satisfied.

At present only Text File format containing data in the form of database records of unstructured data or the algorithm's output format is considered.

The system will accept data as input from the database containing unstructured data as records generated by crawling through the web content by a web based crawler and we need active and smooth internet connectivity for effective working of web crawler. Application software designed must implement the algorithms effectively on the collected data and predict the expected result successfully also the interface of software must be easy and simple to be understood by crime analyst and no extra efforts needed by them to understand the usage of software.

- Time Dependency:

Usability improvements and convenience enhancements that may be added after the application has been developed. Thus, the implementation of these features is entirely dependent upon the time spent designing and implementing the core features. The final decision on whether or not to implement these features will be made during the later stages of the design phase.



III. RESULTS DISCUSSION

We determine that our proposed SecCloud framework has accomplished both integrity auditing and file deduplication. Be that as it may, it can't keep the cloud servers from knowing the substance of files having been put away. In other words, the functionalities of integrity auditing and secure deduplication are just forced on plain files. In this area, we propose SecCloud+, which takes into account integrity auditing and deduplication on scrambled files. Framework Model Compared with SecCloud, our proposed SecCloud+ involves an extra trusted element, to be specific key server, which is in charge of assigning customers with mystery key (according to the file content) for encrypting files. This construction modeling is in line with the late work. However, our work is distinguished with the past work by allowing for integrity auditing on encoded data. SecCloud+ takes after the same three protocols (i.e., the file uploading protocol, the integrity auditing protocol and the proof of proprietorship protocol) as with SecCloud. The main distinction is the file uploading protocol in SecCloud+ involves an extra stage for correspondence between cloud customer and key server. That is, the customer needs to speak with the key server to get the merged key for encrypting the uploading file before the phase in SeeCloud.

CONCLUSION

In this system, we investigated a new primitive called identity-based remote data integrity checking for secure cloud storage. We formalized the security model of two important properties of this primitive, namely, soundness and perfect data privacy. We provided a new construction of of this primitive and showed that it achieves soundness and perfect data privacy. Both the numerical analysis and the implementation demonstrated that the proposed protocol is efficient and practical.

ACKNOWLEDGEMENT

I thank prof.C.P Lachake and Prof. Ravishankar Bhaganagare (Assistant Professor), Department of Computer Science and Engineering, SKN Sinhgad Institute of Technology & Science, Lonavala, for their valuable guidance, help, and encouragement.

I would like to thank SKN Sinhgad Institute of Technology & Science, Lonavala College for giving me the opportunity to enhance my knowledge and skills in machine learning. I am also thankful to my parents and family members for their unwavering moral and economic support.

This acknowledgment would be incomplete if I did not express my sincere gratitude to all those who have contributed directly or indirectly to this work. Any inadvertent omission is purely unintentional and does not reflect a lack of gratitude on my part.

REFERENCES

Est [1] P. Mell, T. Grance, Draft NIST working defini tion of cloud computing,Reference on June. 3rd, 2009. http://csrc.nist.gov/groups/SNC/cloudcomputing/index.html.

[2] Cloud Security Alliance. Top threats to cloud computing. http://www.cloudsecurityalliance.org, 2010.

[3] M. Blum, W. Evans, P.Gemmell, S. Kannan, M. Naor, Checking the correctness of memories. Proc. of the 32nd Anual Symposium on Foundations fo Vomputers, SFCS 1991, pp. 90– 99, 1991.

[4] S.L. Ting, W.H. Ip, Albert H.C. Tsang, "Is Naive Bayes a Good Classifier for Document Classification?", International Journal of Software Engineering and Its Applications Vol. 5, No. 3, July, 2013.

[5] G. Ateniese, R. C. Burns, R. Curtmola, J. Herring, L. Kissner,

Z. N.J. Peterson, D. X. Song, Provable data possession at untrusted stores. ACM Conference on Computer and Communications Security, 598-609,2007
[6] G. Ateniese, R. C. Burns, R. Curtmola, J. Herring, O. Khan,
L. Kissner, Z. N. J. Peterson, and D. Song, Remote data checking using provable data possession. ACM Trans. Inf. Syst. Secur., 14, 1–34, 2011.
[7] A.Juels, and B. S. K. Jr. Pors, proofs of retrievability for large files.Proc.of CCS 2007, 584-597, 2007. [8] H. Shacham, and B. Waters, Compact proofs of retrievability. Proc. of Cryptology-ASIACRYPT 2008, LNCS 5350, pp. 90-107, 2008.
[9] G. Ateniese, S. Kamara, J. Katz, Proofs of storage from homomorphicidentifica tion protocols. Proc. of ASIACRYPT 2009, 319- 333, 2009.
[10] A. F. Barsoum, M. A. Hasan, Provable multicopy dynamic data possession in cloud computing ystems, IEEE Trans. on Information Forensics and Security, 10(3): 485–497, 2015
[11] J. Liu, K. Huang, H. Rong, H. M. Wang, Privacy-preserving publicauditing for regenerating-code-based cloud storage, IEEE Trans. on Information Forensics and Security, 10(7): 1513–1528, 2015.
[12] H. Shacham and B. Waters, "Compact proofs of retrievability," in Proceedings of the 14th International Conference on the Theory and Application of Cryptol ogy and Information Security: Advances in Cryp- tology, ser. ASIACRYPT '08. Springer Berlin Heidelberg, 2008, pp. 90–107.