# ROBUST KEY REVELATION OF PUBLIC AUDITING PROTOTYPE FOR SECURE CLOUD STORAGE

**M.Sahaya Sheela** [1] Associate professor/Deportment of ECE/ Pavai college of technology
**M.Muthuraja,**[2] Assistant professor/Deportment of CSE/ KONGU College of Engineering
**T.Shanthi,**[,3] Assistant professor/Deportment of ECE/ Paavai Engineering College

## ABSTRACT

In cloud storage the data clients can remotely store their data and use on-demand high-quality applications. Data outsourcing users are reassured from the trouble of data storage and maintenance when users put their data in huge size on the cloud, the data integrity protection is a challenging one which enabling public audit for cloud data storage security. Users can ask an external inspection of third party to check the integrity of the out sourced data. However, in such a scheme, the malicious cloud might still forge valid authenticators later than the key-revelation time period if it obtains the current secret key of Data Client. In this paper, propose a prototype named Robust Key Revelation of public auditing for secure cloud storage, in which the security of cloud storage auditing not only earlier than but also later than the key revelation can be preserved. Formalized the definition and the security model of this new kind of cloud storage auditing and design a concrete scheme. In this proposed prototype, the key revelation in one-time period doesn't affect the security of cloud storage auditing in other time periods. The accurate security proof and the experimental results demonstrate that this proposed prototype achieves desirable security and efficiency with the time intervals.

**Keywords:** Secure Cloud Storage, Robust Key Revelation, Security, Data Client, and Public Auditing.

## CHAPTER 1

### 1.1 INTRODUCTION

Cloud storage auditing is an essential part of cloud computing, whose goal is to provide powerful and on-demand out-sourcing data services for users exploiting highly virtualized infrastructures [1], [2]. Due to the low-cost and high-performance of cloud storage, a growing number of organizations and individuals are tending to outsource their data storage to professional cloud services providers (CSP), which buoys the rapid development of cloud storage and its relative techniques in recent years. However, as a new cutting-edge technology, cloud storage still faces many security challenges [3]. One of the biggest concerns is how to determine whether a cloud storage system and its provider meet the legal expectations of customers for data security [4].

The cloud environment is ensured to be secured by a set of protocols which have various mechanisms to login, access and maintains a log of records of users' information. Having the access controlling mechanism is challenging issues in vast cloud environment [5]. When a data client occupies a space in the cloud, the space is actually a virtual space allotted. That virtual space will be having limitations for other users to access data which belongs to others. Conventional access methods will have techniques to protect the device holding the memory. Whereas in a cloud, the data is much farther than the data clients control. A cloud infrastructure is a completely different domain where no single user has a control on. Not only security, a cloud has also number of other features to concern about namely, concurrency, ease of use, integrity and confidentiality.

The auditing protocols for cloud storage have pulled in much consideration and have been examined seriously [6]. These protocols center on a few unique parts of examining, and how to accomplish high data transfer capacity and algorithm effectiveness is one of the fundamental concerns [7]. For that reason, the Homomorphism Linear Authenticator (HLA) procedure that backings squareless check is investigated to lessen the overhead so algorithm and correspondence in auditing protocols, which enables the reviewer to confirm the trustworthiness of the information in cloud without recovering the entire information.

Frequent cloud storage auditing protocols like have been proposed in view of this procedure [8]. The security assurance of information is additionally an essential part of cloud storage auditing. So as to lessen the computational weight of the customer, a third-party auditor (TPA) is acquainted with help the customer to occasionally check the honesty of the information in cloud. Be that as it may, it is feasible for the TPA to get the user’s information after it executes the auditing protocol numerous circumstances [9]. Auditing protocols are intended to guarantee the security of the user’s information in cloud. Another angle having been tended to in cloud storage auditing is the means by which to help information dynamic activities [10].

**Key presentation could occur because of a few reasons:**

**Key service:** Key service is a procedure which is finished by the customer. In the event that any blame happens and if the customer is utilizing a shabby programming based key service, at that point key introduction is conceivable.

**Internet based security attacks:** Suppose if a customer downloads any information or document and if that it contains malevolent program, at that point it might taint the framework. This enables the programmers to effortlessly get to any secret information.

**Trading with programmers:** The cloud acquires motivating forces by exchanging ideas with the concerned programmers in the trading method. In this procedure, the cloud can get the customer's information and manufacture the authenticator by recovering false information or by concealing information misfortune. In this approach, managing key presentation is a fundamental issue in cloud storage and different techniques were embraced.

In this paper contribute that the security of cloud storage auditing scheme in any time period other than the robust key revelation time period. In this detailed construction, the Third Party Auditor (TPA) generates an update message from his secret key in each time period, and then sends it to the client. The client updates his signing secret key based on his private key and the update message from the TPA. This method makes the malicious cloud unable to obtain the signing secret keys in unexposed time periods. The rest of the paper session 2 is discussed about the literature review. Session 3 is discussed about the methodology of the proposed system. Session4 is discussed about the experimental results of the proposed system. Session 5 describes the conclusion of this paper.

## CHAPTER 2

## LITERATUREREVIEW

Wang et al., [11] presented the problems of ensuring data storage correctness and proposed an effective and secure scheme to address these issues. A third party auditor (TPA) is introduced securely, who on behalf of users request will periodically verify integrity of the data stored on cloud server. There will not be any online burden on user and security of data will be maintained as the data will not be shared directly with the third party auditor. A homomorphic encryption scheme is used to encrypt the data by using Elliptic curve Cryptography (ECC) which will be

Shared with the TPA.ECCprovidesefficientandsecuresolutionsforthecloudstorageservers.It leads to fast computation time, reducing in processing power, save the storage and bandwidth.The results can be further extended to enable the third party audit or to do multiple auditing tasks.

Deshmukh et al., [12] proposed to ensure the data security a framework is introduced, which usages distributed scheme. Proposed framework includes a master server and an arrangement of slave servers. There is no direct connection in the middle of clients and slave servers. Master server is responsible for preparing the clients appeals and at slave server chunking operation is completed. Chunking operation is responsible for storing duplicates of records to give backup of data for recovery of document in future. Clients can likewise perform powerful and dynamic data operations. Client's document is kept at main server as tokens and records were chunked on slave servers for file recovery. Subsequently proposed scheme accomplished storage integrity and accessibility of data for that token generation and merging algorithms were used.

Wangetal.,[13]Proposed system permits clients inspecting the cloud data storage. This mechanism uses homomorphic token with reed-Solomon era sure correcting code technique, which promises the correctness assurance and also identifies which server is misbehaving. This design also extended to support block level dynamic operations. If users do not have sufficient resources and time available for processing data then user can delegate this task to TPA. Thus this technique allows user to store data at remote place securely and supports dynamic operations such as insert, update & delete. Zheng et al., [14] proposed a secure sustainable storage auditing protocol that can support key updates for clients in cloud computing. In order to alleviate the high overhead of key updates at the local side, the partial key update tasks are outsourced to the TPA. Moreover, clients can verify the validity of the new updated keys by using the technology of the BLS signature. The security analysis shows that the proposed protocol can provide the security properties of the correctness, the verifiability and the accountability.

Jiangetal.,[15]introduced based on ID-based signature technology, by strengthening information authentication and the computing power of the auditor, it proposed an ID-based public auditing protocol for cloud data integrity checking.It also proved that the proposed protocol is secure in the random oracle model under the assumption that the Diffie-Hellman problem is hard. Furthermore, it compare the proposed protocol with other two ID-based auditing protocols in security features, communication efficiency and computation cost. The comparisons showed that the proposed protocol satisfies more security features with lower computation cost.

Yang et al., [16] proposed a public auditing scheme for data confidentiality, in which user re sorts to a Third-party auditor (TPA) for auditing. This scheme design a special log called attestation in which hash user pseudonym is used to preserve user privacy. Attestation-based data access identifying is presented in this scheme which brings none vulnerabilities toward data confidentiality and no extra online burden for user.It further supports accountability of responsible user for data leakage based on user pseudonym. Extensive security and performance analysis compare our scheme with existing auditing schemes.

Thokchom et al.,[17]proposedanefficientauditingschemeforcheckingtheintegrityofdynamic data shared among a static group of users outsourced at un trusted cloud storage. The scheme is designed based on CDH-based ring signature scheme. The scheme enables a third party auditor to audit the client's data without knowing the content while also preserving the identity privacy of the group member who is signing the data from the auditor as well as from the cloud server. The identity of the group member who is signing the data block can be revealed only by the authorized opener, if needed.

Wu et al., [18] introduced it concentrate on the identity privacy of CLCA schemes. it define the security models of privacy-preserving CLCA schemes, namely the uncheatability and anonymity and proposed an efficient CLCA scheme, which is secure in the security models. As a feature of this scheme, the tag of a message is compact, which consists of only one group element. The uncheatability is based on variants of bi linear Diffie–Hellman assumption in the random oracle model. The identity privacy of the user is information-theoretically guaranteed against the third party auditor.

Han et al., [19] proposed a lightweight and privacy-preserving public cloud auditing scheme for smart cities that does not require bilinear pairings. First, the proposed scheme is pairing-free, and allowing a third party auditor to generate authentication meta set on behalf of users. Furthermore,it also protects data privacy against the third party auditor and the cloud service providers. In addition, this new scheme can be easily and naturally extended to batch auditing in a multi-user scenario. Detailed security and performance analyses show that the proposed scheme is more secure and efficient compared to the existing public cloud auditing schemes. Zhao et al., [20] propose a privacy-preserving and unforgetable searchable encrypted audit log scheme based on PEKS. Only the trusted data owner can generate encrypted audit logs containing access permissions for users. The semi-honest server verifies the audit logs in searchable encryption way before granting the operation rights to users and storing the audit logs. The data owner can perform a fine-grained conjunctive query on the stored audit logs, and accept only the valid audit logs. The scheme is immune to the collusion tamper or fabrication conducted by server and user. Concrete implementations of the scheme is put forward in detail. The correct of the scheme is proved, and the security properties, such as privacy-preserving, searchability, verifiability and unforgeability are analyzed.

## CHAPTER 3

## ROBUST KEY REVELATION PUBLIC AUDITING PROTOTYPE

In this paper contribute that the security of cloud storage auditing scheme in any time period other than the robust key revelation time period. In this detailed construction, the Third Party Auditor (TPA) generates an update message from his secret key in each time period, and then sends it to the client. The client updates his signing secret key based on his private key and the update message from the TPA. This method makes the malicious cloud unable to obtain the signing secret keys in unexposed time periods.

The proposed system comprises three parties: the cloud, the client and the third party auditor (TPA).The cloud offers storage services to the client. The client uploads client documents along with the corresponding authenticators to the cloud, and then deletes these data from client storage space. The client can retrieve them from the cloud when client needs them. The TPA is a controlling third party and is in charge of two important tasks. The first is to provide auditing service, i.e., periodically check the integrity of the documents stored in cloud for the client. Second is to help the client to update their secret keys by providing update messages to the clients in different periods.

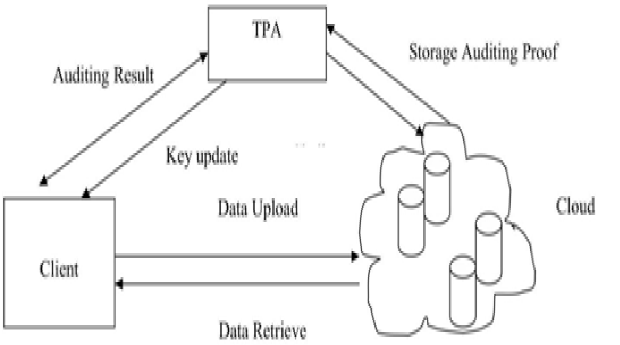


Fig.1.ProposedSystemArchitecture

As same as the majority of public integrity auditing schemes, the TPA is honest for integrity auditing on behalf of cloud users. However, it is not fully reliable for key update in this system model.The system model is shown in Fig.1.Assume that one document D stored in cloud is divided into y blocks xi(i = 1,…., y). Different from the system model in [21], the lifetime of document D in this system model does not need to be fixed initially. It means the total number of time periods in unbounded, which is more close to the reality.

### SECURITYALGORITHMTYPES

An auditing protocol with key presentation flexibility is created by five algorithms (SysSetup, KeyUpdate, AuthGen, ProofGen, ProofVerify), established as follows:

SysSetup(1k, T ) →(OPK, EK0): The framework setup algorithm is a probabilistic algorithm which takes as info a security parameter k and the aggregate number of eras T, and produces an open key OPK and the underlying users secret key EK0. This algorithm is controlled by the client.

KeyUpdate(OPK, j, EK j ) → (EK j+1): The key restore algorithm is a probabilistic algorithm which takes as info people in general key OPK, the present time frame j and a user's secret key EK j , and produces another secret key EK j+1 for the following time frame j + 1. This algorithm is controlled by the client.

AuthGen(OPK, j, EK j , D) → (_): The authenticator age algorithm is a probabilistic algorithm which takes sin for people in general key OPK, the present time frame j, a user's secret key EK j and a document D, and creates the arrangement of authenticators _ for D in day and age j. This algorithm is likewise keep running by the client.

Proof Gen(OPK, j,Chal, R,_) → (P): The proof generation algorithm is a probabilistic algorithm which takes as information the general population key OPK, an era j , a test Chal denotes the challenge phase, a document D and the arrangement of authenticators_, and creates a proof P which implies the cloud has D. Here, ( j,Chal) combine is issued by the reviewer, and after that utilized by the cloud. This algorithm is controlled by the cloud.

Proof Verify(OPK, j,Chal, P) → ("Right" or "Wrong"): The evidence confirming algorithm is a deterministic algorithm which takes as information people in general key OPK, an era j , a test Chal and a proof P, and returns "Right" or "Wrong". This algorithm is controlled by the client.

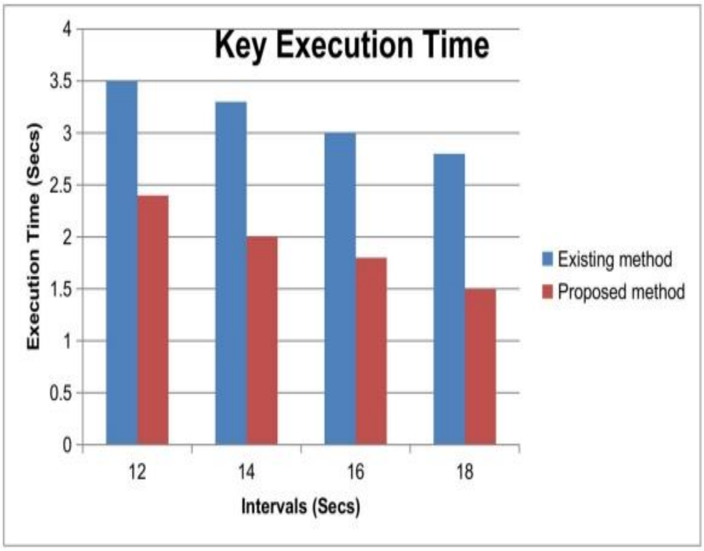
The above security model captures that a challenger cannot provide a valid proof for a time period in which the secret key is not exposed without owning all the blocks corresponding to a given challenge, if it cannot guess all the missing blocks. All the blocks of authenticators can query for each time period. The adversary can also query the secret keys in all except the challenged time period. The goal of the adversary is to construct a valid proof of possession P for

The blocks indicated by Chal in the time period t*. The following definition shows that there exists a knowledge extractor that can extract the challenged data blocks whenever the adversary can produce valid proof of possession P in the time period t*.

## ALGORITHM

Pis the Prototype

P ={X, O, F , K,T, Success, Failure }Where,



X = Set of InputX={X1,X2,X3}

Where,

X1=Login user IDX2=Login passwordX3=File

K=Key set of Secret key and Open keyK={(SK1,OP1),(SK2,OP3)…..,(SKi,OPi

)}

O=Setof Outputs

O={O1, O2, O3,O4}

Where,

O1=Authentication MessageO2=Encrypted File

O3=AttackDetectionO4=PeriodickeyO5=Original Data file

T=Time Period for key generation F=Set of Functions

F={F1,F2,F3, F4,F5}

Where,F1=AuthenticationO1←F1(X1,X2)

F2=EncryptionO2←F2(X3,K)

F3=Attack DetectionO3←F3(K)

F4=PeriodickeyGenerationO4←F4(O3,T)

F5=DecryptionO5←F5(O2,K)

## CHAPTER 4

## RESULT AND DISCUSSION

In this section results has to be discussed with the data clients connected to the cloud service providers. Time taken for monitoring the data clients to retrieve the documents in the time periods using the encryption key, this has to be compared with the existing method with proposed prototype. The communication overhead is considerably higher than expected and the current number of users made the overhead in a controlled level. The results displayed portray the effect of having such a monitoring prototype over the architecture and how the system performs. The existing method compare with the proposed method that discussed below.

Fig.2. Key Execution Time of existing and proposed methods

Open Stack Time is the overall time taken to process the request, obtain the changes from utilizer, authorize the changes and update them onto the cloud space modules. Monitoring time is also displayed to depict the importance of implementing a security model, with such efficiency and it is not going to affect any performance of the cloud provider. Ping time is the total time; a data originator is connected to the monitoring prototype during the process. Fig. 2 shows that the results of the robust key execution time compare with existing and proposed methods.

## CHAPTER 5

## CONCLUSION

In this paper deal with the key revelation problem in the cloud storage auditing, thus it concludes the proposed a prototype named Robust Key Revelation of public auditing for secure cloud storage, in which the security of cloud storage auditing not only earlier than but also later than the key revelation can be preserved. Formalized the definition and the security model of this new kind of cloud storage auditing and design a concrete scheme. In this proposed prototype, the key revelation in one time period doesn’t affect the security of cloud storage auditing in other time periods. The accurate security proof and the experimental results demonstrate that this proposed prototype achieves desirable security and efficiency with the time intervals.

## CHAPTER 6
## REFERENCES:

1. Dewan,H. and Hansdah,R.C.,2011,July.A survey of cloud storage facilities. In 2011 IEEE World Congresson Services( pp.224-231).IEEE.

2. Wang, C., Wang, Q., Ren, K., Cao, N. and Lou, W., 2012. Toward secure and dependable storage services in cloud computing. IEEE transactions on Services Computing, 5(2),pp.220-232.

3. Ren, K., Wang, C. and Wang, Q., 2012. Security challenges for the public cloud. IEEE Internet Computing, 16(1), pp.69-73.

4. Ryoo, J., Rizvi, S., Aiken, W. and Kissell, J., 2014. Cloud security auditing: challenges and emerging approaches. IEEE Security & Privacy, 12(6),pp.68-74.

5. Massonet, P., Naqvi, S., Ponsard, C., Latanicki, J., Rochwerger, B. and Villari, M., 2011,May. A monitoring and audit logging architecture for data location compliance in federated cloud infrastructures. In 2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum (pp.1510-1517).IEEE.

6. Wang, C., Ren, K., Lou, W. and Li, J., 2010. Toward publicly auditable secure cloud data storage services.IEEEnetwork,24(4),pp.19-24.

7. Yang, K. and Jia, X., 2013. An efficient and secure dynamic auditing protocol for data storage in cloud computing. IEEE transactions on parallel and distributed systems, 24(9),pp.1717-1726.

8. Wang, C., Chow, S.S., Wang, Q., Ren, K. and Lou, W., 2013. Privacy-preserving public auditing for secure cloud storage. IEEE transactions oncomputers,62(2), pp.362-375.

9. Varalakshmi, N. and Krishna, K.J., 2018. Secure Cloud Storage Auditing Protocol with Resisting Key-Exposure.

10. Wang, C., Chow, S.S., Wang, Q., Ren, K. and Lou, W., 2013. Privacy-preserving public auditing for secure cloud storage. IEEE transactions on computers,62(2), pp.362-375.

11. Deshmukh, P.M., Gughane, A.S., Hasija, P.L. and Katpale, S.P., 2012. Maintaining file storage security in cloud computing. International Journal of Emerging Technology and Advanced Engineering, 2(10),pp.2250-2459.

12. Wang, C., Wang, Q., Ren, K., Cao, N. and Lou, W., 2012. Toward secure and dependable storage services in cloud computing. IEEE transactions on Services Computing, 5(2),pp.220-232.

13. Zheng, W., Liu, D., Li, X. and Sangaiah, A.K., 2018. Secure sustainable storage auditing protocol(SSSAP)with efficient key updates for cloud computing. Sustainable Computing: Informatics and

Systems.

14. Jiang, H., Xie, M., Kang, B., Li, C. and Si, L., 2018. ID-Based Public Auditing Protocol for Cloud Storage Data Integrity Checking with Strengthened Authentication and Security. Wuhan University Journal of Natural Sciences, 23(4),pp.362-368.

15. Yang, Z., Wang, W., Huang, Y. and Li, X., 2019. Privacy-Preserving Public Auditing Scheme for Data Confidentiality and Accountability in Cloud Storage. Chinese Journal of Electronics, 28(1),pp.179-187.

16. Thokchom, S. and Saikia, D.K., 2018, October. Efficient scheme for dynamic Cloud data shared within a static group with privacy preserving auditing and traceability. In Proceedings of the 2018 International Conference on Cloud Computing and Internet of Things (pp. 25-32).ACM.

17. Wu,G.,Mu,Y.,Susilo,W.,Guo,F.andZhang,F.,2019.Privacy-Preserving Certificate less Cloud Auditing with Multiple Users. Wireless Personal Communications,pp.1-22.

18. Han, J., Li, Y. and Chen, W., 2019. A Lightweight And privacy-preserving public cloud auditing scheme without bilinear pairings in smart cities. Computer Standards & Interfaces,62, pp.84-97.

19. Zhao, W., Qiang, L., Zou, H., Zhang, A. and Li, J., 2018, June. Privacy-Preserving and Unforgetable Searchable Encrypted Audit Logs for Cloud Storage. In 2018 5th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2018 4thIEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom) (pp.29-34).IEEE.

20. Yu, J., Ren, K., Wang, C. and Varadharajan, V., 2015. Enabling cloud storage auditing with key-exposure resistance. IEEE Transactions on Information forensics and security,10(6),pp.1167-1179