A FRAMEWORK FOR AUDITING KEY EXPOSURE AND INTEGRITY VALIDATION FOR SECURE CLOUD STORAGE

[1]G. Likitha Reddy, [2]G. Swetha, [3]N. Sahana, [4]J. Vijay Shekar Reddy, [5]V. Sairaj
[1]Assistant Professor, [2,3,4,5]B.Tech
[1,2,3,4,5]Department of Computer Science and Engineering,
[1,2,3,4,5]St. Martin's Engineering College, Hyderabad, India

Abstract : Cloud Computing prolongs to transmute the manner in which the company utilizes, stores and shares data and applications along with new security threats and challenges. As large volume of data is stored on the cloud these resources become natural targets for attackers. Since data integrity is the fundamental component of information security and data loss might happen in the cloud, it is natural for data owners to question about the correctness of their data in the cloud. Most of the existing protocols are based on the surmise that the client's secret key is secure. But this supposition may not always be held due to weak security settings by the client. In order to overcome the above problems, we proposed an efficient model which eliminates the threat caused due to key-exposure and integrity of data. We formulated a new architecture in which we included a trusted TPA (third party auditor) for authentication and key-update message generation. Also, the accuracy and consistency of data stored on the cloud can be verified by an analyzer. This proposed scheme proves to be more reliable and achieves greater security.

***IndexTerms* - Key exposure resistance, Cloud storage auditing, Data integrity.**

I. INTRODUCTION

Cloud computing is the delivery of computing services- servers, storage, databases and more- over the Internet ("the cloud"). It offers various benefits- eradicates the capital expense of buying hardware and software, provides on-demand service, ability to scale elastically. It provides three ways of deploying cloud computing resources- public, private and hybrid cloud. Public clouds are possessed and managed by a third party cloud service provider, which furnish their computing resources like storage and servers over the Internet. Private clouds are used overly by a sole business or a corporation. It can be physically positioned in the premises of the company. Hybrid clouds integrate public and private clouds, which allow data to be shared between them.

Apart from these services and benefits, cloud computing presents privacy and security concerns which are classified into two- one concern that is faced by the cloud providers and the other issue faced by the companies who store data on the cloud. Protecting client's data and ensuring the security of the infrastructure must be done by the provider. Moreover, the client must take measures to protect their data and use vigorous passwords and authentication steps. There might be a possibility that the information could be fortuitously or intentionally modified or removed or shared with any third parties by the service provider. As stated by the Cloud Security Alliance, the major threats in the cloud are Insecure Interfaces and API's, Data Loss and Leakage and Hardware Failure- which accounted for 29, 25 and 10 percent of all cloud security outages respectively. Jointly, these form shared technology vulnerabilities. Also, the report states that the insider attacks are the sixth biggest threat to cloud computing.

Over the years, protocols for inspecting cloud storage have enticed much attention and have been investigated in-depth. These protocols focused on divergent attributes of auditing in order to achieve high bandwidth and computational efficiency. To reduce the aerial of computation and communication in auditing protocols, an approach called Homomorphic Linear Authenticator (HLA) is considered. The existing approaches overlooked the low- security settings at the user and concentrated on the flaws in the cloud. To deal with the client's secret key disclosure, it requires downloading the entire information from the cloud, creating new authenticators and re-uploading entirety back to the cloud. This entire scenario can be monotonous and unmanageable. It can lead to high computation cost for the storage auditing.

II. LITERATURE SURVEY

With a certain end objective to test the reliability of data present in the remote server, various protocols were suggested which focused on various preconditions such as stateless confirmation, high proficiency, data dynamic operation, security insurance, etc. To ensure that the server holds the client's actual data located at distrustful servers, a method called Provable Data Possession (PDP) was first proposed by Ateniese et al. which used HLA strategy and arbitrary example to verify outsourced data. Proof of Retrievability (PoR) prototype was scrutinized by Juels and Kaliski Jr. They employed tactics of error-correcting codes and spot-checking to safeguard both proprietorship and retrievability of information on remote servers. Shacham and Waters presented two homomorphic authenticators: one with private verifiability which relies on pseudorandom operations; the other with public verifiability which relies on BLS signature. Dodiset al engrossed on the differential prevailing PoR work. Wang et al united the HLA with erratic masking method to enable the auditor impotent to retrieve the actual information during auditing task. Erway et al used skip list-based pattern and propounded PDP. Zhu et al. proffered an appealing PDP to support an essential auditing. Yang and Jia contemplated privacy-preserving and dynamic operation in cloud storage auditing programme. Cash et al. posited a strong PoR with the help of ram method. An auditing plan for low-power users was given by Guan et al which was identical and obscure. A preponderance of the above protocols depends on the hypothesis that the client's secret key is unassailable. As we have signified it prior, this speculation may not be verifiable. Consequently, the ongoing mechanism inspects on how to carry out the resilience of key-exposure in cloud storage under the new problem environment.

The existing system fail to suppress two major security issues related to the cloud - data integrity and key-exposure. Coming to the data integrity, when a user uploads a file to the cloud, the service provider must assure the accuracy of the file. The resources of the cloud must be readily available, easy to access and must be in a consistent manner. Integrity of the file is assured only when the service provider guarantees that the file is not tampered and stored correctly. Nevertheless, the existing model does not guarantee the integrity of data. On the other hand, the service provider must ensure that the owner's secret key for downloading the file is secure. It must not be shared with any third party nor be given to any person who has no access rights. But all the existing protocols are based on the assumption that the client's key is absolutely safe. This may not be true as there is a possibility of exposure of secret key due to weak security settings.

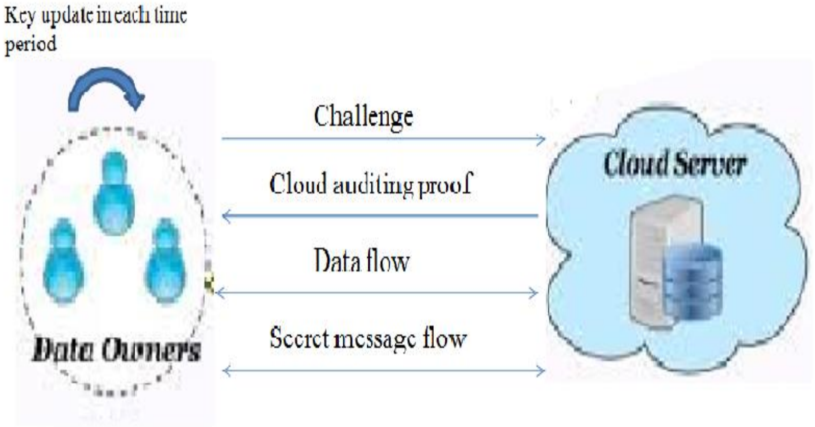


Fig. 1 Architecture of Existing System In Cloud Storage Auditing

In the existing system architecture it is clearly depicted on how the interaction takes place between the data owners and cloud server. Though the cloud server sends an auditing proof on receiving a challenge from the cloud, it cannot be guaranteed that the generated proof is valid or not. There might be a possibility of sending wrong information to the owner and misleading them. Also, the secret message flow between the two may not be secure.

2.1 Major Drawbacks of Existing Method

- Clients are unaware of the accuracy of their information present in the cloud.
- Storage security is less.
- The secret key of the data owner is prone to attack due to weak security settings. Also, the attacker might stop intruding at once if the secret key is obtained. This makes it difficult for the client to detect the key exposure. It can be found out only by recognizing whether the valid authenticators are generated by the client or not. In such a scenario, the client needs to generate a new key with the help of his private and public keys.

In order to overcome the above security threats, we proposed a novel and efficient mechanism. For mitigating the client's burden of frequently verifying the accuracy of file and preventing the key exposure, we introduced a third party auditor (TPA) and an analyzer. The monotonous job of verifying the file details is done by the analyzer and auditing of client's secret key is done by the TPA.

2.2 Advantages of Proposed System

- Clients can authenticate the integrity of data present in the cloud with the help of analyzer.
- TPA triggers key update message to the client in each time interval in order to help the client in updating the secret key.
- It attains stronger security for key update without reducing the efficiency.
- Each time a user downloads a file from the cloud, an authentication mail is sent to the owner with the details of the user who has downloaded the file.

III. OVERVIEW

We examined the manner in which the key-exposure and the integrity of the data issues can be overcome by presenting a concrete and systematic approach called 'Auditing key-exposure and integrity validation'. This proposed paradigm protects the secret key in each time interval except the key-exposure stage. This system comprises of four principal components namely: cloud, TPA (third party auditor), analyzer and the client/user. The responsibility of each module in this system is explained briefly below:

3.1 Module Description

Client- Here the client is considered as the owner in one case and user in other each of them performing different roles. The owner of the file first registers with the application and these details are sent to the TPA for verification. He can login into the application with valid username and password on successful validation. The owner then uploads an encrypted file to the cloud by selecting valid authenticators from the list of users to whom he want to give access rights. The secret key to decrypt the file is sent to the TPA. The owner of the file receives an update message from the TPA in each time interval for updating the secret key. He then generates a new secret key and delivers it back to the TPA. If any user downloads the file uploaded by the owner, a mail is sent to the owner to verify if the user is a valid authenticator who is given access rights by him. On the other hand, the users can view the files in cloud and request cloud for the decryption key to access the file.

TPA- It plays an important role in the proposed system. TPA can log in to the application with valid username and password. It is considered as a trusted party who is responsible for activating the clients and for authentication purpose. TPA can view the files uploaded by the client and generates an update message to the client in each time period. It validates the request received from the cloud and checks if the requested client has permission to download the file. If verification is passed, the TPA sends a challenge back to the cloud for updating the old secret key. Once the user downloads the file using the decryption key, the TPA generates a key update message again and sends it to the client. The TPA is unaware of the contents of the file as it contains only the file details and secret key, not the entire file.

Cloud- Cloud can log in with valid username and password. It sends the client request to TPA for downloading the file. It can view the TPA challenge to update secret key and sends the file to the client. It can also view the details of the clients and the files uploaded. The cloud storage server provides storage services to the registered clients for storing outsourced files.

Analyzer- It plays a major role in verifying the integrity of the client's data. It stores the file details sent by the client. It retrieves the file attributes from the cloud server and periodically validates the existing file parameters with the details obtained from the cloud. An acknowledgement is sent back to the client if both the file parameters are identical. This scenario is a continuous process which helps the client to know the accuracy of their data.

3.2 Architecture

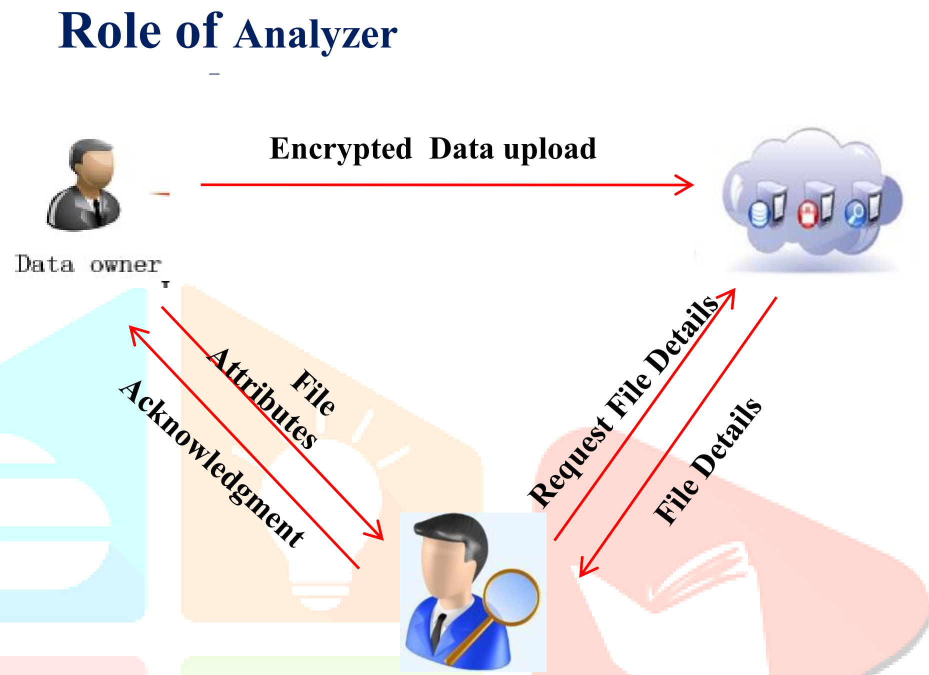


Fig. 2 Architecture of Analyzer

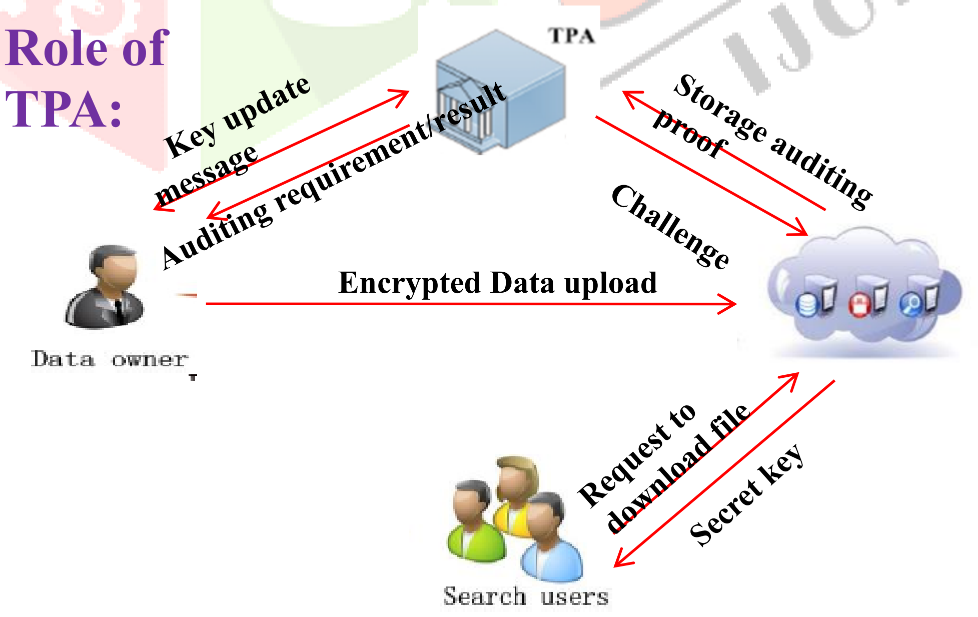


Fig. 3 Architecture of TPA

IV. METHODOLOGY

The proposed framework comprises of the below six algorithms:

1. SysSetup (System setup) - This algorithm is operated by the client.

Input: Security variable k
Output: Client's private key SK_c, System public key PK, TPA's secret key SK_{TPA}.

2. AuthGen (Authenticator generation) - This algorithm is operated by the client.
   Input: File F, Current time period t, Client's signing secret key SK_t, Public key PK.
   Output: Authenticators Φ for File F in time period t.
3. UMGen (Update message generation) - This algorithm is operated by the TPA.
   Input: TPA's secret key SK_{TPA}, Current time period t, Public key PK.
   Output: Update message δ_t .
4. CKeyUpdate (Client key update) - This algorithm is operated by the client.
   Input: Current time period t, Client's private key $SK_{c,}$ Public key PK, Update message δ_t .
   Output: Signing secret key SK_t for time period t.
5. ProofGen (Proof generation) - This algorithm is operated by the cloud.
   Input: AuthenticatorsΦ, Challenge Chal, File F, Public key PK.
   Output: Proof P to ensure that cloud stores the file F correctly.
6. ProofVerify - This algorithm is operated by the TPA.
   Input: Current time period t, Proof P, Challenge Chal, Public key PK.
   Output: Returns true if verification is passed or false accordingly.

In addition to the above six algorithms, we use AES (Advanced Encryption Standard) algorithm for encrypting the owner file before storing it in cloud which is of 128-bit block. It is a repetitive symmetric block cipher which means it uses the same key for encryption and decryption. It is based on substitution permutation network i.e., each round involves byte-level substitution followed by word-level permutation. It does the process of encrypting a 128-bit block in 10 rounds. The execution of the each of the four stages namely SubBytes, ShiftRows, MixColumns and AddRoundKey constitute one round.

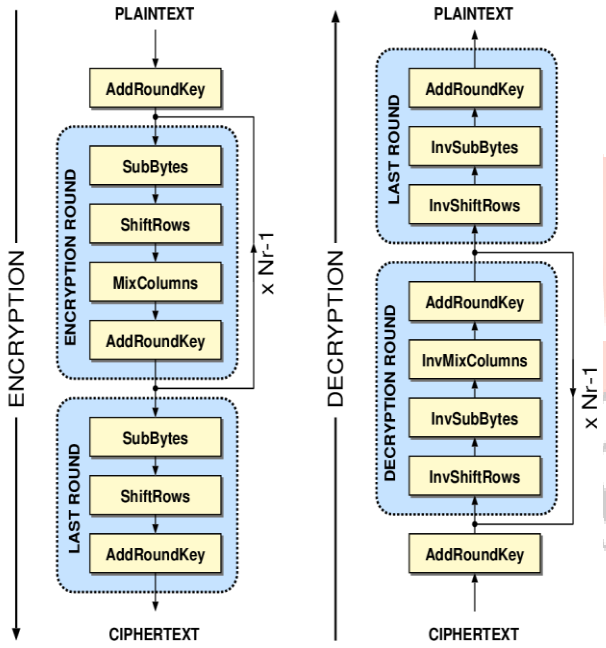
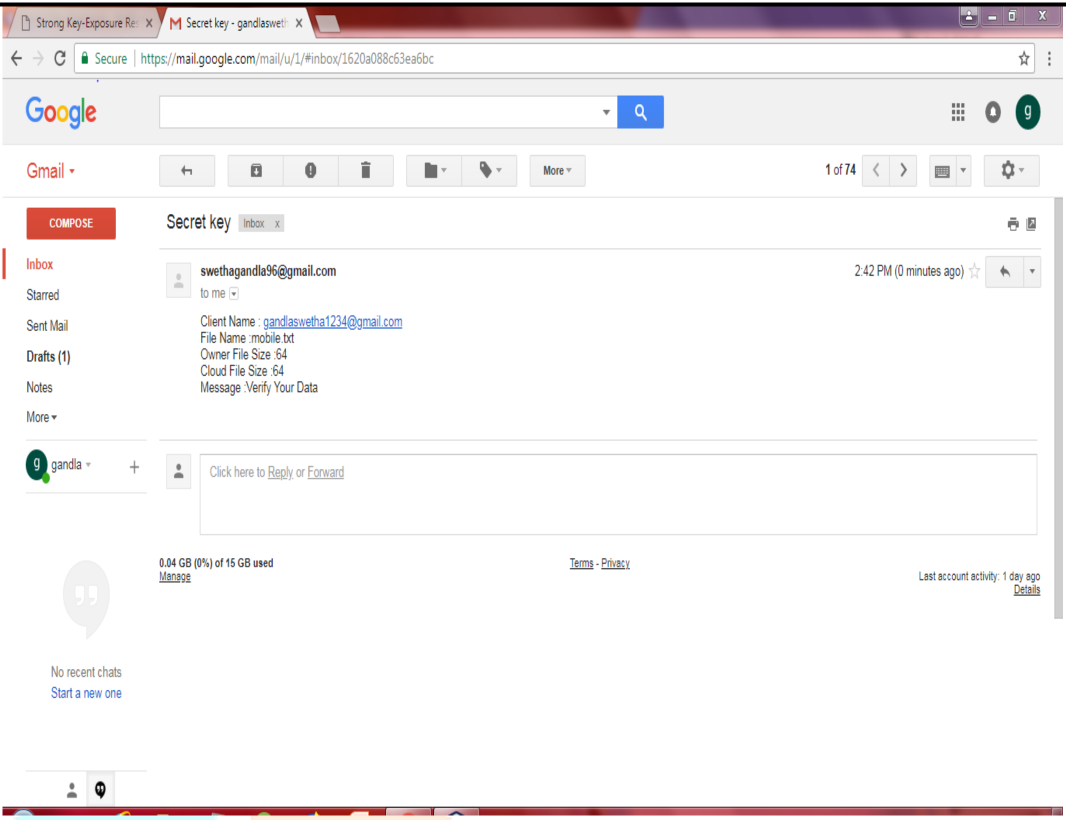


Fig. 4 Process in AES algorithm

V. RESULTS

Once the user registers with the application, the TPA must activate the user in order to log into the application. When an encrypted file is uploaded into the cloud, the secret key is sent to the TPA automatically. The request from the client to download the file is sent to the TPA by the cloud for verification. Only if the user has the access rights they are provided with the decryption key. An acknowledgement is sent to the user when there is change in the file size. On downloading a file from the cloud, the details of the file and downloaded user is sent to the owner of the file.

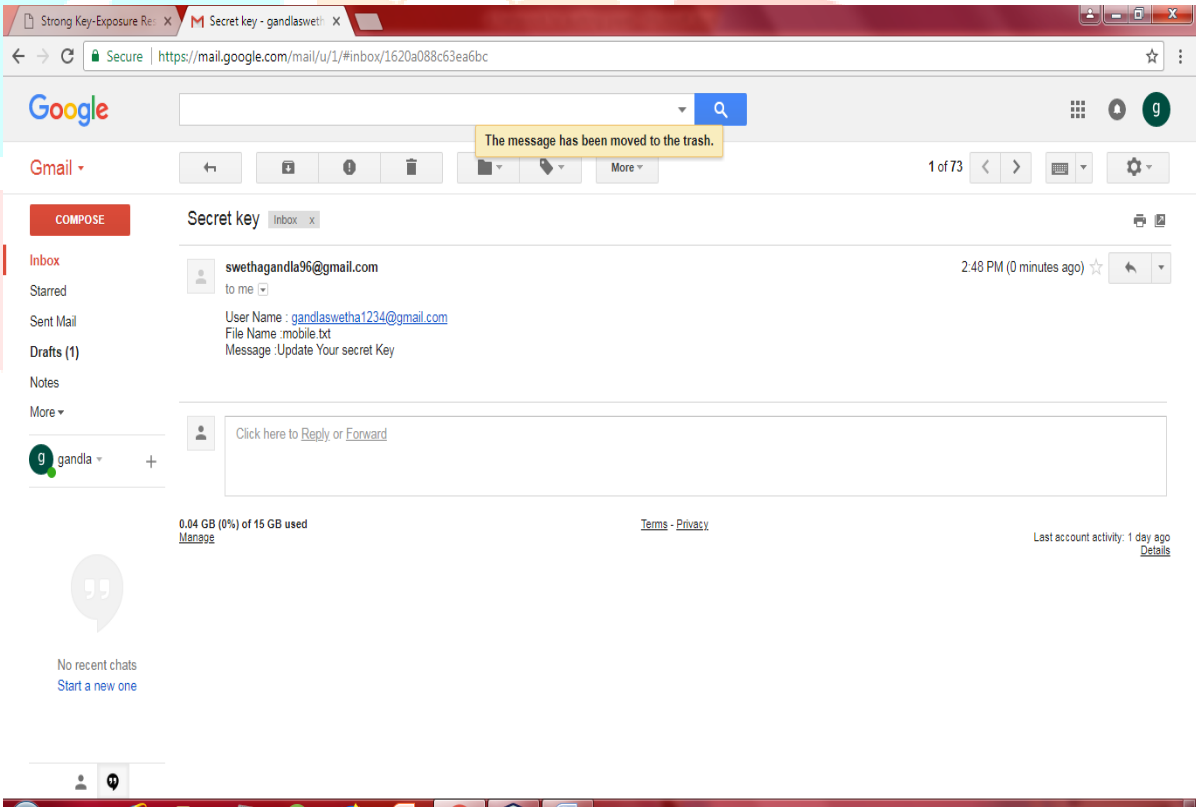


Secret key

swethagandla96@gmail.com
to me
2:42 PM (0 minutes ago)

Client Name : gandlaswetha1234@gmail.com
File Name :mobile.txt
Owner File Size :64
Cloud File Size :64
Message :Verify Your Data

Fig. 5 Acknowledgment Sent By Analyser to the User for Verification



The message has been moved to the trash.

Secret key

swethagandla96@gmail.com
to me
2:48 PM (0 minutes ago)

User Name : gandlaswetha1234@gmail.com
File Name :mobile.txt
Message :Update Your secret Key

Fig. 6 Update Secret Key Sent by the TPA to the User

VI. CONCLUSION

In this paper, we mentioned few of the existing protocols that dealt with the key-exposure mechanism. Analyzing the prevailing ones, a new standard mechanism called key-exposure auditing and integrity validation is presented to deal with the client's key exposure and integrity issues in the cloud. Even if the client's secret key for cloud storage auditing is simple in these kinds of protocols, the integrity of the data previously stored in the cloud can still be validated. The proposed model achieves desirable security and proves to be more efficient. It not only deals with the client key exposure but also verifies the correctness of the data in the cloud. The client's secret key is preserved not only before the key exposure takes place but also later than the key exposure. It mitigates the owner task of verifying the data stored in the cloud by assigning the responsibility to an analyzer.

VII. FUTURE SCOPE

It can be further enhanced by sending an encrypted secret key to the TPA. Also, the task of generating a new secret key by the client in each time period on receiving an update message from the TPA can be eliminated. This job can be assigned to the TPA who will generate a secret key in each time interval and this updated key can be sent to the client's mail.

REFERENCES

[1] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," Proc . 14th ACM Conf. Computer and Comm. Security, pp. 598-609, 2007.

[2] A. Juels and B. Kaliski, "PORs: Proofs of Retrievability for Large Files," Proc . 14th ACM Conf. Computer and Comm. Security, pp. 584-597, 2007.

[3] H. Sha, pp. cham and B. Waters, "Compact Proofs of Retrievability," Advances in Crytpology Asiacrypt*'08*, pp. 90-107, 2008.

[4] Y. Dodis, S.P. Vadhan and D. Wichs, "Proofs of Retrievability via Hardness Amplification," Proc. Theory of Cryptography IEEE Trans. Parallel and Distributed Systems Conf. Theory of Cryptography, pp. 109-127, 2009.

[5] Q. Wang, C. Wang, K. Ren, W. Lou and J. Li, "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing," IEE Trans. Parallel and Distributed Systems, vol. 22, no. 5, pp. 847-859, May 2011.

[6] C. Erway, A. Kupcu, C. Papamanthou and R. Tamassia, "Dynamic provable data possession," Proc. of the 16th ACM conference on Computer and communications security, pp. 213-222, 2009.

[7] Y. Zhu, H. Hu, G. Ahn, and M. yu, "Cooperative Provable Data Possession for Integrity Verification in Multi- Cloud Storage,", vol. 23, no. 12, pp. 2231-2244, Dec. 2012.

[8] K. Yang and X. Jia, "An efficient and secure dynamic auditing protocol for data storage in cloud computing," IEEE Trans. Parallel and Distributed Systems, Vol. 24, No. 9, pp. 1717-1726, 2013.

[9] D. Cash, A. kupcu and D. Wichs, "Dynamic proofs of retrievability via oblivious ram," Advances in Cryptology- Eurocrypt'13, pp. 279-295, 2013.

[10] C. Guan, K. Ren, F. Zhang, K. Florian and J. Yu. "Symmetric-Key Based Proofs of Retrievability Supporting Public Verification," Proc. of the 20th European Symposium on Research in Computer Security (ESORICS'15), pp. 203-223, 2015.