iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 4 **Issue:** X **Month of publication:** October 2016

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

An Advanced protective Auditing and Deduplicating Data in Cloud Servers

K. Munikrishna[1], Munisekhar Prudhvi[2]
[1]*M. Tech, Shree Institute of Technical Education, Tirupati*
[2]*Assistant Professor, Shree Institute of Technical Education, Tirupati*

Abstract: ***As the distributed computing innovation creates amid the most recent decade, outsourcing information to cloud administration for capacity turns into an appealing pattern, which benefits in saving endeavors on substantial information support and administration. By the by, since the outsourced distributed storage is not completely reliable, it raises security worries on the most proficient method to acknowledge information deduplication in cloud while accomplishing trustworthiness evaluating. I concentrate on the issue of uprightness reviewing and secure deduplication on cloud information. In particular, going for accomplishing both information respectability and deduplication in cloud, we propose two secure frameworks, in particular SecCloud and SecCloud+. SecCloud presents an evaluating substance with an upkeep of a MapReduce cloud, which helps customers produce information labels before transferring and in addition review the honesty of information having been put away in cloud. Contrasted and past work, the calculation by client in SecCloud is enormously decreased amid the document transferring and inspecting stages. SecCloud+ is composed inspired by the way that clients dependably need to scramble their information before transferring, and empowers trustworthiness reviewing and secure deduplication on encoded information.***

Keywords***: Seccloud , seccloud+, integrity auditing ,secure de-duplication , proof of ownership convergent encryption.***

I. INTRODUCTION

Even though cloud storage system has been mostly adopted, it fails to accommodate some important eme rging needs such as the capability of auditing integrity of c loud files by cloud clients and detecting duplicated files by c loud servers. We disclose both problems below. The first problem is integrity audit ing. The cloud server is able to relieve clients from the bulky burden of storage management and maintenance. The most difference of cloud storage from traditional in-house storage is that the data is transferred via Internet and stored in an uncertain domain, not under control of the clients at all, which inevitably raises clients great concerns on the integrity of their data. These concerns originate from the fact that the cloud storage is affected to security threa ts from both outside and inside of the cloud, and the uncontrolled cloud servers may passively hide some data loss incidents from the clients to maintain their reputation. What is more serious is that for saving money and space, the cloud servers might even actively and deliberately d iscard barely accessed data files belonging to an ordinary client. Considering the large size of the outsourced data files and the clients' constrained resource capabilities, the first problem is generalized as how can the client efficiently perform regularly integrity verifications even without the local copy of data file.

A. Cloud Clients

Cloud Clients have large data files to be stored and rely on the cloud for data maintenance and computation. They can be either individual consumers or commercial organizations.

B. Cloud Servers

Cloud Servers virtualize the resources according to the requirements of clients and expose them as storage pools.
Typically, the cloud clients may buy or lease storage capacity from cloud servers, and store their individual data in these bought or rented spaces for future utilization.

C. Auditor

Auditor which helps clients upload and audit their out - sourced data maintains a MapReduce cloud and acts like a certificate authority. This assumption presumes that the auditor is associated with a pair of public and private keys. Its public key is made available to the other entities in the system.

II. RELATED WORK

A. *Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing*

Author: Qian Wang

Cloud Computing system has been predicted as the next -generation architecture of IT Enterprise. It moves the application software and databases to the centralized with large data centers, where the management of the data and services may not be fully trus tworthy. This unique ensample brings about many new security challenges, which have not been well understood. Our research work examine the problem of assuring the integrity of data storage in Cloud Computing. In particular, we consider the task of allowing a third party auditor (TPA), on concern of the cloud client, to verify the integrity of the dynamic data stored in the cloud. The introduction of TPA dismiss the involvement of client through the auditing of whether user's data stored in the cloud is truly intact, which can be important in achieving economies of scale for Cloud Computing. The support for data dynamics through the most general forms of data operation, such as block modificat ion, insertion and deletion, is also more powe rful step to – ward practicality, since services in Cloud Computing are not limited to archive or backup data only. While presiding work on ensure remote data integrity often lac k the supports of either public verifiability or dynamic data operation. Proofs of Ownership in Remote Storage Systems.

B. *Proofs of Ownership in Remote Storage Systems*

Author: Shai Halevi

Cloud storage systems are becoming more and more popular. A promising technology that keeps their cost down is deduplication, which stores only a single copy of duplicatin g data. Client-side deduplication attempts to identify deduplication opportunities already at the client side and save the bandwidth of uploading copies of e xisting files to the server. In th is work we identify attacks that e xploit client-side deduplication, granting an attacker to gain access to arbitrary-size files of other users based on a very sma ll hash signature of these files. More specifica lly, an attacker who knows the hash signature of a file can assure the storage service that it owns that file, hence the server lets the attacker download the entire file.

C. *DupLESS: Server-Aided Encryption for Deduplicated Storage*

Author: Mihir Bellare

Cloud storage service providers such as Dropbox, Mozy, and others perform deduplicat ion to save space by only storing one copy of each file uploaded. Should clients frequently encrypt their files, however, savings are lost. Message-locked encryption (the mos t remarkable manifestation of which is convergent encryption) resolves this tension. However it is inherently vulnerable to brute - force attacks that can recover files fa lling into a known set. We propose an architecture that provides secure deduplicated storage opposing brute-force attacks, and realize it in a system called DupLESS. In DupLESS, clients encrypt the under message-based keys obtained from a key-server via an oblivious PRF protocol. It enables clients to store encrypted data with an current service, have the service perform deduplication on their behalf, and yet achieves strong confidentiality guarantees. We show that encryption for deduplicated storage can achieve performance and space savings near to that of using the storage service with plaintext data.

D. *Provable Data Possession at Untrusted Stores*

Authors: Giuseppe Ateniese

We introduce a model for provable data possession (PDP) that allows a client that has stored data at an untrusted server to verify that the server possesses the original data without retrieving it. The model generates probabilistic proofs of possession by sampling random sets of blocks from the server, which drastically reduces I/O costs. The client maintains a constant amount of metadata to verify the proof. The challenge/response protocol transmits a small, constant amount of data, which minimizes network communicat ion. Thus, the PDP model for remote data checking supports large data sets in widely - distributed storage systems .

E. *Remote Data Checking Using Provable Data Possession*

Authors: Giuseppe Ateniese

We suggest a model for provable data possession (PDP) that can be used for remote data checking: A client that has stored data at an untrusted server can verify that the server possesses the original data without retrieving it. The model generates probabilistic proofs of possession by sampling random sets of blocks from the server, which drastically reduces I/O costs. The client maintains a constant amount of metadata to verify the proof. The challenge/response protocol transmits a small, constant amount of data, which

minimizes network communication. Thus, the PDP model for remote data checking is lightweight and supports large data sets in distributed storage systems. The model is also robust in that it incorporates mechanisms for mitigating arbitrary amounts of data corruption

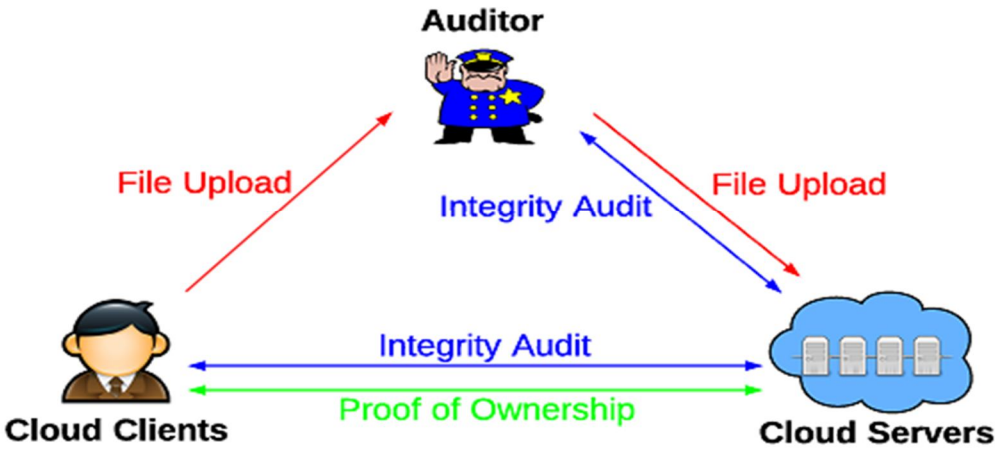
III. PROPOSED SYSTEM

We determine that our proposed SecCloud system has achieved both integrity auditing and file deduplication. However, it cannot avoid the cloud servers from knowing the content of file s having been stored. In other words, the functionalit ies of integrity auditing and secure deduplication are only imposed on plain files. In this section, we propose SecCloud+, which grant fo r integrity auditing and deduplication on encrypted files. System Model Compared with SecCloud, our recommended SecCloud+ involves further trusted entity, namely key server, which is responsible for assigning clients with secret key (accord ing to the file content) for encrypting files. This architecture is in line with the recent work. But our work is distinguished with the past work by allowing for integrity auditing on encrypted data. SecCloud+ follows the same three protocols (i.e., the file uploading protocol, the integrity auditing protocol and the proof of ownership protocol) as with SecCloud. The only anomaly is the file uploading protocol in SecCloud+ involves an additional stages for communication among cloud client and key server. That is, the client needs to communicate with the key server to get the convergent key for encrypting the uploading file before the phase in SecCloud.

A. Advantages

This plan settles the issue of past work that the computational burden at client or inspector is excessively tremendous for label era. For fulfillment of fine-grained, the usefulness of reviewing composed in SecCoud is upheld on both square level and division level. Furthermore, SecCoud likewise empowers secure deduplication. The test of deduplication on encoded is the avoidance of lexicon assault. Our proposed SecCloud framework has accomplished both respectability evaluating and document deduplication

IV. SYSTEM ARCHITECTURE



A. Cloud Clients

Cloud Clients have large data files to be stored and rely on the cloud for data maintenance and computation. They can be either individual consumers or commercial organizations.

B. Cloud Servers

Cloud Servers virtualize the resources according to the requirements of clients and expose them as storage pools. Typically, the cloud clients may buy or lease storage capacity from cloud servers, and store their individual data in these bought or rented spaces for future utilization.

C. Auditor

Auditor which helps clients upload and audit their outsourced data maintains a MapReduce cloud and acts like a certificate authority. This assumption presumes that the auditor is associated with a pair of public and private keys. Its public key is made

available to the other entities in the system.

The SecCloud system supporting file-level deduplication includes the following three protocols respectively highlighted by red, blue and green in Fig.[25]

D. File Uploading Protocol

This protocol aims at allowing clients to upload files via the auditor. Specifically, the file uploading protocol includes three phases:

1) *Phase 1 (cloud client → cloud server):* Client takes the duplicate check with the cloud server to confirm if such a file is stored in cloud storage or not before uploading a file. If there is a duplicate, another protocol called Proof of Ownership will be run between the client and the cloud storage server. Otherwise, the following protocols (including phase 2 and phase 3) are run between these two entities.
2) *Phase 2 (cloud client → auditor):* Client uploads files to the auditor, and receives a receipt from auditor.
3) *Phase 3 (auditor → cloud server):* Auditor helps generate a set of tags for the uploading file, and send them along with this file to cloud server.

E. Integrity Auditing Protocol

It is an interactive protocol for integrity verification and allowed to be initialized by any entity except the cloud server. In this protocol, the cloud server. plays the role of prover, while the auditor or client works as the verifier. This protocol includes two phases:

1) *Phase 1 (cloud client/auditor → cloud server):* Verifier (i.e., client or auditor) generates a set of challenges and sends them to the prover (i.e., cloud server).
2) *Phase 2 (cloud server → cloud client/auditor):* Based on the stored files and file tags, prover (i.e., cloud server) tries to prove that it exactly owns the target file by sending the proof back to verifier (i.e., cloud client or auditor). At the end of this protocol, verifier outputs true if the integrity verification is passed.

F. Proof of Ownership Protocol

It is an interactive protocol initialized at the cloud server for verifying that the client exactly owns a claimed file. This protocol is typically triggered along with file uploading protocol to prevent the leakage of side channel information. On the contrast to integrity auditing protocol, in PoW the cloud server works as verifier, while the client plays the role of prover. This protocol also includes two phases

1) *Phase 1 (cloud server → client)* Cloud server generates a set of challenges and sends them to the client.
2) *Phase 2 (client → cloud server):* The client responds with the proof for file ownership, and cloud server finally verifies the validity of proof. Our main objectives are as follows.

G. Integrity Auditing

The first design goal of this work is to provide the capability of verifying correctness of the remotely stored data. The integrity verification further requires two features those are public verification and stateless verification.

1) *Secure Deduplication:* The second design goal of this work is secure deduplication. In other words, it requires that the cloud server is able to decrease the storage space by keeping only one copy of the same file. Notice that, regarding to secure deduplication, our objective is distinguished from previous work [3] in that we propose a method for allowing both deduplication over files and tags.
2) *Cost-Effective:* The computational overhead for providing integrity auditing and secure deduplication should not show a major additional cost to traditional cloud storage, nor should they alter the way either uploading or downloading operation.

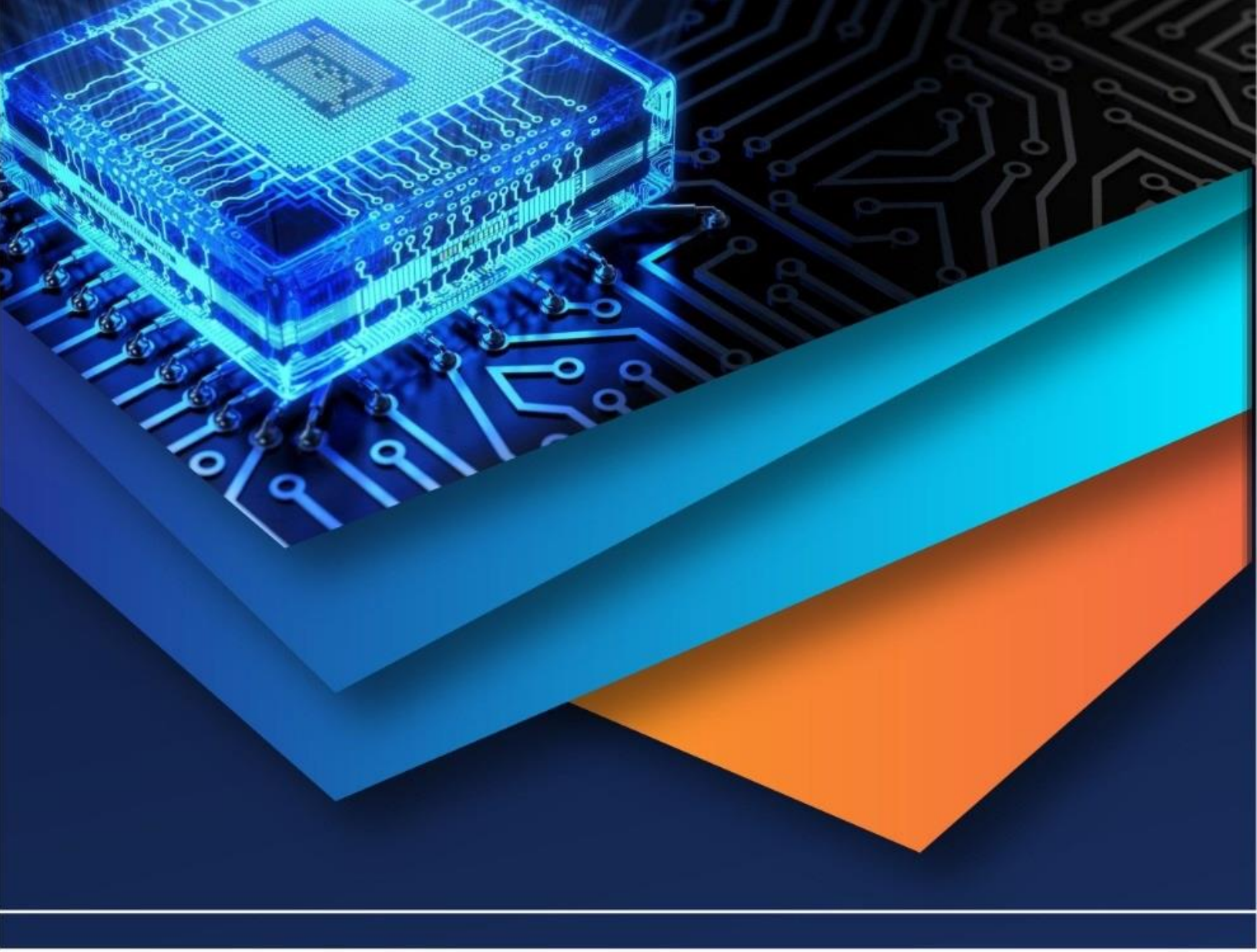
V. CONCLUSION

Aiming at achieving both data integrity and deduplication in cloud, we propose SecCloud and SecCloud+. SecCloud introduces an auditing entity with maintenance of a MapReduce cloud, which helps clients generate data tags before uploading as well as audit the integrity of data having been stored in cloud. In addition, SecCoud enables secure deduplication through introducing a Proof of Ownership protocol and preventing the leakage of side channel information in data deduplication. Compared with previous work, the computation by user in SecCloud is greatly reduced during the file uploading and auditing phases. SecCloud+ is an advanced

construction motivated by the fact that customers always want to encrypt their data before uploading, and allows for integrity auditing and secure deduplication directly on encrypted data.

REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," Communication of the ACM, vol. 53, no. 4, pp. 50–58, 2010.
[2] J. Yuan and S. Yu, "Secure and constant cost public cloud storage auditing with deduplication," in IEEE Conference on Communications and Network Security (CNS), 2013, pp. 145–153.
[3] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Proofs of ownership in remote storage systems," in Proceedings of the 18th ACM Conference on Computer and Communications Security. ACM, 2011, pp. 491–500.
[4] S. Keelveedhi, M. Bellare, and T. Ristenpart, "Dupless: Serveraided encryption for deduplicated storage," in Proceedings of the 22Nd USENIX Conference on Security, ser. SEC'13. Washington, D.C.: USENIX Association, 2013, pp. 179–194. [Online]. Available: https://www.usenix.org/conference/usenixsecurity13/technicalsessions/presentation/bellare
[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 the 14th ACM Conference on Computer and Communications Security, ser. CCS '07. New York, NY, USA: ACM, 2007, pp. 598–609.
[6] G. Ateniese, R. Burns, R. Curtmola, J. Herring, O. Khan, L. Kissner, Z. Peterson, and D. Song, "Remote data checking using provable data possession," ACM Trans. Inf. Syst. Secur., vol. 14, no. 1, pp. 12:1–12:34, 2011.
[7] G. Ateniese, R. Di Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in Proceedings of the 4th International Conference on Security and Privacy in Communication Netowrks, ser. SecureComm '08. New York, NY, USA: ACM, 2008, pp. 9:1–9:10.
[8] C. Erway, A. K¨upc¸ ¨u, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in Proceedings of the 16th ACM Conference on Computer and Communications Security, ser. CCS '09. New York, NY, USA: ACM, 2009, pp. 213–222.
[9] F. Seb´e, J. Domingo-Ferrer, A. Martinez-Balleste, Y. Deswarte, and J.-J. Quisquater, "Efficient remote data possession checking in critical information infrastructures," IEEE Trans. on Knowl. and Data Eng., vol. 20, no. 8, pp. 1034–1038, 2008.
[10] H. Wang, "Proxy provable data possession in public clouds," IEEE Transactions on Services Computing, vol. 6, no. 4, pp. 551–559, 2013.
[11] Y. Zhu, H. Hu, G.-J. Ahn, and M. Yu, "Cooperative provable data possession for integrity verification in multicloud storage," IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 12, pp. 2231–2244, 2012.
[12] H. Shacham and B. Waters, "Compact proofs of retrievability," in Proceedings of the 14th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology, ser. ASIACRYPT '08. Springer Berlin Heidelberg, 2008, pp. 90–107.
[13] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in Computer Security – ESORICS 2009, M. Backes and P. Ning, Eds., vol. 5789. Springer Berlin Heidelberg, 2009, pp. 355–370.
[14] J. Xu and E.-C. Chang, "Towards efficient proofs of retrievability," in Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security, ser. ASIACCS '12. New York, NY, USA: ACM, 2012, pp. 79–80.
[15] E. Stefanov, M. van Dijk, A. Juels, and A. Oprea, "Iris: A scalable cloud file system with efficient integrity checks," in Proceedings of the 28th Annual Computer Security Applications Conference, ser. ACSAC '12. New York, NY, USA: ACM, 2012, pp. 229–238.
[16] M. Azraoui, K. Elkhiyaoui, R. Molva, and M. O¨ nen, "Stealthguard: Proofs of retrievability with hidden watchdogs," in Computer Security - ESORICS 2014, ser. Lecture Notes in Computer Science, M. Kutyłowski and J. Vaidya, Eds., vol. 8712. Springer International Publishing, 2014, pp. 239–256.
[17] J. Li, X. Tan, X. Chen, and D. Wong, "An efficient proof of retrievability with public auditing in cloud computing," in 5th International Conference on Intelligent Networking and Collaborative Systems (INCoS), 2013, pp. 93–98.
[18] J. Li, X. Chen, M. Li, J. Li, P. Lee, and W. Lou, "Secure deduplication with efficient and reliable convergent key management," IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 6, pp. 1615–1625, June 2014.
[19] R. Di Pietro and A. Sorniotti, "Boosting efficiency and security in proof of ownership for deduplication," in Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security, ser. ASIACCS '12. New York, NY, USA: ACM, 2012, pp. 81–82.
[20] W. K. Ng, Y. Wen, and H. Zhu, "Private data deduplication protocols in cloud storage," in Proceedings of the 27th Annual ACM Symposium on Applied Computing, ser. SAC '12. New York, NY, USA: ACM, 2012, pp. 441–446.
[21] J. Douceur, A. Adya, W. Bolosky, P. Simon, and M. Theimer, "Reclaiming space from duplicate files in a serverless distributed file system," in 22nd International Conference on Distributed Computing Systems, 2002, pp. 617–624.
[22] M. Bellare, S. Keelveedhi, and T. Ristenpart, "Message-locked encryption and secure deduplication," in Advances in Cryptology – EUROCRYPT 2013, ser. Lecture Notes in Computer Science, T. Johansson and P. Nguyen, Eds. Springer Berlin Heidelberg, 2013, vol. 7881, pp. 296–312.
[23] M. Abadi, D. Boneh, I. Mironov, A. Raghunathan, and G. Segev, "Message-locked encryption for lock-dependent messages," in Advances in Cryptology – CRYPTO 2013, ser. Lecture Notes in Computer Science, R. Canetti and J. Garay, Eds. Springer Berlin Heidelberg, 2013, vol. 8042, pp. 374–391.
[24] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in Advances in Cryptology — CRYPTO 2001, ser. Lecture Notes in Computer Science, J. Kilian, Ed. Springer Berlin Heidelberg, 2001, vol. 2139, pp. 213–229.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY