International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 1, Issue 4, April 2014)

Cooperative Provable Data Possession for Integrity Verification in Multi-Cloud Storage

Thippana. Sreelatha
M.Tech , Student, CSE
AVN Institute of Engg. & Tech
Hyderabad, AP, India
sreelathareddythippana@gmail.com

Prof Shaik.Abdul Nabi
Professor & Head of the Dept, CSE
AVN Institute of Engg. & Tech
Hyderabad, AP, India

Mrs. M.Swapna
Assistant Professor, CSE
AVN Institute of Engg. & Tech
Hyderabad, AP, India

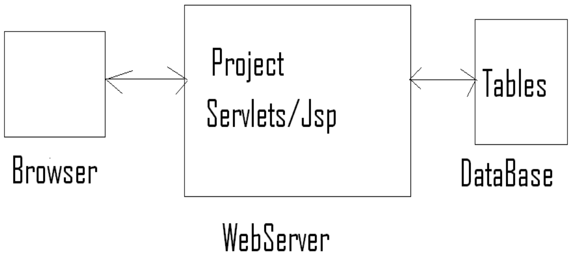
Abstract: Provable data possession (PDP) is a technique for ensuring the integrity of data in storage outsourcing. In this paper, we address the construction of an efficient PDP scheme for distributed cloud storage to support the scalability of service and data migration, in which we consider the existence of multiple cloud service providers to cooperatively store and maintain the clients' data. We present a *cooperative* PDP (CPDP) scheme based on homomorphic verifiable response and hash index hierarchy. We prove the security of our scheme based on multi-prover zero-knowledge proof system, which can satisfy completeness, knowledge soundness, and zero- knowledge properties. In addition, we articulate performance optimization mechanisms for our scheme, and in particular present an efficient method for selecting optimal parameter values to minimize the computation costs of clients and storage service providers. Our experiments show that our solution introduces lower computation and communication overheads in comparison with non-cooperative approaches.

1. INTRODUCTION

In recent years, cloud storage service has become a faster profit growth point by providing a comparably low-cost, scalable, position-independent platform for clients' data. Since cloud computing environment is constructed based on open architectures and interfaces, it has the capability to incorporate multiple internal and/or external cloud services together to provide high interoperability. We call such a distributed cloud environment as a *multi-Cloud* (or *hybrid cloud*). Often, by using virtual infrastructure management (VIM), a multi-cloud allows clients to easily access his/her resources remotely through interfaces such as Web services provided by Amazon EC2. There exist various tools and technologies for multi cloud, such as Platform VM Orchestrator, VMware v Sphere, and Ovirt. These tools help cloud providers construct a distributed cloud storage platform (DCSP) for managing clients' data. However, if such an important platform is vulnerable to security attacks, it would bring irretrievable losses to the clients. For example, the confidential data in an enterprise may be illegally accessed through a remote interface provided by a multi-cloud, or relevant data and archives may be lost or tampered with when they are stored into an uncertain storage pool outside the enterprise. Therefore, it is indispensable for cloud service providers (CSPs) to provide security techniques for managing their storage services. Provable data possession (PDP)(or proofs of retrievability (POR)) is such a probabilistic proof technique for a storage provider to prove the integrity and ownership of clients' data without downloading data. The proof-checking without downloading makes it especially important for large-size files and folders (typically including many clients' files) to check whether these data have been tampered with or deleted without downloading the latest version of data. Thus, it is able to replace traditional hash and signature functions in storage outsourcing. Various PDP schemes have been recently proposed, such as Scalable PDP and Dynamic PDP. However, these schemes mainly focus on PDP issues at untrusted servers in a *single* cloud storage provider and are not suitable for a multi-cloud environment.

2. ARCHITECTURE DESCRIPTION

Provable data possession (PDP) (or proofs of retrievability (POR)) is such a probabilistic proof technique for a storage provider to prove the integrity and ownership of clients' data without downloading data. The proof-checking without downloading makes it especially important for large-size files and folders (typically including many clients' files) to check whether these data have been tampered with or deleted without downloading the latest version of data. Thus, it is able to replace traditional hash and signature functions in storage outsourcing. Various PDP schemes have been recently proposed, such as Scalable PDP and Dynamic PDP. However, these schemes mainly focus on PDP issues at un trusted servers in a *single* cloud storage provider and are not suitable for a multi-cloud environment.



3. MODULES

Multi cloud storage

Distributed computing is used to refer to any large collaboration in which many individual personal computer owners allow some of their computer's processing time to be put at the service of a large problem. In our system the each cloud admin consist of data blocks. The cloud user uploads the data into multi cloud. Cloud computing environment is constructed based on open architectures and interfaces; it has the capability to incorporate multiple internal and/or external cloud services together to provide high interoperability. We call such a distributed cloud environment as a *multi-Cloud* .A multi-cloud allows clients to easily access his/her resources remotely through interfaces.

Cooperative PDP

Cooperative PDP (CPDP) schemes adopting zero-knowledge property and three-layered index hierarchy, respectively. In particular efficient method for selecting the optimal number of sectors in each block to minimize the computation costs of clients and storage service providers. Cooperative PDP (CPDP) scheme without compromising data privacy based on modern cryptographic techniques.

Data Integrity

Data Integrity is very important in database operations in particular and Data warehousing and Business intelligence in general. Because Data Integrity ensured that data is of high quality, correct, consistent and accessible.

Third Party Auditor

Trusted Third Party (TTP) who is trusted to store verification parameters and offer public query services for these parameters. In our system the Trusted Third Party, view the user data blocks and uploaded to the distributed cloud. In distributed cloud environment each cloud has user data blocks. If any modification tried by cloud owner a alert is send to the Trusted Third Party.

Cloud User

The Cloud User who has a large amount of data to be stored in multiple clouds and have the permissions to access and manipulate stored data. The User's Data is converted into data blocks. The data blocks are uploaded to the cloud. The TPA views the data blocks and Uploaded in multi cloud. The user can update the uploaded data. If the user wants to download their files, the data's in multi cloud is integrated and downloaded.

4. PROPOSED SYSTEM

In this paper, we address the problem of provable data possession in distributed cloud environments from the following aspects: *high security*, *transparent Verification*, and *high performance*. To achieve these goals, we first propose a Verification framework for multi-cloud storage along with two fundamental techniques: hash index hierarchy (HIH) and homomorphic verifiable response (HVR). We then demonstrate that the possibility of constructing a cooperative PDP (CPDP) scheme without compromising data privacy based on modern cryptographic techniques, such as interactive proof system (IPS).

5. DESIGN

Design is a meaningful engineering representation of something that is to be built. Software design is a process through which the requirements are translated into a representation of the software. Design is the place where quality is fostered in software engineering. Design is the perfect way to accurately translate a customer's requirement in to a finished software product. Design creates a representation or model, provides detail about software data structure, architecture, interfaces and components that are necessary to implement a system. This chapter discusses about the design part of the project. Here in this document the various UML diagrams that are used for the implementation of the project are discussed.

The Unified Modeling Language (UML) is a visual modeling language used to specify, visualize, construct and document a software intensive system. The embedded real-time software systems encountered in applications such as telecommunications, school systems, aerospace, and defense typically tends to be large and extremely complex. It is crucial in such systems that the software is designed with a sound architecture. A good architecture not only simplifies construction of the initial system, but also, readily accommodates changes forced by a steady stream of new requirements.

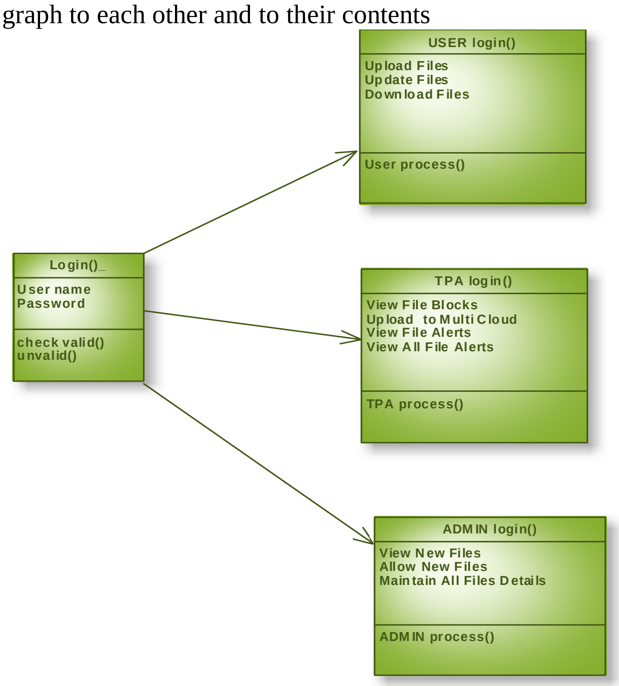
The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software.

The primary goals in the design of the UML are: Provide users with a ready-to-use, expressive visual modeling language so they can develop and exchange meaningful models. Provide extensibility and specialization mechanisms to extend the core concepts. Be independent of particular programming languages and development processes. Provide a formal basis for understanding the modeling language. Encourage the growth of the OO tools market. Support higher-level development concepts such as collaborations, frameworks, patterns and components. Integrate best practices.

Modeling

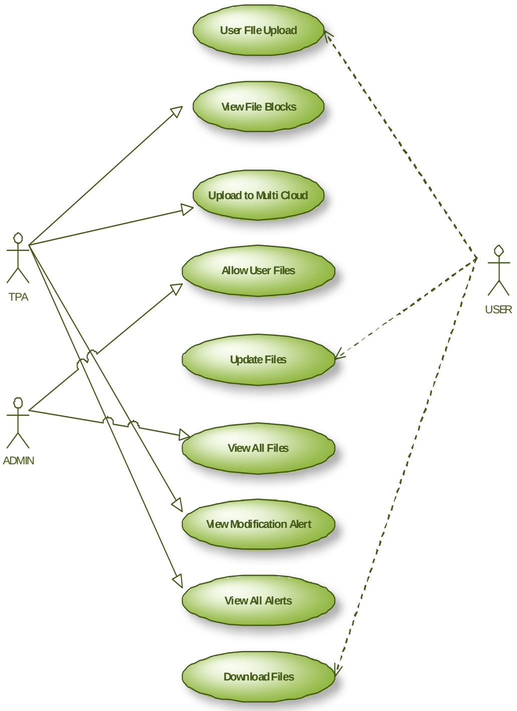
Class Diagram

UML Class diagram shows the static structure of the model. The class diagram is a collection of static modeling elements, such as classes and their relationships, connected as a graph to each other and to their contents

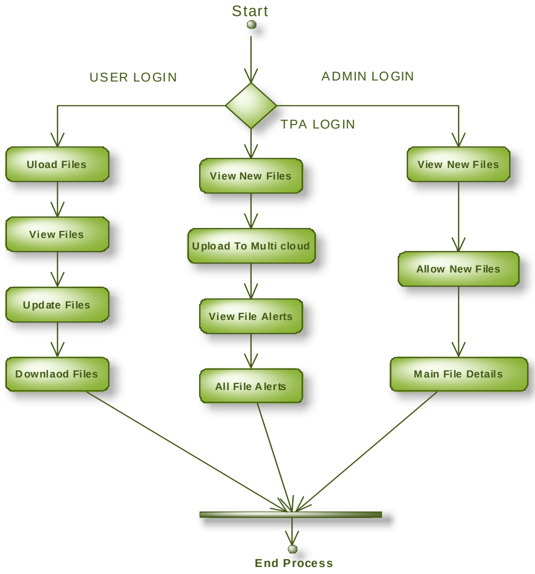


Use Case Diagram

A use case diagram is a graph of actors, a set of use cases enclosed by a system boundary, communication (participation) associations between the actors and users and generalization among use cases. The use case model defines the outside (actors) and inside (use case) of the system's behavior

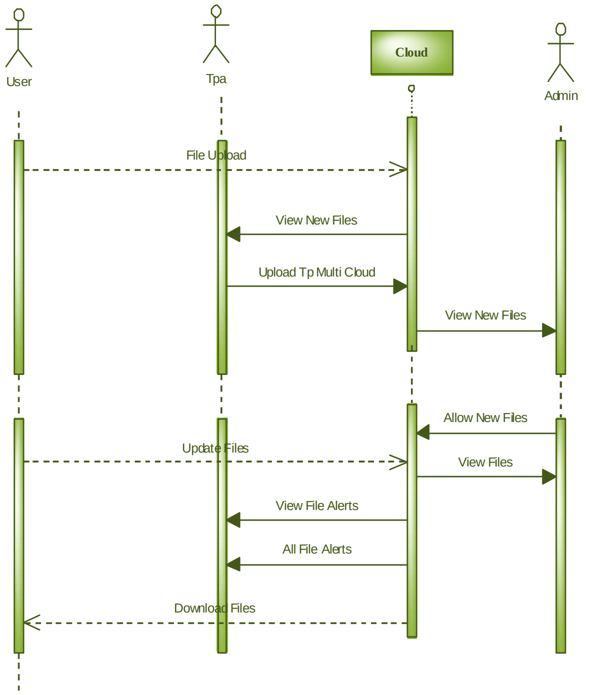


ACTIVITY DIAGRAM:



Sequence Diagram

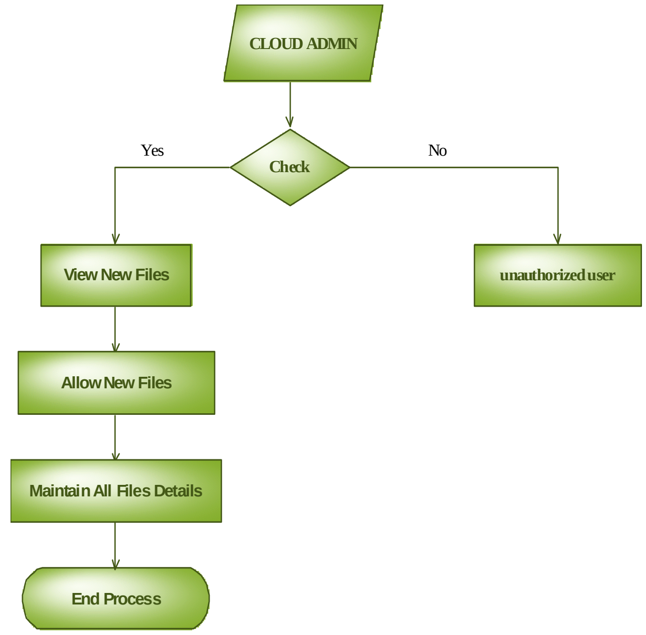
Sequence diagram are an easy and intuitive way of describing the behavior of a system by viewing the interaction between the system and its environment. A Sequence diagram shows an interaction arranged in a time sequence. A sequence diagram has two dimensions: vertical dimension represents time; the horizontal Dimension represents different objects. The vertical line is called is the object's life line. The lifeline represents the object's existence during the interaction.



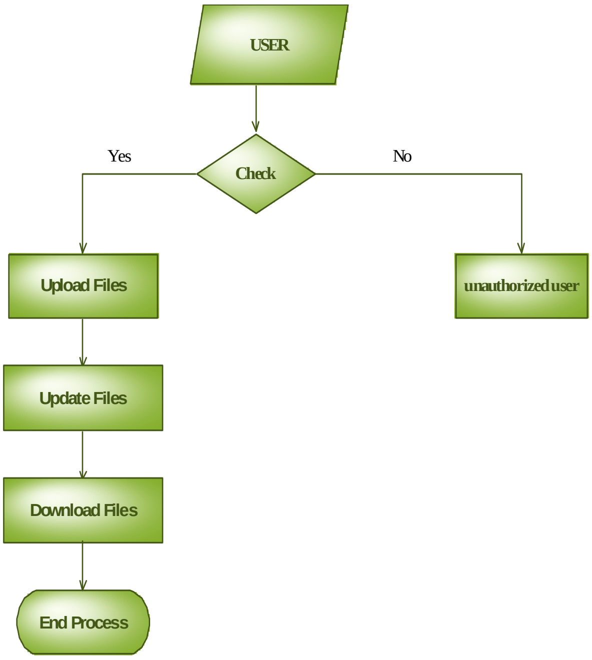
Data Flow Diagram

The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of the input data to the system, various processing carried out on these data, and the output data is generated by the system.

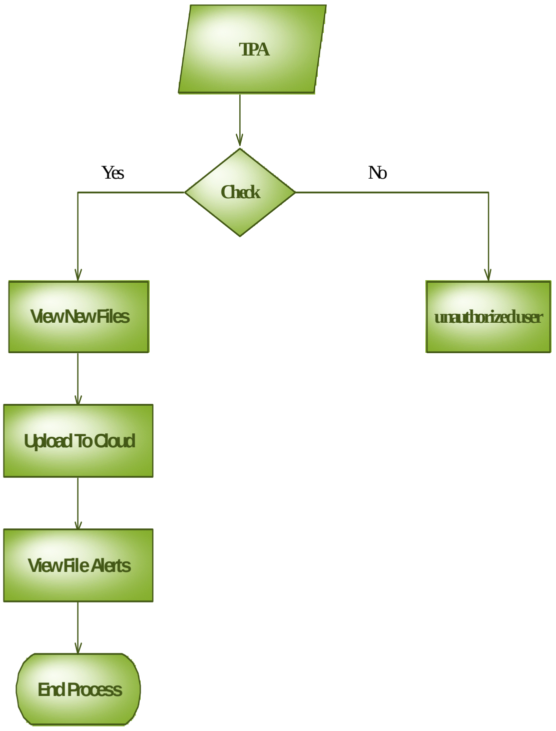
Data Flow Diagram:
SYSTEM DESIGN **:(** Admin**)**



User

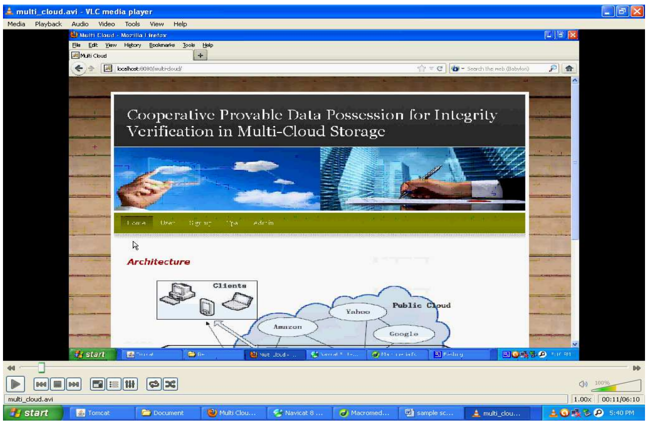


TPA



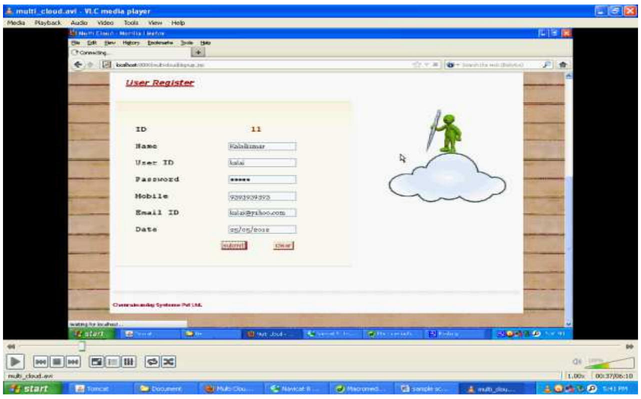
8. TESTING & SCREENS

Screens



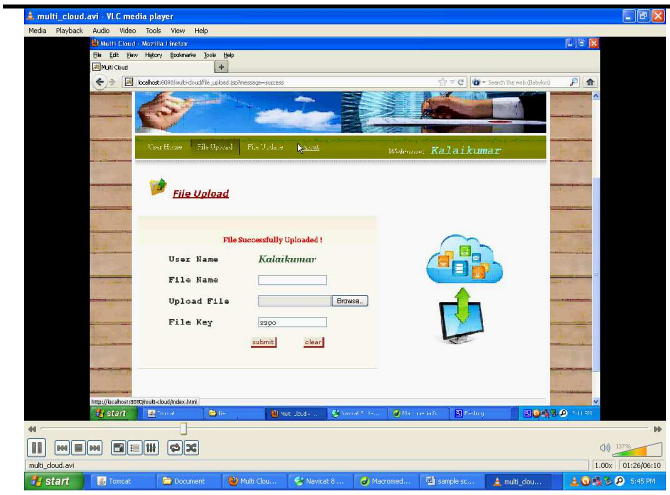
Cooperative Provable Data Possession for Integrity Verification in Multi-Cloud Storage

Architecture



User Register

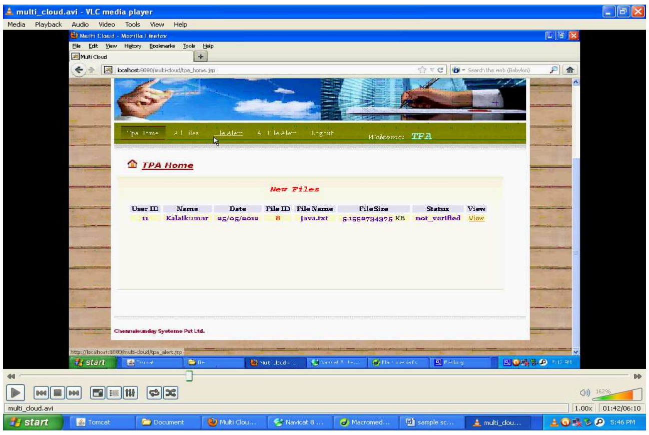
ID 11
Name Kalaikumar
User ID kalai
Password •••••
Mobile 9393939393
Email ID kalai@yahoo.com
Date 05/05/2012
submit clear


multi_cloud.avi - VLC media player
Multi Cloud - Mozilla Firefox
localhost:8080/multi-cloud/file_upload.jsp?message=success
User Home
File Upload
File Update
Logout
Welcome: Kalaikumar
Browse..
submit
clear
http://localhost:8080/multi-cloud/index.html
start
multi_cloud.avi
1.00x
01:26/06:10
5:45 PM

File Upload

File Successfully Uploaded !

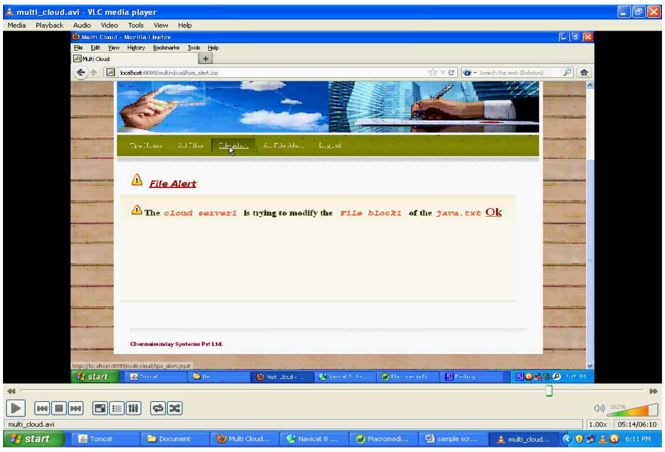
User Name Kalaikumar
File Name
Upload File
File Key 2290


multi_cloud.avi - VLC media player
Multi Cloud - Mozilla Firefox
localhost:8080/multi-cloud/tpa_home.jsp
Tpa Home
All Files
File Alert
All File Alert
Logout
Welcome: TPA
Chennaisunday Systems Pvt Ltd.
start
multi_cloud.avi
1.00x
01:42/06:10
5:46 PM

TPA Home

New Files

User ID	Name	Date	File ID	File Name	FileSize	Status	View
11	Kalaikumar	25/05/2012	8	java.txt	5.1552734375 KB	not_verified	View


multi_cloud.avi - VLC media player
Multi Cloud - Mozilla Firefox
localhost:8080/multi-cloud/tpa_alert.jsp
Tpa Home
All Files
File Alert
All File Alert
Logout
Chennaisunday Systems Pvt Ltd.
start
multi_cloud.avi
1.00x
05:14/06:10
6:11 PM

File Alert

The cloud server1 is trying to modify the File block1 of the java.txt Ok

CONCLUSION

We presented the construction of an efficient PDP scheme for distributed cloud storage. Based on homomorphic verifiable response and hash Index hierarchy, we have proposed a cooperative PDP scheme to support dynamic scalability on multiple storage servers. We also showed that our scheme provided all security properties required by zero knowledge interactive proof system, so that it can resist various attacks even if it is deployed as a public audit service in clouds. Furthermore, we optimized the probabilistic query and periodic verification to improve the audit performance. Our experiments clearly

Demonstrated that our approaches only introduce a small amount of computation and communication overheads. Therefore, our solution can be treated as a new candidate for data integrity verification in outsourcing data storage systems. As part of future work, we would extend our work to explore more effective CPDP constructions. Finally, it is still a challenging problem for the generation of tags with the length irrelevant to the size of data blocks. We would explore such a issue to provide the support of variable-length block verification.

REFERENCES

[1]. B. Sotomayor, R. S. Montero, I. M. Llorente, and I. T. Foster, Virtual infrastructure management in private and hybrid clouds," IEEE Internet Computing, vol. 13, no. 5, pp. 14–22,2009.

[2]. G. Ateniese, R. C. Burns, R. Curtmola, J. Herring, L. Kissner,Z. N. J. Peterson, and D. X. Song, "Provable data possession at untrusted stores," in ACM Conference on Computer andCommunications Security, P. Ning, S. D. C. di Vimercati, and P. F. Syverson, Eds. ACM, 2007, pp. 598–609.

[3]. A. Juels and B. S. K. Jr., "Pors: proofs of retrievability for large files," in ACMConference on Computer and Communications
Security, P. Ning, S. D. C. di Vimercati, and P. F. Syverson, Eds. ACM, 2007, pp. 584–597.

[4]. G. Ateniese, R. D. 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, SecureComm, 2008, pp. 1–10.

[5]. C. C. Erway, A. K¨upc¸ ¨u, C. Papamanthou, and R. Tamassia,"Dynamic provable data possession," in ACM Conference on Computer and Communications Security, E. Al-Shaer, S. Jha, and A. D. Keromytis, Eds. ACM, 2009, pp. 213–222.

[6]. H. Shacham and B. Waters, "Compact proofs of retrievability," in ASIACRYPT, ser. Lecture Notes in Computer Science,J. Pieprzyk, Ed., vol. 5350. Springer, 2008, pp. 90–107.

[7]. Q. Wang, C.Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in ESORICS, ser. Lecture Notes in Computer Science, M. Backes and P. Ning, Eds., vol. 5789. Springer, 2009, pp. 355–370.

[8]. Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S. S. Yau, "Dynamic audit services for integrity verification of outsourced storages in clouds," in SAC, W. C. Chu, W. E. Wong, M. J. Palakal, and C.-C. Hung, Eds. ACM, 2011, pp. 1550–1557.

[9]. K. D. Bowers, A. Juels, and A. Oprea, "Hail: a high-availability and integrity layer for cloud storage," in ACM Conference on Computer and Communications Security, E. Al-Shaer, S. Jha, and A. D. Keromytis, Eds. ACM, 2009, pp. 187–198.

[10]. Y. Dodis, S. P. Vadhan, and D. Wichs, "Proofs of retrievability via hardness amplification," in TCC, ser. Lecture Notes in
Computer Science, O. Reingold, Ed., vol. 5444. Springer, 2009,pp. 109–127.

[11]. L. Fortnow, J. Rompel, and M. Sipser, "On the power of multiprover interactive protocols," in Theoretical Computer Science, 1988, pp. 156–161.

[12]. Y. Zhu, H. Hu, G.-J. Ahn, Y. Han, and S. Chen, "Collaborative integrity verification in hybrid clouds," in IEEE Conference on the 7th International Conference on Collaborative Computing: Networking, Applications and Worksharing, CollaborateCom, Orlando, Florida, USA, October 15-18, 2011, pp. 197–206.

[13]. M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A berkeley view of cloud computing," EECS Department, University of California, Berkeley, Tech. Rep., Feb 2009.

[14]. D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in Advances in Cryptology (CRYPTO'2001), vol. 2139 of LNCS, 2001, pp. 213–229.

[15]. O. Goldreich, Foundations of Cryptography: Basic Tools. Cambridge University Press, 2001.

About the authors

THIPPANA SREELATHA has obtained her B.Tech in Computer Science Engineering from JNTU HYDERABAD and currently pursuing her M.Tech in CSE at AVN Institute of Engineering & Technology, Hyderabad.

Prof Shaik.Abdul Nabi is the **Head of the Dept.** of CSE, AVN Inst.Of Engg.& Tech, Hyderabad, AP, India. He completed his B.E (Computer Science) from Osmania University, A.P. He received his M.Tech. from JNTU Hyderabad campus and he submitted his Ph.D. thesis in the area of Web Mining from AcharyaNagarjuna University, Guntur, AP, India. He is a certified professional by Microsoft. His expertise areas are Data warehousing and Data Mining, Data Structures & UNIX Networking Programming.

Mrs. M.Swapna is an **Assistant Professor** of CSE, AVN Inst.Of Engg.& Tech, Hyderabad, AP, India. She completed her B.Tech (IT) from JNTU Hyderabad, A.P. She received her M.Tech (DSCE) from JNTU Hyderabad. Her areas of interest are Cloud Computing, Data warehousing and Data Mining & Data Base Systems.