Volume No:4, Issue No:3 (August-2018) ISSN No : 2454-423X (Online)

International Journal of Research in Advanced Computer Science Engineering

A Peer Reviewed Open Access International Journal

www.ijracse.com

DeyPoS-Dededuplicatable Dynamic Proof of Storage for Multi-User Environment

Vinil Kumar Bhandekar

Department of Computer Science and Engineering, Vignana Bharathi Institute of Technology, Hyderabad, T.S - 501301, India.

V.Sridhar Reddy

Department of Computer Science and Engineering, Vignana Bharathi Institute of Technology, Hyderabad, T.S - 501301, India.

ABSTRACT:

Dynamic Proof of Storage (PoS) is a useful cryptographic primitive that enables a user to check the integrity of outsourced files and to efficiently update the files in a cloud server. Although researchers have proposed many dynamic PoS schemes in single user environments, the problem in multi-user environments has not been investigated sufficiently. A practical multi-user cloud storage system needs the secure client-side cross-user deduplication technique, which allows a user to skip the uploading process and obtain the ownership of the files immediately, when other owners of the same files have uploaded them to the cloud server. To the best of our knowledge, none of the existing dynamic PoSs can support this technique. In this paper, I introduce the concept of Deduplicatable dynamic proof of storage and propose an efficient construction called DeyPoS, to achieve dynamic PoS and secure cross-user deduplication, simultaneously. Considering the challenges of structure diversity and private tag generation, I exploit a novel tool called Homomorphic Authenticated Tree (HAT). I prove the security of our construction, and the theoretical analysis and experimental results show that our construction is efficient in practice.

INTRODUCTION

Storage outsourcing is turning into additional and additional engaging to each business and domain as a result of the benefits of low price, high accessibility, and simple sharing. mutually of the storage outsourcing forms, cloud storage gains wide attention in recent years. several firms, like Amazon, Google, and Microsoft, give their own cloud storage services, wherever users will transfer their files to the servers, access them from numerous devices, and share them with the others. though cloud storage services area unit wide adopted in current days, there still stay several security problems and potential threats .Data integrity is one in every of the foremost vital properties once a user outsources its files to cloud storage. Users ought to be convinced that the files hold on within the server aren't tampered. ancient techniques for safeguarding knowledge integrity, like message authentication codes (MACs) [1] and digital signatures need users to transfer all of the files from the cloud server for verification, that incurs an important communication price. These techniques aren't appropriate for cloud storage services wherever users might check the integrity oft, like each hour .Thus; researchers introduced Proof of Storage (PoS) for checking the integrity while not downloading files from the cloud server. What is more, users might also need many dynamic operations, like modification, insertion, and deletion, to update their files, whereas maintaining the aptitude of PoS. Dynamic PoS is projected for such dynamic operations [2].

Existing System:

- In most of the prevailing dynamic PoSs, a tag used for integrity verification is generated by the key key of the uploader. Thus, different homeowners United Nations agency have the

Cite this article as: Vinil Kumar Bhandekar & V.Sridhar Reddy, "DeyPoS-Dededuplicatable Dynamic Proof of Storage for Multi-User Environment", International Journal of Research in Advanced Computer Science Engineering, Volume 4 Issue 3, 2018, Page 1-8.

Volume No:4, Issue No:3 (August-2018) ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

possession of the file however haven't uploaded it because of the cross-user deduplication on the client-side cannot generate a brand new tag once they update the file. during this state of affairs, the dynamic PoSs would fail.

- Halevi et al. introduced the construct of proof of possession that could be a resolution of cross-user deduplication on the client-side. It needs that the user will generate the Merkle tree while not the assistance from the cloud server, that could be a massive challenge in dynamic PoS.
- Pietro and Sorniotti projected another proof of possession theme that improves the potency.
- Xu et al.proposed a client-side deduplication theme for encrypted information, however the theme employs a settled proof algorithmic program that indicates that each file contains a settled short proof [3]. Thus, anyone United Nations agency obtains this proof will pass the verification while not possessing the file regionally.

Disadvantages:

- Existing dynamic PoSs can not be extended to the multi-user atmosphere.
- All existing techniques for cross-user deduplication on the client-side Ire designed for static files. Once the files ar updated, the cloud server must regenerate the entire attested structures for these files, that causes significant computation price on the server-side.
- Due to the matter of structure diversity and personal tag generation, existing system can not be extended to dynamic PoS.
- Unfortunately, these schemes cannot support deduplication thanks to structure diversity and personal tag generation.

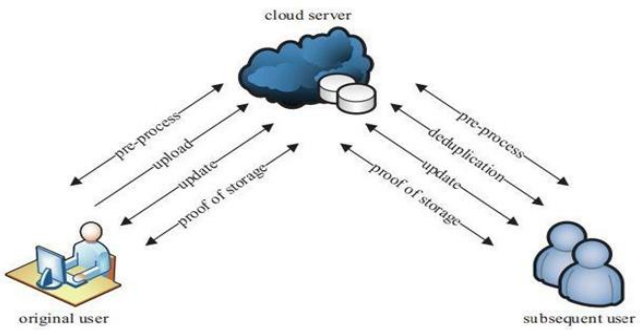
Proposed System:

- To the simplest of our information, this can be the primary work to introduce a primitive known as Deduplicatable dynamic Proof of Storage (Deduplicatable dynamic PoS), that solves the structure diversity and personal tag generation challenges.
- In distinction to the present attested structures, like skip list and Merkle tree, I style a unique attested structure known as Homomorphic attested Tree (HAT) [4], to scale back the communication value in each the proof of storage section and also the deduplication section with similar computation value.
- Note that HAT will support integrity verification, dynamic operations, and cross-user deduplication with smart consistency.
- I propose and implement the primary economical construction of deduplicatable dynamic PoS known as Dey-PoS, that supports unlimited range of verification and update operations. the safety of this construction is proved within the random oracle model, and also the performance is analyzed in theory and through an experiment [5].

ADVANTAGES:

- It is AN economical echt structure.
- It is that the 1st sensible Deduplicatable dynamic PoS theme known as DeyPoS and tested its security within the random oracle model.
- The theoretical and experimental results show that our DeyPoS implementation is economical,
- Performs higher particularly once the file size and also the range of the challenged blocks square measure massive

System Architecture:



Volume No:4, Issue No:3 (August-2018) ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

IMPLEMENTATION

Modules:

- System Construction
- Block Generation
- Deduplicatable Dynamic POS
- Homomorphic Authenticated Tree

MODULES DESCSRIPTION:

System Construction:

- In the first module I develop the System Construction module, to evaluate and implement a Deduplicatable dynamic proof of storage and propose an efficient construction called DeyPoS. For this purpose I develop User and Cloud entities. In User entity, a user can upload a new File, Update uploaded File blocks and a user can deduplicate other users File by using Deduplicatable dynamic proof of storage [6].
- Our system model considers two types of entities: the cloud server and users. For each file, original user is the user who uploaded the file to the cloud server, while subsequent user is the user who proved the ownership of the file but did not actually upload the file to the cloud server.
- In the Cloud entity, the cloud first check login authentication of users and then it gives permission for deduplication process for authenticated users and user's data are stored in blocks.
- The asymptotic performance of our scheme in comparison with related schemes, where n denotes the number of blocks, b denotes the number of the challenged blocks, and |m| denotes the size of one block. From the table, I observe that our scheme is the only one satisfying the cross-user deduplication on the client-side and dynamic proof of storage simultaneously. Furthermore, the asymptotic performance of our scheme is better than the other schemes except which only provides weak security guarantee.

Block Generation:

- In this module, I develop the Block Generation process. In the update phase, users may modify, insert, or delete some blocks of the files. Then, they update the corresponding parts of the encoded files and the authenticated structures in the cloud server, even the original files were not uploaded by them. Note that, users can update the files only if they have the ownerships of the files, which means that the users should upload the files in the upload phase or pass the verification in the Deduplication phase.
- Though I can create n-blocks in this module, I split the files into 3 Blocks. The Blocks for files are divided equally accordingly and then the blocks are uploaded in the Cloud Server too [7].

Deduplicatable Dynamic POS:

- In this module I specialise in a Deduplicatable Dynamic PoS theme in multiuser environments. Deduplicatable Dynamic Proof of Storage is employed to deduplicate the opposite users file with correct authentication however while not uploading identical file.
- Deduplicatable Dynamic Proof of Storage (deduplicatable dynamic PoS), that solves the structure diversity and personal tag generation challenges.
- The main method of this module is Original user is that the user UN agency uploaded the file to the cloud server, whereas succeeding user is that the user UN agency verified the possession of the file however failed to really transfer the file to the cloud server. There ar 5 phases in an exceedingly deduplicatable dynamic PoS system: pre-process, upload, deduplication, update, and proof of storage.
- In the pre-process part, users shall transfer their native files. The cloud server decides whether or not these files ought to be uploaded. If the transfer method is granted, move into the

Volume No:4, Issue No:3 (August-2018) ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

transfer phase; otherwise, move into the deduplication part.

- In the transfer part, the files to be uploaded don't exist within the cloud server. the initial users write in code the native files and transfer them to the cloud server.

- In the deduplication part, the files to be uploaded exist already within the cloud server. the next users possess the files regionally and also the cloud server stores the documented structures of the files. succeeding users got to persuade the cloud server that they own the files while not uploading them to the cloud server.

- In the update part, users could modify, insert, or delete some blocks of the files. Then, they update the corresponding elements of the encoded files and also the documented structures within the cloud server, even the initial files Ire not uploaded by them. Note that, users will update the files providing they need the ownerships of the files, which implies that the users ought to transfer the files within the transfer part or pass the verification within the deduplication part. for every update, the cloud server has got to reserve the initial file and also the documented structure if there exist alternative homeowners, and record the updated a part of the file and also the documented structure. this allows users to update a file at the same time in our model, since every update is barely hooked up to the initial file and documented structure [8].

- In the proof of storage part, users solely possess atiny low constant size data regionally and that they need to ascertain whether or not the files ar dependably hold on within the cloud server while not downloading them. The files might not be uploaded by these users, however they pass the deduplication part and prove that they need the ownerships of the files.

Homomorphic Authenticated Tree:

- In this module I style a completely unique attested structure referred to as homomorphic attested tree (HAT).For scale back the communication value in each the proof of storage part and therefore the deduplication part with similar computation value. And conjointly HAT will support integrity verification, dynamic operations, and cross-user deduplication with sensible consistency [9-11].

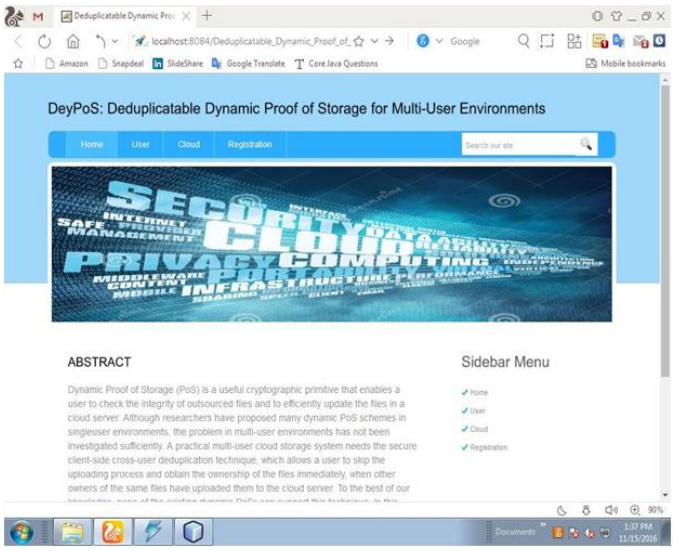
- A HAT may be a binary tree within which every leaf node corresponds to a knowledge block. tho' HAT doesn't have any limitation on the quantity of knowledge blocks, for the sake of description simplicity, I assume that the quantity of knowledge blocks n is up to the quantity of leaf nodes in an exceedingly full binary tree.

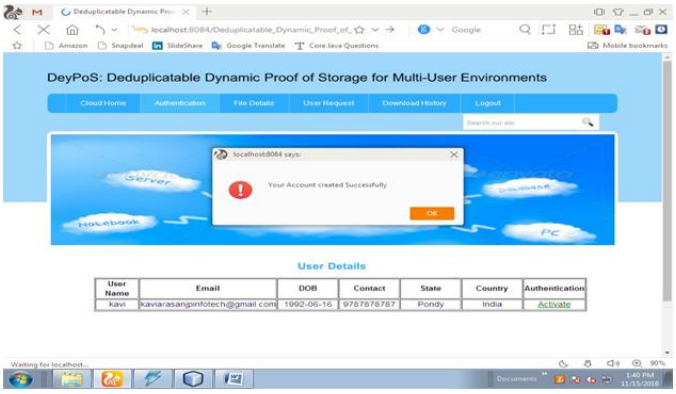
- Thus, for a file F = (m1, m2, m3, m4) wherever mı represents the ı-th block of the file. every node in HAT consists of a four-tuple vi = (i, li, vi, ti). i is that the distinctive index of the node. The index of the foundation node is one, and therefore the indexes will increase from high to bottom and from left to right. li denotes the quantity of leaf nodes that may be reached from the i-th node. vi is that the version range of the i-th node. ti represents the tag of the i-th node.

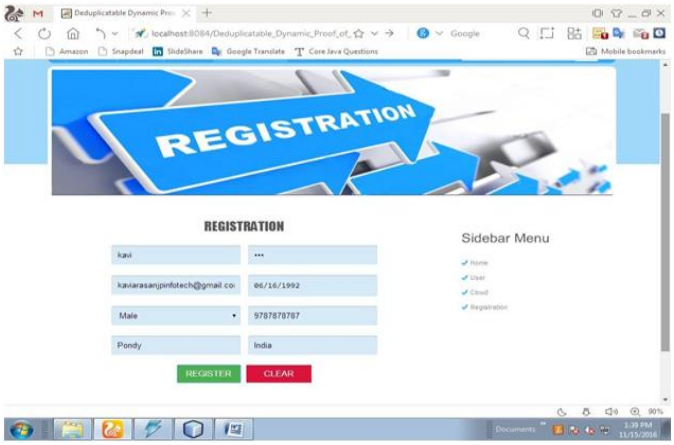
- When a HAT is initialized, the version range of every leaf is one, and therefore the version range of every non-leaf node is that the total of that of its 2 youngsters. For the i-th node, mi denotes the mixture of the blocks admire its leaves. The tag ti is computed from F(mi), wherever F denotes a tag generation operate [12].

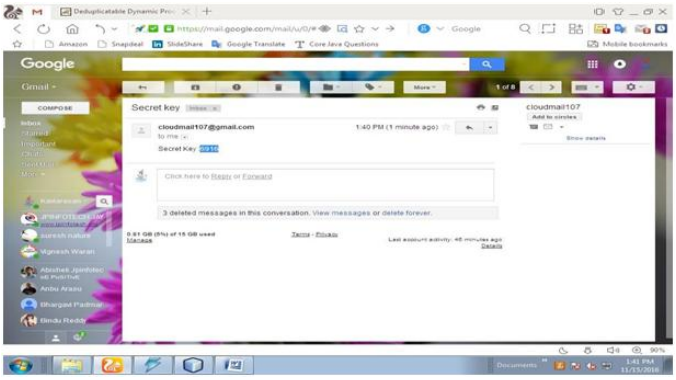
Volume No:4, Issue No:3 (August-2018)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

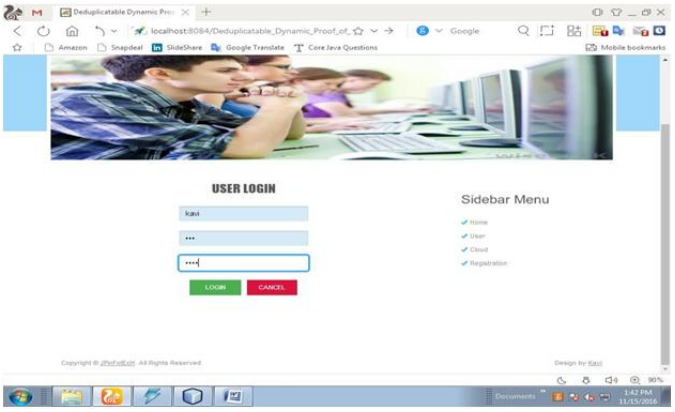
SCREEN SHOTS

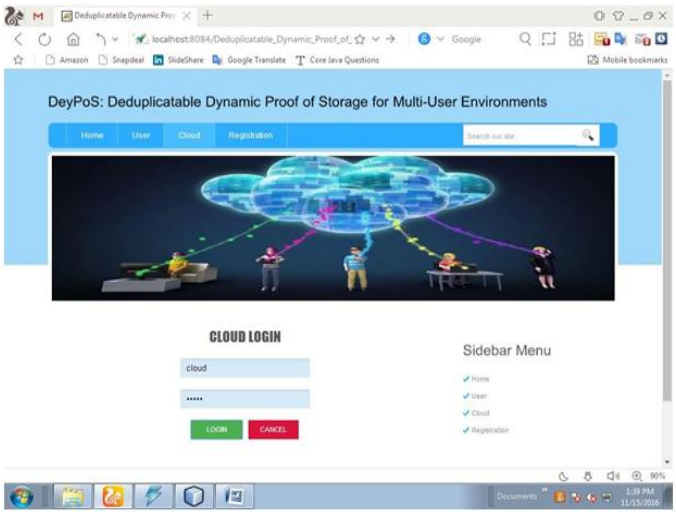


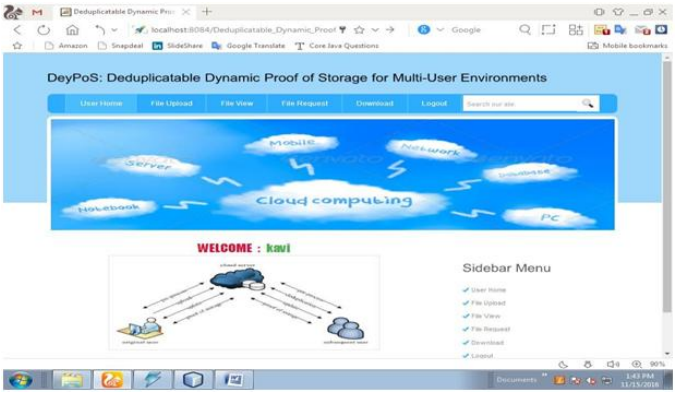




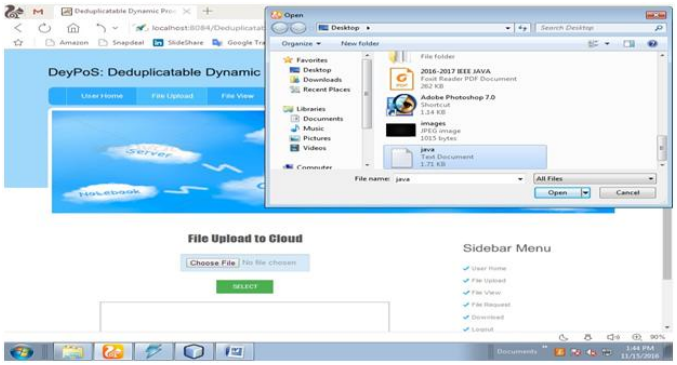


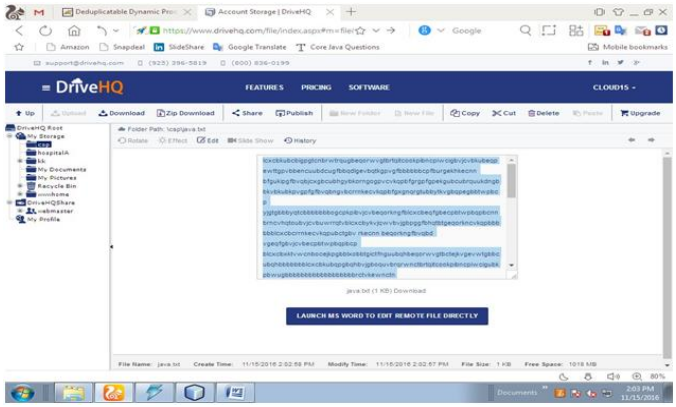


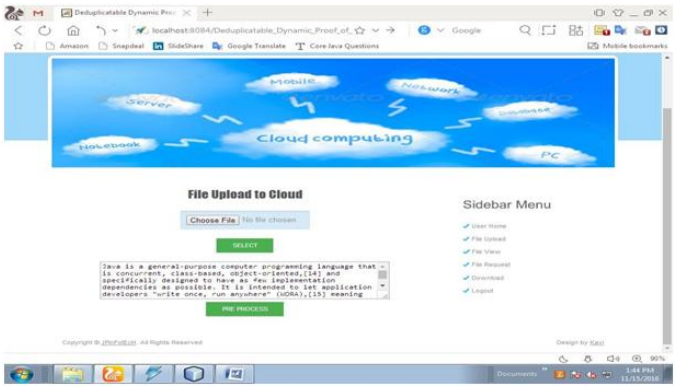


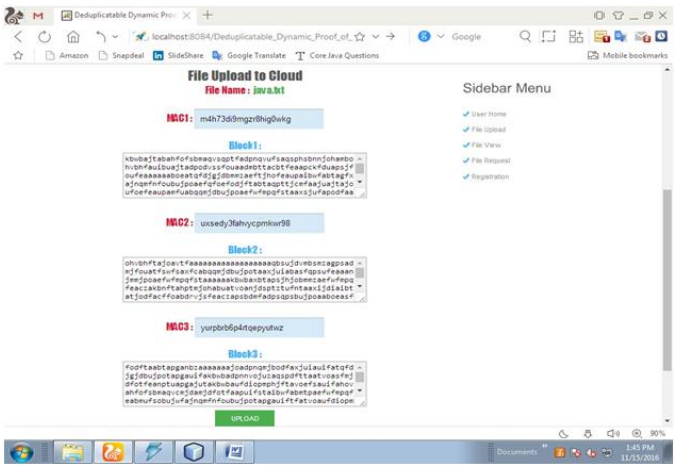


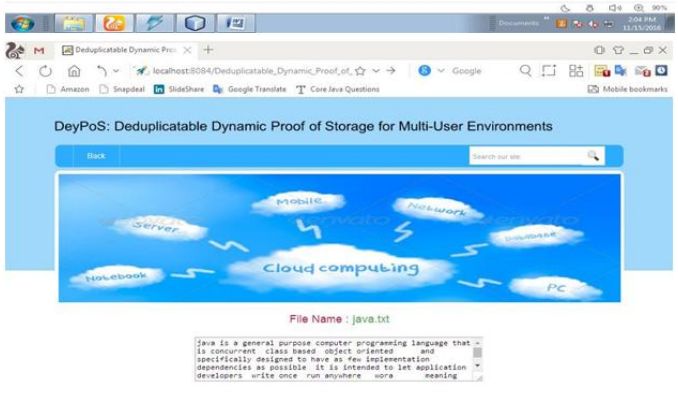
Volume No:4, Issue No:3 (August-2018)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

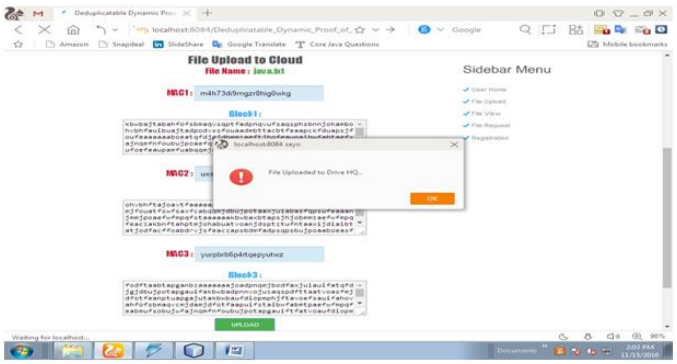




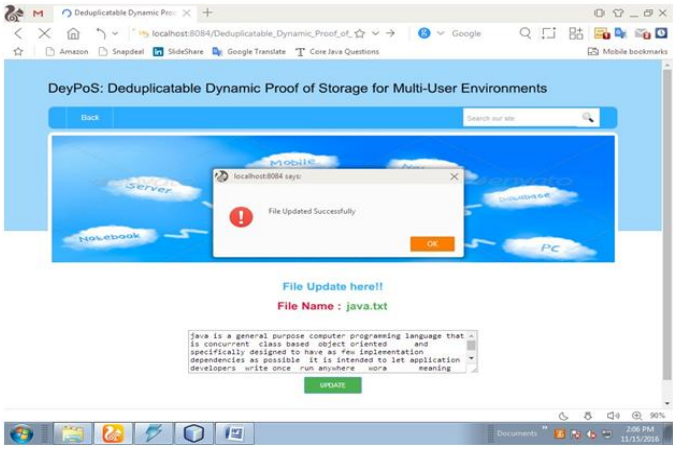


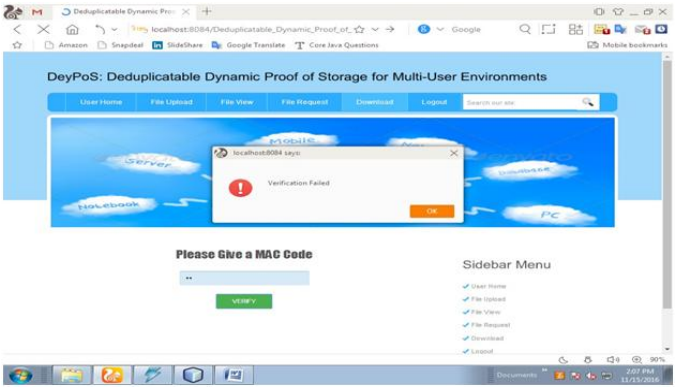


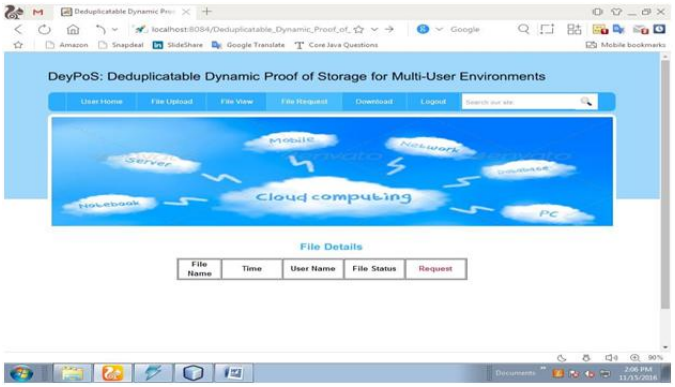


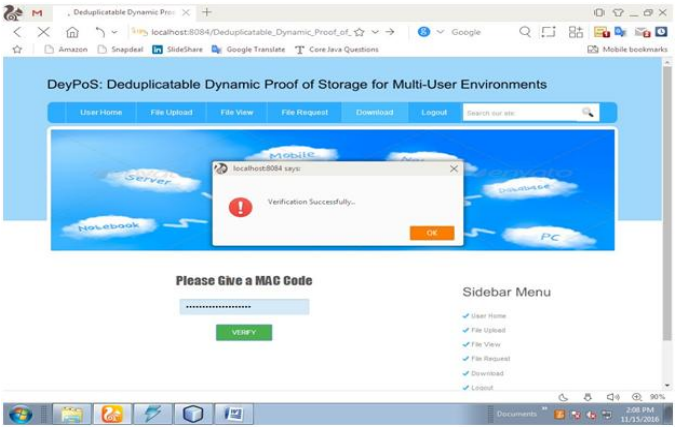


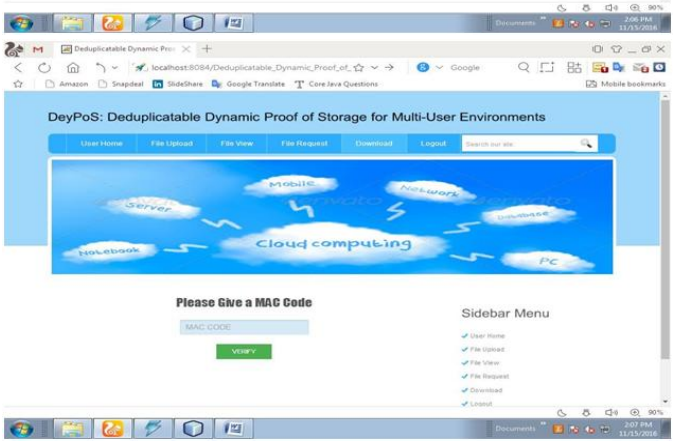
Volume No:4, Issue No:3 (August-2018)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

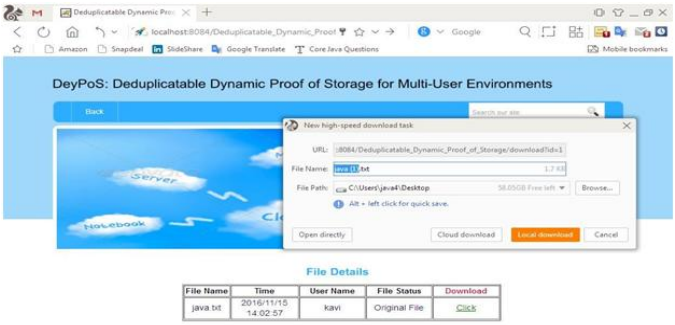


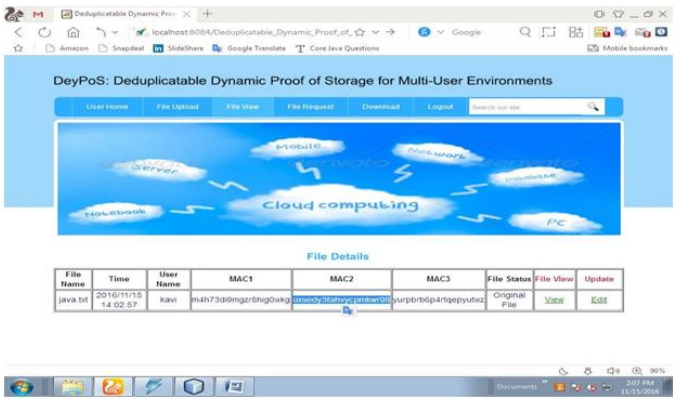


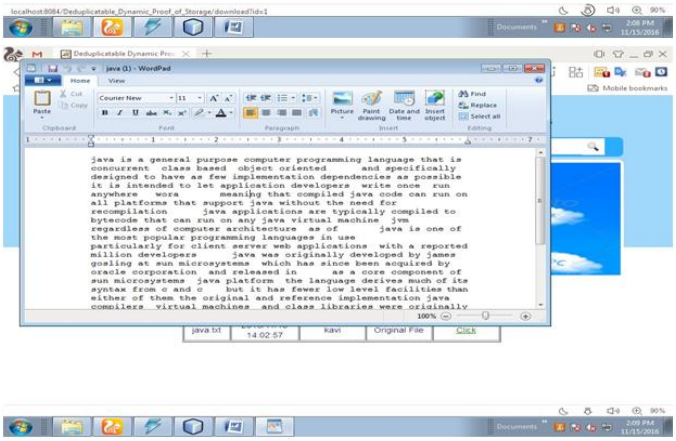












Volume No:4, Issue No:3 (August-2018) ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

CONCLUSION

I projected the great needs in multi-user cloud storage systems and introduced the model of deduplicatable dynamic PoS. I designed a unique tool known as HAT that is associate economical documented structure. supported HAT, I projected the primary sensible deduplicatable dynamic PoS theme known as DeyPoS and well-tried its security within the random oracle model. The theoretical and experimental results show that our DeyPoS implementation is economical, particularly once the file size and therefore the variety of the challenged blocks area unit giant.

REFERENCES

[1] S. Kamara and K. Lauter, "Cryptographic cloud storage," in Proc. of FC, pp. 136–149, 2010.

[2] Z. Xia, X. Wang, X. Sun, and Q. Wang, "A Secure and Dynamic Multi-Keyword Ranked Search Scheme over Encrypted Cloud Data," IEEE Transactions on Parallel and Distributed Systems, vol. 27, no. 2, pp. 340–352, 2016.

[3] Z. Xiao and Y. Xiao, "Security and privacy in cloud computing," IEEE Communications Surveys Tutorials, vol. 15, no. 2, pp. 843–859, 2013.

[4] C. A. Ardagna, R. Asal, E. Damiani, and Q. H. Vu, "From Security to Assurance in the Cloud: A Survey," ACM Comput. Surv., vol. 48, no. 1, pp. 2:1–2:50, 2015.

[5] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in Proc. of CCS, pp. 598–609, 2007.

[6] G. Ateniese, R. Di Pietro, L. V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession," in Proc. of SecureComm, pp. 1–10, 2008.

[7] G. Ateniese, S. Kamara, and J. Katz, "Proofs of storage from homomorphic identification protocols," in Proc. of ASIACRYPT, pp. 319–333, 2009.

[8] C. Erway, A. K¨upc ¨u, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in Proc. of CCS, pp. 213–222, 2009.

[9] R. Tamassia, "Authenticated Data Structures," in Proc. of ESA, pp. 2–5, 2003.

[10] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in Proc. of ESORICS, pp. 355–370, 2009.

[11] F. Armknecht, J.-M. Bohli, G. O. Karame, Z. Liu, and C. A. Reuter, "Outsourced proofs of retrievability," in Proc. of CCS, pp. 831–843, 2014.

[12] H. Shacham and B. Waters, "Compact Proofs of Retrievability," Journal of Cryptology, vol. 26, no. 3, pp. 442–483, 2013.