Middle-East Journal of Scientific Research 24 (1): 43-50, 2016
ISSN 1990-9233
© IDOSI Publications, 2016
DOI: 10.5829/idosi.mejsr.2016.24.01.22851

Light Weight Multi User Profile Inference Model for Efficient Public Auditing and Shared Data Management in Cloud Environment

[1]A. Kalaivani and [2]B. Ananthi

[1]MCA, Nehru Institute of Information Technology & Management, Coimbatore, India
[2]Vellalar College for Women, Thindal, Erode, India

Abstract: A light weight user profile inference model has been proposed in this paper to improve the efficiency of public auditing and data management in cloud environment. The method maintains multi user profiles which has information about the rights provided. The cloud service provider publish resources by performing block based encryption technique. The registered user will be given with the key parameters and the data owner specifies profile users who can access and modify the original data. Upon requesting any resource, the light weight multi user profile inference model performs the inference and verifies the user rights in updating the original data. The modification will be performed only if the inference model returns a positive result to the verification process. The method performs block based encryption and the data management is performed in sequential manner. The modification of any block upon successful verification will be performed based on the result of inference model. The shared data can be updated by different users but the level of access is restricted by the profile based approach. The proposed method improves the performance of public auditing and data management.

Key words: User Profile Inference Model · Public Auditing · Data Sharing · Data Management · Cloud Computing

INTRODUCTION

The cloud environment can be visualized as a collection of resources in different layers like Platform, Data, Network and presentation. To access the resources available in different layers the cloud service providers (CSP) provides variety of services like Software as a service (SaaS), platform as a service (PaaS) and so on. The entry of cloud computing solves the problem of maintaining valuable resources for any organizations and it makes possible for the data owner to share the resource between many users. So that the cloud users can access the same content at any point of time without fail.

The data located in the cloud could be shared between large numbers of users to provide Data Sharing. By sharing the data between different users of the network, the cloud user can share data between them and could work on the same copy of the cloud resource. For example, in a collaborative working environment, the organization units works on the same resource and whatever the reflection made in the resource has to reply on the copy of others. This supports the distributed computing in the organizations and supports faster development or rapid working strategy.

The problem of data sharing in cloud environment is, the users of the environment does not knew to each other. In such loosely coupled environment, ensuring the exact vision of data to the user is must. In cloud the user has to come to an idea or mentality that he is viewing the same content or using the same content what the others are viewing and he is using the correct data. Providing such trustworthy to the cloud user can be named as public auditing. Providing public auditing in cloud environment has been studied in numerous approaches, but struggles with the accuracy in public auditing. Also in order to provide public auditing the identity of the cloud user has to be verified before modifying the original content.

In a shared environment, the data can be shared between different users. When the original data can be modified, the data has to be verified for the correctness.

Corresponding Author: A. Kalaivani, MCA, Nehru Institute of Information Technology & Management, Coimbatore, India.

The data owner can specify restriction on different users. For certain kind of users, the data owner can specify read permissions and for the other case the data owner can specify write permissions. By maintaining such user profiles for different users, the access restrictions can be enhanced.

Related Works: There are number of methods has been discussed for the development of public auditing in cloud environment. This section specifies some of the methods discussed earlier for the problem of public auditing in cloud.

Data storage auditing service in cloud computing: challenges, methods and opportunities [1], investigate this kind of problem and give an extensive survey of storage auditing methods in the literature. First, we give a set of requirements of the auditing protocol for data storage in cloud computing. Then, we introduce some existing auditing schemes and analyze them in terms of security and performance. Finally, some challenging issues are introduced in the design of efficient auditing protocol for data storage in cloud computing [2-4].

Identity Privacy-Preserving Public Auditing with Dynamic Group for Secure Mobile Cloud Storage [5], proposes a dynamic group key agreement is employed for key sharing among mobile users group and the idea of proxy re-signatures is borrowed to update tags efficiently when users in the group vary. In addition, the third party auditor (TPA) is able to verify the correctness of cloud data without the knowledge of mobile users' identities during the data auditing process. We also analyze the security of the proposed protocol.

Securing Public Data Storage in Cloud Environment, ICT and Critical Infrastructure [6-14], considers the lack of physical access to servers constitutes a completely new and disruptive challenge for investigators. The Users are store, transfer or exchange their data using public cloud. This paper represents the encryption method for public cloud and also the cloud service provider's verification mechanism using the third party auditors.

Privacy-Preserving Public Auditing for Secure Cloud Storage [15-17], discusses the public auditability for cloud storage is of critical importance so that users can resort to a third party auditor (TPA) to check the integrity of outsourced data and be worry-free. To securely introduce an effective TPA, the auditing process should bring in no new vulnerabilities towards user data privacy and introduce no additional online burden to user. In this paper, we propose a secure cloud storage system supporting privacy-preserving public auditing.

Securing personal health records in cloud computing: patient-centric and fine-grained data access control in multi-owner settings [2], propose a novel framework for access control to PHRs within cloud computing environment. To enable fine-grained and scalable access control for PHRs, we leverage attribute based encryption (ABE) techniques to encrypt each patient's PHR data. To reduce the key distribution complexity, we divide the system into multiple security domains, where each domain manages only a subset of the users. In this way, each patient has full control over her own privacy and the key management complexity is reduced dramatically. Our proposed scheme is also flexible, in that it supports efficient and on-demand revocation of user access rights and break-glass access under emergency scenarios.

Privacy-preserving public auditing for data storage security in cloud computing [3], utilize and uniquely combine the public key based homomorphic authenticator with random masking to achieve the privacy-preserving public cloud data auditing system, which meets all above requirements. To support efficient handling of multiple auditing tasks, we further explore the technique of bilinear aggregate signature to extend our main result into a multi-user setting, where TPA can perform multiple auditing tasks simultaneously. Extensive security and performance analysis shows the proposed schemes are provably secure and highly efficient.

Enabling public auditability and data dynamics for storage security in cloud computing [4-9], studies the problem of ensuring the integrity of data storage in Cloud Computing. In particular, we consider the task of allowing a third party auditor (TPA), on behalf of the cloud client, to verify the integrity of the dynamic data stored in the cloud. The introduction of TPA eliminates the involvement of the client through the auditing of whether his data stored in the cloud are indeed intact, which can be important in achieving economies of scale for Cloud Computing. The support for data dynamics via the most general forms of data operation, such as block modification, insertion and deletion, is also a significant step toward practicality, since services in Cloud Computing are not limited to archive or backup data only. While prior works on ensuring remote data integrity often lacks the support of either public audit ability or dynamic data operations, this paper achieves both. We first identify the difficulties and potential security problems of direct extensions with fully dynamic data updates from prior works and then show how to construct an elegant verification scheme for the seamless integration of these two salient features in our protocol design.

NaEPASC [10-18], combine ID-based aggregate signature and public verification to construct the protocol of provable data integrity. With the proposed mechanism, the TPA not only verifies the integrity of outsourced data on behalf of cloud users, but also alleviates the burden of checking tasks with the help of users' identity. Compared to previous research, the proposed scheme greatly reduces the time of auditing a single task on the TPA side. Security analysis and performance evaluation results show the high efficiency and security of the proposed scheme.

A Framework for Secure Data Sharing over Cloud Based on Group Key Management [19], we are using new public key cryptography technique for provide security of data. This paper basically contains two concepts i.e. key generation, encryption and decryption of data. First one is the key generation we are using improved Diffe Hellman key exchange technique. The second one is advanced cryptography technique for data encryption and decryption. So that by proposing those techniques we can provide more secure, efficient and flexible of sharing data.

All the above discussed methods suffers with the problem of providing public auditing in efficient manner and increase the security of data storage in cloud environment.

Light Weight Multi User Profile Inference Model: The proposed light weight multi user profile inference model handles the user request and performs inference based on the user profile and the resource request being received. The method splits the resource into number of small scale block and generates key for each block using block based encryption scheme and modular padding scheme. The light weight method maintains user profile for various users and for each user there will be number of access permissions presented by the data owner. Based on the profile generated by the data owner, the CSP verifies the request and profile based on which the request will be handled. The entire process has been split into number of stages namely: Request Handler, Multi User Profile Inference, Block Based Encryption, Modular Padding. This section discusses about all the stages of the proposed inference model.

Request Handler: The request handler receives the user request and performs the complete coordination of the request cycle. The method receives the user request and verifies the user identity with the available user keys. Once the user identity has been verified, then the method performs multi user profile inference to make Updation in the original data. If the inference result is positive then the user request will be proceeded to modify the original data block, otherwise the modification request will be rejected.

```
Algorithm:
Input: User Request Req
Output: Boolean
Start
    Receive User Request Req.
    Identify the resource name R-name = Req.Res-Name.
    Identify the request type R-type = Req.Req-Type
    Receive User Key Ukey.
    Verify the presence of key Ukey.

    $Kflag = \int_{i=0}^{size(ks)} (Ks(i) == Ukey, 1, 0)$

    if K-flag==1
        Perform User Inference.
        if True Then
                Perform Block Verification.
                if true then
                perform modification.
            end
        End
    End
Stop.
```

The above discussed algorithm performs the complete coordination of the entire request/reply cycles of the cloud environment.

Block Based Encryption: The block based encryption performs the important role of the public auditing process in the cloud environment. The method computes the file size and generate a random integer to compute the number of blocks of the file. Based on the number of blocks the method splits the file into N number of blocks and for each block of the file, the method computes the total number of overflow bytes and total number of empty bytes. Based on the values of both overflow and empty counts, the method selects the key from the key set which will be used as the encryption key for the block.

```
Algorithm:
Input: Key Set Ks, Resource Res
Output: Block Set Bs
Start
    Read Key set Ks.
    compute file size Fs = $\Sigma$ $bytes \in Res$
    Generate random integer Rint= Rand(Fs).
    Split file into Rint size of blocks.
    Block Set Bs = $\int_{i=1}^{size(Rint)} Split(\mathrm{Re}s,i)$ for each block Bi from Bs
        compute overflow bytes Ofb = $\int_{i=1}^{size(Bi)} \Sigma bytes(Bi) \leftrightarrow 32$
        Compute empty bytesEb = $\int_{i=1}^{size(Bi)} \Sigma bytes(Bi) < 32$
        compute difference Bd = Ofb-Eb
        Select Encryption key Ekey = Ks(Bd).
        Perform Encryption.
        Perform Modular Padding (Bd, Ofb, Eb).
    End
Stop.
```

The above discussed method performs the block based encryption of the blocks of the file in cloud.

Modular Padding: The modular padding is the process of adding K number of empty bits of zeros to the end of a block which is encrypted. The method is given with encrypted block and the computed overflow, empty bytes of the block. Using the values of overflow and empty number of bytes, the method computes the modulus value and the method appends the computed number of modulus bits to the end. The padded data will be given to the block based encryption algorithm. On the other side, the user can perform the same process and identify the padded details and remove them before decrypting the data.

```
Algorithm:
Input: Encrypted Block Eb, Overflow Bytes Ofb, Empty Bytes Eb
Output: Eb
Start
    Read Encrypted Block Eb.
    compute modulus value of Ofb and Eb.
    X = Mod(Ofb, Eb).
    Padd X number of zeros to Eb.
    returnEb.
Stop.
```

The above discussed algorithm computes the modulus values of Ofb and Eb computed and using the value of modulus the method adds the number of zeros in the end of the encrypted block.

User Profile Inference: The inference algorithm performs the verification of the user request. The method receives the user request and resource id and the type of request. Using all these results the method verifies the user profile about the authorization of the user for the particular resource and also the method reads the trace of access belongs to the user and identifies whether the user has any malformed action on accessing the resources. Based on both the values the method computes the profile trust weight which represents the trustworthy of the user. If the trust weight is more than specific threshold then the user will be allowed to perform modification.

```
Algorithm:
Input: User Request Req.
Output: Null.
Start
        Read user profile UP.
        Read User Trace Ut.
        Initialize Flag.
        Identify resource id RID = Req.RID
        Perform matching in profile.
        for each profile Pi from UP
if $\int_{i=1}^{size(Up)} UP(i), RID == RID \& \&UP(i).User == User \& \&Up(i).\mathrm{Re}q.Type == \mathrm{Re}q.Type$ then
        flag=true.
        End
    End
    If Flag = = True Then
        Compute Total Number of access Ta = $\int_{i=1}^{size(Up)} \Sigma Up(i).User == User$
        Compute Profile Trust weight PTW = $\frac{Ta}{size(Up)} \times 100$
        if PTW>TTh then //Trust Threshold
        Allow Modification.
    Else
        Ignore Request
        End
    End
Stop.
```

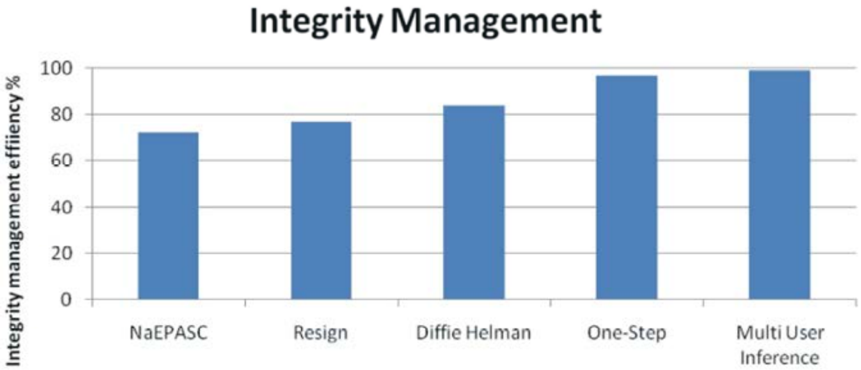
The above algorithm performs the inference of user request and computes the trust weight for each request. Based on the values of trust weight, the method allows or denies the user request.

RESULTS AND DISCUSSION

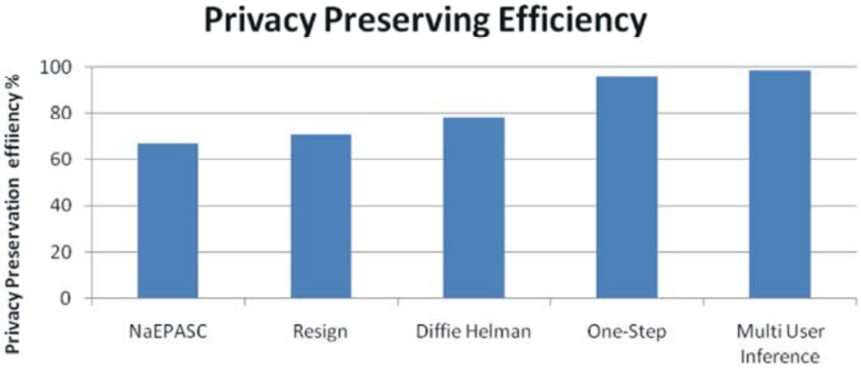
The proposed multi user profile inference model based public auditing has been implemented using cloud simulator and has been evaluated for its efficiency. The method has been tested with different number of scenarios and different simulation environment. The method has produced efficient results in all the scenarios and has been listed below.

Table 1: Details of simulation parameter

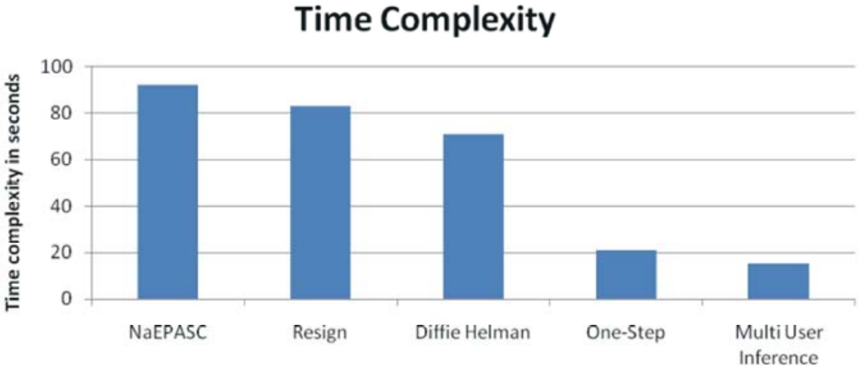
Simulation Parameter	Value
Simulator Name	Cloud Sim
Number of Services	20
Total Number of users	200
Trace Period	2 months



Graph 1: Comparison of integrity management



Graph 2: Comparison of privacy preservation efficiency



Graph 3: Comparison of time complexity

The Table 1, shows the details of simulation parameter being used to evaluate the performance of the proposed method.

The Graph 1 shows the comparison of integrity management efficiency produced by different methods and the proposed method has produced efficient results than other approaches.

The Graph 2, shows the comparison of privacy preservation efficiency and it shows clearly that the proposed method has produced more privacy preservation.

The Graph 3, shows the comparison of time complexity of different methods and it shows clearly that the proposed method has produced less time complexity than other methods.

CONCLUSION

In this paper, we proposed an light weight multi user inference model for the public auditing and data management in cloud environment. The method performs the block based encryption which splits the file into number of small blocks and the method has used modular padding scheme which increases the security of data sharing. The method verifies the user identity using key verification and the user request is validated using the multi user inference model. The method computes the profile trust weight for the user request and based on the trust weight the method allow or deny the modification request. The method has produced efficient results in securing the cloud services and enhances the quality of public auditing and data management. Also the method improves the efficiency of data sharing in cloud environment.

REFERENCES

1. Yang, Kan and Xiaohua Jia, 2012. Data storage auditing service in cloud computing: challenges, methods and opportunities, Springer, World Wide Web, 15(4): 409-428.
2. Li, M., S. Yu, K. Ren and W. Lou, 2010. Securing personal health records in cloud computing: patient-centric and fine-grained data access control in multi-owner settings. In: Security and Privacy in Communication Networks, pp: 89-106.
3. Wang, C., Q. Wang, K. Ren and W. Lou, 2010. Privacy-preserving public auditing for data storage security in cloud computing. In: Proceedings of the 29th Conference on Information Communications, INFOCOM'10, IEEE Press, Piscataway, NJ, USA, pp: 525-533.
4. Yu, S., C. Wang, K. Ren and W. Lou, 2010. Achieving secure, scalable and fine-grained data access control in cloud computing. In: Proceedings of the 29th Conference on Information Communications, IEEE Press, pp: 534-542.
5. Yong Yu, Yi Mu, Jianbing Ni, Jiang Deng and Ke Huang, 2014. Identity Privacy-Preserving Public Auditing with Dynamic Group for Secure Mobile Cloud Storage, Springer, Network and System Security Lecture Notes in Computer Science, 8792: 28-40.
6. Canepa, H. and D. Lee, 2010. A virtual cloud computing provider for mobile devices I. In: Proceeding of 1st ACM Workshop on Mobile Cloud Computing and Services Social Networks and Beyond (MCS 2010), ACM Digital Library, San Francisco, pp: 6.
7. Huang, D., T. Xing and H. Wu, 2013. Mobile cloud computing service models: a user-centric approach. IEEE Network, 27(5): 6-11.
8. Dinh, H.T., C. Lee, D. Niyato and P. Wang, 2013. A survey of mobile cloud computing: architecture, applications and approaches. Wireless Communication and Mobile Computing, 13(8): 1587-1611.
9. Wang, Q., C. Wang, K. Ren, W. Lou and J. Li, 2012. Enabling public auditability and data dynamics for storage security in cloud computing. IEEE Transactions on Parallel Distribted Systems, 22(5): 847-859.
10. Zhu, Y., H. Hu, G.J. Ahn and M. Yu, 2012. Cooperative provable data possession for integrity verification in multicloud storage, IEEE Transactions on Parallel Distribted Systems, 23(12): 2231-2244.
11. Yang, K. and X. Jia, 2013. An efficient and secure dynamic auditing protocol for data storage in cloud computing, IEEE Transactions on Parallel Distribed Systems, 24(9): 1717-1726.
12. Wang, C., S.S.M. Chow, Q. Wang, K. Ren and W. Lou, 2013. Privacy-preserving public auditing for secure cloud storage. IEEE Transactions on Computers, 62(2): 362-375.
13. Wang, B., B. Li and H. Li, 2013. Public auditing for shared data with efficient user revocation in the cloud. In: Proceeding of IEEE Conference on Computer Communications (IEEE INFOCOM 2013), Turin, Italy, pp: 2904-2912.
14. Boopathy, D. and M. Sundaresan, 2014. Securing Public Data Storage in Cloud Environment, ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society of India- Vol I Advances in Intelligent Systems and Computing, 248: 555-562.
15. Tripathi, A. and P. Yadav, 2012. Enhancing Security of Cloud Computing using Elliptic Curve Cryptography, International Journal of Computer Applications (0975 - 8887), 57(1): 26-30.
16. Mishra, A., D.K. Gupta and G. Sahoo, 2013. BIT Mesra Ranchi, Jharkhand. The Secure Data Storage in Cloud Computing Using Hadamard Matrix. International Journal of Engineering Science and Innovative Technology (IJESIT), 2(2): 389-395.

17. Chourasiya Prof. N.L., Dayanand Lature, Arun Kumavat, Vipul Kalaskar and Sanket Thaware, 2015. Privacy-Preserving Public Auditing for Secure Cloud Storage, International Journal of Engineering Research and General Science, 3(2).
18. Tan, Shuang and Yan Jia, 2014. NaEPASC: a novel and efficient public auditing scheme for cloud data, Journal of University SCIENCE, 15(9): 794-804.
19. Umamaheswari Ginjupalli and Behara Vineela, 2014. A Framework for Secure Data Sharing over Cloud Based on Group Key Management, International Journal of Engineering Trends and Technology (IJETT), 17(6): 276-279.