

A Light Weight Single Parametric Approach Based Secure Computing For Dynamic Data Management In Multi Clouds With Public Auditing

V Joseph Michael Jerard

Research scholar, Karpagam University, Coimbatore, Tamil Nadu, India.

Dr. Nedunchezian M.E. Ph.D.,

Principal, Sri Ranganathar Institute of Engineering and Technology, Athipalayam, Coimbatore, Tamil Nadu, India

Abstract

The loosely coupled nature of cloud environment is more prone for various threats while providing dynamic access to many users. The cloud users are able to access the data available in the cloud through set of services and the data modified has to be reflected to the others at the next visit. The cloud has to provide a view that everyone views the same copy of information called public auditability. There are many approaches has been discussed earlier to provide dynamic data management and to maintain the public auditability, but suffers with the problem of verification overhead and time complexity. To overcome the above said problem, we propose a novel light weight single parametric approach to perform secure computing in cloud environment. The cloud server generates a group key and private key for the client using a single parametric hashing function which is given to the user. The user will be allowed to perform modification based on private key and group key. Unlike block based or attribute based encryption the proposed method verifies the user identity using a light weight hashing function. We introduce a dynamic paging technique which performs data management. The proposed approach reduces the overhead of verification and reduces the time complexity also.

Key Words: Public Auditability, Data Management, Secure Computing, Multi Cloud, Dynamic Paging.

Introduction

The growth of information needs more space, which cost more and which is not feasible for the small scale organizations and individual users. The entry of cloud environment has provided the way for them to store and retrieve their information with little payment. The cloud service provider provides set of services for the users

to store their information and retrieve them on necessary. The data stored may be personal information and other financial information which has to be preserved to provide privacy for the cloud users. So that in order to access the information stored in the cloud the user will be assigned with various security measures and the user can be access the data through the service. The user identity will be verified by the third party auditor and the TPA stores various security information of users. Upon receiving a request the cloud manager handover the request to the TPA and the TPA verifies the user identity and trustworthy. Once the user clears the trust test then his request will be processed to complete the request.

The key generator generates public and private keys for each users and distribute them to the third party auditor and the users. Whenever a user request a service to the service provider, the third party auditor performs verification of user identity. Based on the result of TPA the user request will be processed by the service provider.

In multi cloud environment, the data are stored in multiple clouds and same copy will be available in different clouds. The user will be allowed to access the data stored in multiple clouds and able to modify them. The modified data will be reflected to the copies of the data also. Public auditability is the process of providing assurance that every user accesses the correct copy of data. The public auditability shows the security enforcement available in the cloud environment.

The data management is also an important factor of secure cloud computing where the modification performed by a single user of the cloud has to reflect at all the users copy. In an collaborative environment the users of the organization works together in the cloud, and the modification performed by any user has to reflect on the copies of others. The cloud environment has to provide dynamic data management so that the cloud could provide recent data to the users.

Dynamic paging is the process of modifying the address of the block to a new one where the recent one is available. Each block in the page has reference to the next block of the page and has to be updated in efficient manner, so that whatever the modification performed will be reflected easily and soon.

Also for the verification of user identity the third party auditor performs various steps using number of secret keys. The construction and verification process requires more amount of time and costs more. Also this introduces more latency in service completion which has to be reduced. To consider this a light weight verification process has to be introduced and should be done in shorter time.

Related Works

To provide security and data public auditability, data management there are many approaches has been discussed earlier. We discuss few of them in this chapter.

Metadata Partitioning for Large-scale Distributed Storage Systems [2], a dynamic programming method combined with binary search to solve the partitioning problem. With theoretical analysis and extensive experiments, we show that our algorithm finds the partitioning that minimizes load imbalance among servers and maximize efficiency of metadata operations.

With the advances in cloud computing technology it is now possible to store a huge number of images and raw data throughout the world. In order to access these distributed data with a reduced latency, this paper describes into the dynamic metadata model in cloud computing database. When designing a metadata, the storage location of metadata and the attributes inside the metadata is of importance for the efficient retrieval of data. They propose a new semantic metadata modeling [3] to reduce the overhead problem while retrieving the data from the data server. With theoretical analysis and experiments we show that our metadata modeling minimizes the latency time for fetching the data by reducing the search time to get the appropriate data.

Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing [6], 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 auditability 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. In particular, to achieve efficient data dynamics, we improve the existing proof of storage models by manipulating the classic Merkle Hash Tree construction for block tag authentication. To support efficient handling of multiple auditing tasks, we further explore the technique of bilinear aggregate signature to extend our main result into a multiuser setting, where TPA can perform multiple auditing tasks simultaneously. Extensive security and performance analysis show that the proposed schemes are highly efficient and provably secure.

Efficient Data Storage and Security in Cloud [7], propose an efficient and secure protocol to address these issues. Our design is based on Elliptic Curve Cryptography and Sobol Sequence (random sampling). Our method allows third party auditor to periodically verify the data integrity stored at CSP without retrieving original data. It generates probabilistic proofs of integrity by challenging random sets of blocks from the server, which drastically reduces the communication and I/O costs. The challenge-response protocol transmits a small, constant amount of data, which minimizes network communication. Most importantly, our protocol is confidential: it never reveals the data contents to the malicious parties. The proposed scheme also considers the dynamic data operations at block level while maintaining the same security assurance. Our solution removes the burden of verification from the user, alleviates both the user's and storage service's fear about data leakage and data corruptions.

Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing [10], ng 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 auditability 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. In particular, to achieve efficient data dynamics, we improve the existing proof of storage models by manipulating the classic Merkle Hash Tree construction for block tag authentication. To support efficient handling of multiple auditing tasks, we further explore the technique of bilinear aggregate signature to extend our main result into a multiuser setting, where TPA can perform multiple auditing tasks simultaneously

Privacy-Preserving Public Auditing for Secure Cloud Storage [11], propose a secure cloud storage system supporting privacy-preserving public auditing. We further extend our result to enable the TPA to perform audits for multiple users simultaneously and efficiently. Extensive security and performance analysis show the proposed schemes are provably secure and highly efficient. Our preliminary experiment conducted on Amazon EC2 instance further demonstrates the fast performance of the design.

Data Security in Cloud for Health Care Applications [12], describes into the benefit of cloud computing for healthcare organizations and examines the availability and security considerations that healthcare data requires. The proposed model strengthens the availability and security of data using metadata. The metadata created based on DCMI standards provides easy access of data by locating the server and secure the data resting in the cloud. In this paper security is enforced by cipher key which is generated from the attributes of metadata by providing two novel features. 1. Security is provided, where the encryption and decryption keys cannot be compromised without the involvement of data owner and health care organization, hence makes the data secured 2. The cipher key generated using modified feistel network increases the complexity of the key which strengthens the security effect.

Pseudonymization and Personal Metadata Encryption for Privacy-Preserving Searchable Document [13], presents a security protocol for data privacy that is strictly controlled by the data owner. Therefore, we integrate Pseudonymization and encryption techniques to create a methodology that uses pseudonyms as access control mechanism, protects secret cryptographic keys by a layer-based security model, and provides privacy preserving querying.

Secure Overlay Cloud Storage with Access Control and Assured Deletion. [14], costs. However, we must provide security guarantees for the outsourced data, which is

now maintained by third parties. We design and implement FADE, a secure overlay cloud storage system that achieves fine-grained, policy-based access control and file assured deletion. It associates outsourced files with file access policies, and assuredly deletes files to make them unrecoverable to anyone upon revocations of file access policies. To achieve such security goals, FADE is built upon a set of cryptographic key operations that are self-maintained by a quorum of key managers that are independent of third-party clouds. In particular, FADE acts as an overlay system that works seamlessly atop today's cloud storage services. We implement a proof-of-concept prototype of FADE atop Amazon S3, one of today's cloud storage services. We conduct extensive empirical studies, and demonstrate that FADE provides security protection for outsourced data, while introducing only minimal performance and monetary cost overhead. Our work provides insights of how to incorporate value-added security features into today's cloud storage services.

All the above discussed approaches has the problem of providing efficient public auditability and data management in a secure manner. Also the methods has more overhead in verification process and time complexity.

Proposed Method

The proposed single parametric approaches generate a group key and private key for the user and distribute them to the third party auditor and the users. The key generation is performed using single parametric hashing function and verification is performed using one step verification process. The dynamic data management is performed using dynamic paging technique. We discuss each step of the proposed approach in detail here in this chapter.

Single Parametric Hashing

At the first phase the key generator generates the group key and generates the private key for each user registered. The private key is generated using the details like client id, cloud id, service id and file id. The private key is generated with the size of 8 bit and for each detail fo private key a 2 bit slot is used. The generated private key and the group key is given to the user as well as the third party auditor. The key given to the third party auditor will be used to very the user identify at the time of request phase.

Algorithm:

Input: Client id Clid, Cloud id CID, service id Sid, File Id Fid.

Output: group key Gk, Private key Pk.

Step1: Generate group key $Gk = \int Cid \cup \text{Number of file in the group}$

Step2: initialize Pk=null

Step3: Private key $Pk = \int Clid \cup Cid \cup Fid \cup Sid$

Step4: stop.

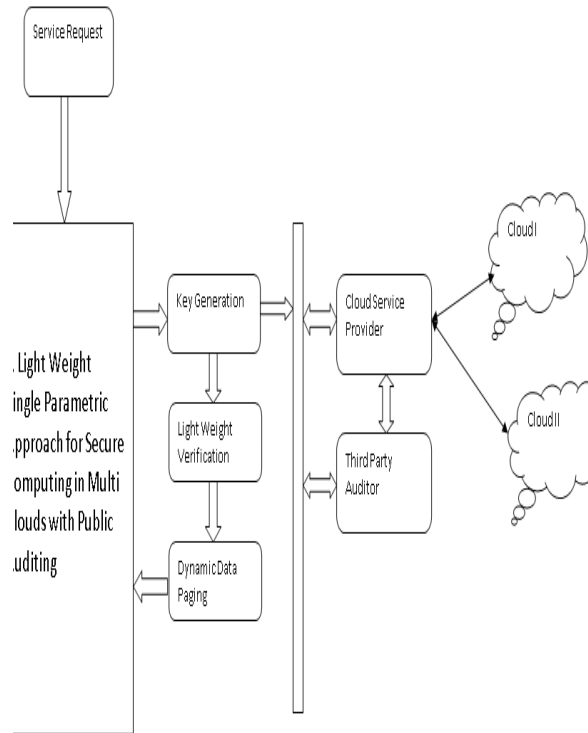


Figure 1: Proposed System Architecture

The figure 1 shows the architecture of the proposed light weight single parametric approach. The architecture has number of functional blocks namely Key Generation, Verification and Dynamic paging. Each is discussed in detail in this chapter.

Dynamic Data Paging

The dynamic paging is performed whenever there is a modification performed in the original copy of the source or the block of data has been modified. The user is allowed to perform modification only when the user clears the trust verification. The method maintains the files in sequence of blocks and maintains a file paging where it stores information about the address of the file block. Upon receiving any request for modification the user will be provided with the copy of the block and the modified content will be stored in a new block. The new block address is indexed in the paging details. The new block address is added in the paging details which enables the other users to get the result of modification and provides the public auditability.

Algorithm:

Input: File Set F_s , Modified Block M_b

Output: Modified File Set F_s .

Step1: for each file F_i from F_s

 For each file index from file paging fp

 If $\int_{i=1}^{size(F_s)} Fp(F_s \in M_b), 1,0$ then

 Modify the reference in file paging $Fp(F_s(b)) = Addr(M_b)$.

end

Step 2: stop. Stop.

One Step Verification:

The one step verification is the process of verifying the user identity in easy way and is performed using the encryption key given by the key generator and the key received from the user. The third party auditor has the list of private keys allocated to the users and each user has their own private key and group key. While submission of the client request the user private key is obtained and the signature verification is performed based on the other parameters also. The third party auditor has been provided with the encryption key , client id's allocated, Hash integer and so on.

Algorithm:

Input: Client Id Clid, Cloud ID CID, Service ID sid, File ID Fid, Private key pk

Output: Boolean

Step1: Read private key pk

Step2: Convert pk into bit stream

$$Bs = \int_{i=1}^4 \sum_{j=k}^{j+2} pk(j, j + 2)$$

Step3: verify Client ID

If $\int_{i=1}^2 Bs(i) == Bit(Clid)$ then

If $\int_{i=2}^4 Bs(i, j) == Bit(Cid)$ then

If $\int_{i=4}^6 Bs(i, j) == Bit(sid)$ then

If $\int_{i=6}^8 Bs(i, j) == Bit(Fid)$ then Return true

Else

Return false.

End

Else

Return false

End

Else

Return false

End

Else

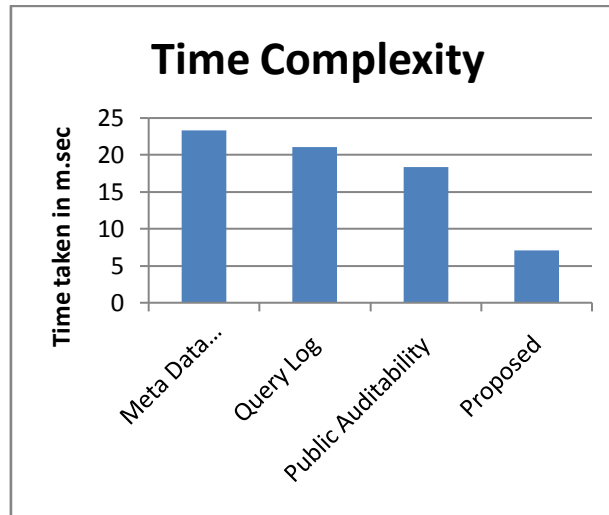
Return false

End

Step 4: stop.

Results and Discussion

The proposed light weight single parametric approach has been implemented with various number of services and varying number of users. The method has produced efficient results in all the circumstances. The key generation process has been effective and the TPA take very less time to perform the verification process.



Graph 1: Comparison of Verification Time Complexity

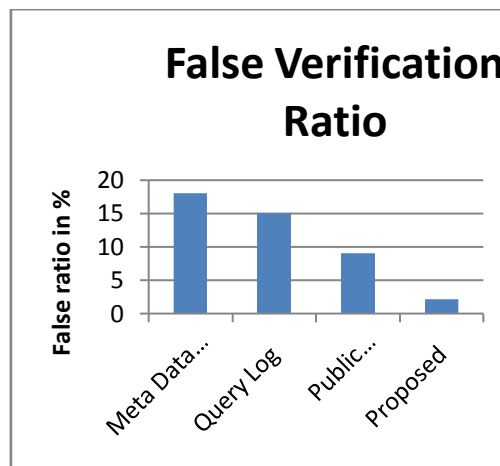
The graph1 shows the comparison of time taken by various methods for performing verification process. It shows that the proposed method has produced less time complexity for verification than other approaches.

The false positive ratio specifies the number of fake user keys verified as genuine from number of keys given.

$$\text{False positive ratio FPR} = \frac{\text{Number of false verification}}{\text{Total number of verification}} \times 100$$

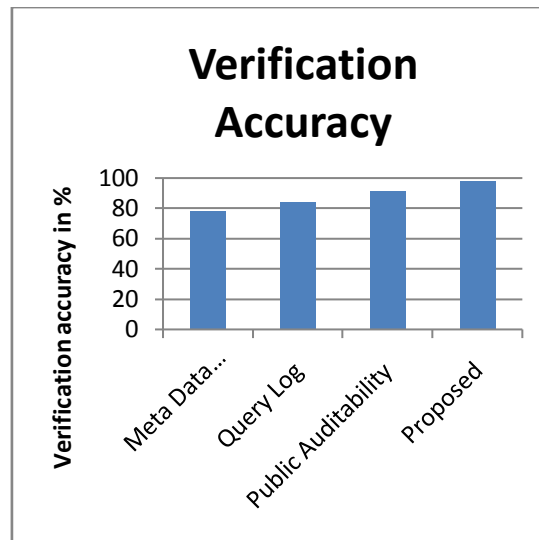
Accuracy of the method is computed based on the number of keys verified correctly from set of given keys.

$$\text{Accuracy acc} = \frac{\text{Number of keys verified correctly}}{\text{Total number of keys}} \times 100$$



Graph 2: Comparison of False Verification Ratio

The graph2 shows the comparison of false verification ratio produced by various methods. It shows clearly that the proposed method has produced less false verification ratio than other methods.



Graph 3: Comparison of Verification Accuracy

The graph 3, shows the comparison of verification accuracy produced by various methods. It shows clearly that the proposed method has produced higher accuracy in verification.

Conclusion

We proposed a light weight single parametric approach for secure cloud computing with public auditability. The method generates a private key for the user, and distribute them to the client and third party auditor. Based on the private key the user identify is maintained and the data modification is maintained using dynamic paging technique. The dynamic paging technique produces more efficient throughput ratio and reduces the overall time complexity of the system. The proposed method produces efficient results in data dynamic and public auditability.

References

- [1] R.Anitha, Data Security in Cloud for Health Care Applications, Springer, Advances in Computer Science and its Applications Lecture Notes in Electrical Engineering Volume 279, 2014, pp 1201-1209
- [2] Wu, J.-J., Liu, P., Chung, Y.-C.: Metadata Partitioning for Large-scale Distributed Storage Systems. In: IEEE International Conference on Cloud Computing (2010)

- [3] Anitha, R., Mukherjee, S.: A Dynamic Metadata Model in Cloud Computing. In: Krishna, P.V., Babu, M.R., Ariwa, E. (eds.) ObCom 2011, Part II. CCIS, vol. 270, pp. 13–21. Springer, Heidelberg (2012)
- [4] Mathew, A.: Survey Paper on Security & Privacy Issues in Cloud Storage Systems. In: Electrical Engineering Seminar and Special Problems 571B (2012)
- [5] Tang, Y., Lee, P.P.C., Lui, J.C.S., Perlman, R.: Secure Overlay Cloud Storage with Access Control and Assured Deletion. *IEEE Transactions on Dependable and Secure Computing* 9(6) (2012)
- [6] Wang, Q., Wang, C., Ren, K., Lou, W., Li, J.: Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing. *IEEE Transactions on Parallel and Distributed Systems* 22(5) (2011)
- [7] Ravi Kumar, J., Revati, M.: Efficient Data Storage and Security in Cloud. *International Journal of Emerging trends In Engineering and Development* 6(2) (2012)
- [8] Modi, C., Patel, D., Borisaniya, B., Patel, A., Rajarajan, M.: A survey on security issues and solutions at different layers of Cloud computing. *Journal of SuperComputers*, 561–592 (2013)
- [9] Arai, Y., Watanabe, C.: Query Log Perturbation Method for Privacy Preserving Query. In: 4th International Conference on Ubiquitous Information Management and Communication (2010)
- [10] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing," Proc. 14th European Symp. Research in Computer Security (ESORICS '09), pp. 355-370, 2009.
- [11] Cong Wang, Privacy-Preserving Public Auditing for Secure Cloud Storage, *IEEE Conference on Computer communication, INFOCOM*, 2010.
- [12] R. Anitha, Saswati Mukherjee, *Data Security in Cloud for Health Care Applications*, Springer, *Advances in computer science and applications*, vol. 279, pp. 1201-1209, 2014.
- [13] Heurix, J., Karlinger, M., Neubauer, T.: Perimeter – Pseudonymization and Personal Metadata Encryption for Privacy-Preserving Searchable Documents. In: *International Conference on Health Systems*, vol. 1(1), pp. 46–57 (2012)
- [14] Tang, Y., Lee, P.P.C., Lui, J.C.S., Perlman, R.: Secure Overlay Cloud Storage with Access Control and Assured Deletion. *IEEE Transactions on Dependable and Secure Computing* 9(6) (2012)
- [15] R. Geambasu, T. Kohno, A. Levy, and H.M. Levy, "Vanish: Increasing Data Privacy with Self-Destructing Data," Proc. 18th Conf. USENIX Security Symp, Aug. 2009.