Efficient and Secure Auditing of Cloud Data with Key -Updating

Niranjana S, Manjusha M S

[1]M.Dasan Institute of Technology, Calicut University, Kerala, India

[2]Assistant Professor, Department of Computer Science and Engineering, M. DIT,Ulliyeri, Kozhikode, Kerala, India

Abstract: ***Cloud storage auditing is the process of verifying the integrity of the data stored in public cloud. The existing auditing protocols are based on the assumption that the client's secret key for auditing is secure. In practical the secret key is not absolutely secure due to weak security settings at the client. If secret key for auditing is exposed, most of the current auditing protocols become unable to work. This paper focus on how to solve the problem of key-exposure in public cloud and give the practical solution for it. In the proposed design, use a binary tree structure and the preorder traversal technique to update the client's secret keys. Also develop a novel authenticator construction for forward security and the property of blockless verifiability. The secret key updating is done by a trusted authority and the updating process is done automatically. If the verification result is negative then it is possible to retrieve the original data.***

Keywords: cloud storage auditing, homomorphic linear authenticator, key-exposure resistance

1. Introduction

Cloud storage auditing is the process of checking integrity of data that stored in the cloud, which is one of the important security techniques in cloud. In recent years, auditing protocols for cloud storage auditing have attracted much attention and have been researched intensively. These protocols focus on several different aspects of auditing, and how to achieve high bandwidth, low communication cost and computation efficiency is one of the essential concerns. For that purpose, the Homomorphism Linear Authenticator (HLA) technique that supports block less verification is used to reduce the overheads of computation and communication in cloud storage auditing protocols, which allows the auditor to verify the integrity of the data in cloud without retrieving the whole data.

Most of the existing cloud storage auditing protocols are based on this technique. The privacy protection of the stored data is also an important aspect of cloud storage auditing. In order to reduce the computational burden of the client, a third-party auditor (TPA) is used. There is a chance for getting client's data to TPA after it executes the auditing protocol multiple times. Auditing protocols are designed to ensure the privacy of the client's data in cloud. Another aspect of cloud storage auditing protocol is to support data dynamic operations. Many research works about cloud storage auditing have been done in recent years, but the researches does not consider the security problem in client. All existing protocols focus on the faults or dishonesty of the cloud. Due to low security settings at the client there may be a chance for client's auditing key exposure.

The previous auditing protocols did not consider this critical issue, and any exposure of the client's secret auditing key it make most of the existing auditing protocols unable to work properly.

To reduce the damage of the client's key exposure in cloud storage auditing, design a cloud storage auditing protocol with built-in key-exposure resilience. Applying the traditional solution of key revocation to cloud storage auditing is not practical because, whenever the client's secret key for auditing is exposed, then the client needs to produce a new pair of public key and secret key for auditing and regenerate the authenticators for the client's data that are previously stored in cloud. That means download the whole data from the cloud, producing new authenticators, and re-uploading back to the cloud, these are more complicated processes.

The key updating is done by a trusted authority automatically at the end of each time period it automatically generate secret keys. If the verification shows error message there possible to retrieve the original data and upload into the cloud using new secret key and we can continue the verification process without any additional actions.

2. Literature Survey

The paper "Efficient Integrity Checking of Untrusted Network Storage" by A. Heitzmann, B. Palazzi, C.Papamanthou, R.Tamassia presents a general method and a practical prototype application for auditing the integrity of data that stored in an untrusted network storage service. The auditing process is managed by an application running in a trusted environment (client) that stores cryptographic hash value of constant size, corresponding to the digest of an authenticated data structure. Here use an effective probabilistic data structure skip list [1]. Verification is achieved through comparisons to a hash value stored by the client, the basis of the authenticated skip list on the authentication server. This hash is the only data which must be stored on the client, and it has constant size dependent only on the cryptographic hash function used. The architecture is both space-efficient (the user stores only a single hash value) and time efficient (a very small overhead is added to the operations of the storage service).

Licensed Under Creative Commons Attribution CC BY

In the provable data possession (PDP) model [2], the client first processes the data and then sends it to an untrusted server for storage and keep a small amount of metadata. Then the client can check the integrity of data without downloading the actual data. The original PDP scheme applies only in the case of static files. In “Dynamic Provable Data Possession” by C. Chris Erway, Alptekin Küpçü, C.Papamanthou, R. Tamassia present a framework and efficient constructions for dynamic provable data possession (DPDP), which extends the PDP model to support updates to stored data. Here use a new version of authenticated dictionaries based on rank information.

In “Toward publicly auditable secure cloud data storage services” by Q. C. Wang, K. Ren, W. Lou, and J. Li propose a new method that facilitate rapid deployment of cloud data storage auditing service and regain security assurances with cloud data dependability, efficient methods that enable on-demand data auditing on behalf of cloud data owners have to be designed. Use a trusted TPA for auditing purpose. Such an auditing service helps to save data owner’s computation resources and also provides a transparent yet cost-effective method for data owners to gain trust in the cloud. To reduce the large communication overhead for public auditability without introducing any burden on the data owner, use homomorphic authenticator technique [3]. Homomorphic authenticators are generate metadata from individual data blocks.

The cloud auditing system has many security challenges. “Enabling public auditability and data dynamics for storage security in cloud computing” by Q. Wang, C. Wang, K. Ren, W. Lou. This work studies the problem of ensuring the integrity of data storage in Cloud. To allowing a TPA, on behalf of the cloud and client, to verify the integrity of the dynamic data stored in the cloud. The use of TPA eliminates the involvement of the client through the auditing of whether his data stored in the cloud is 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. Prior works on ensuring remote data integrity often lacks the support of either public auditability or dynamic data operations, this proposed method achieves both.

To improve the existing auditing protocol models by manipulating the classic Merkle Hash Tree construction for block tag authentication. It also support efficient handling of multiple auditing tasks. The Merkle Hash Tree (MHT) is a authentication data structure [4], which is used to efficiently and securely prove that a set of elements are undamaged and unaltered. It is like a binary tree where the leaves in the MHT are the hashes of authentic data values. Public auditability also allows clients to delegate the auditing tasks to TPA while they themselves can be unreliable or not be able to commit necessary computation resources performing continuous verifications. Another major concern is how to construct verification protocols that can accommodate dynamic data files. In this paper, they explored the problem of providing simultaneous public auditability and data dynamics for remote data integrity check in Cloud Computing. The construction is designed to meet these two important goal.

Cloud has different type of customers and data owners and data servers have different identities and different business interests. Therefore, an independent auditing service is required to make sure that the data is correctly hosted in the

Cloud.” Data storage auditing service in cloud computing: Challenges, methods and opportunities” by K. Yang and X. Jia in this paper investigate this kind of problem and give an extensive survey of storage auditing. They introduce some challenging issues in the design of auditing protocols in cloud computing. Most of the previous auditing protocols (also denoted as Proof of Retrievability (POR) or Provable Data Possession (PDP)) are mainly designed for the static archive storage systems. The dynamic scalability is a significant issue in cloud storage auditing that means the data owner may need to update his data as: block modification, deletion and insertion. Another challenge is group collaboration. It is an important application in cloud computing because the benefit of group collaboration may be the main reason for an organization or a corporation deciding to subscribe a cloud service. Lin also proposed a data storage auditing protocol for the scenario of group collaboration based on the protocols proposed by Smart et al. [5] and Wang et al. [6]. Next one is batch auditing. It can save the communication bandwidth, because the Server just needs to send the linear combination of all the sampled data blocks whose size is equal to one data block, it can also reduce the computation complexity for auditing on both the third party auditor and the cloud server.

Most of the auditing protocol are based on the assumption that the TPA is trusted but due to some security issues in TPA there is a chance of “flowing away” client’s data towards external parties during the auditing process. In “Privacy preserving public auditing for secure cloud storage” by C. Wang, S. S. M. Chow, Q. Wang, K. Ren proposes a secure cloud storage system supporting privacy-preserving public auditing. It is independent to encryption. To achieve privacy-preserving public auditing, propose a new method that uniquely integrate the homomorphic linear authenticator with random masking technique. In this protocol, the linear combination of sampled blocks in the server’s response is masked with randomness generated by the server. With random masking technique, the TPA no longer has all the necessary information to build up a correct group of linear equations and therefore cannot derive the user’s data content by the TPA, no matter how many linear combinations of the same set of file blocks can be collected. The protocol can extend privacy-preserving public auditing protocol into a multi-user setting, where the TPA can perform multiple auditing tasks in a batch manner for better efficiency.

Audit service is constructed based on the techniques, fragment structure, random sampling, and index-hash table, supporting provable updates to outsourced data and timely anomaly detection. In addition, in “Dynamic audit services for outsourced storages in clouds” by Y. Zhu, G.-J. Ahn, H. Hu, S. S. Yau, H. G. An, and C.-J. Hu propose a method based on probabilistic query and periodic verification for improving the performance of audit services. The method is

Licensed Under Creative Commons Attribution CC BY

based on interactive proof system (IPS) with the zeroknowledge property, the audit service can provide public auditability without downloading whole data and protect privacy of the data. Also, our audit system can support dynamic data operations and timely anomaly detection by periodic sampling. They also propose an efficient approach based on probabilistic query and periodic verification for improving the performance of audit services.

The paper “An efficient and secure dynamic auditing protocol for data storage in cloud computing” by K. Yang and X. Jia mainly concentrated on the security issues arrived due to dynamic operations. The dynamic operations may make the auditing protocols insecure. Specifically, the server may cause two following attacks: 1) Replay Attack. The attack is due to the server may not update correctly the owner’s data on the server and may use the previous version of the data to pass the auditing. 2) Forge Attack. Is the problem due to the data owner updates the data to the current version, the server may get enough information from the dynamic operations to forge the data tag. If the server could forge the data tag, it can use any data and its forged data tag to pass the auditing. The proposed efficient and inherently secure dynamic auditing protocol, it protects the data privacy against the auditor by combining the cryptography methods with the bilinearity property, rather than using the mask technique[7]. Thus, the multi-cloud batch auditing protocol does not require any additional organizer. The batch auditing protocol can also support the batch auditing for multiple owners.

The existing remote data possession checking (RDPC) protocols have been designed in the PKI (public key infrastructure) setting. In Identity-Based Remote Data Possession Checking in Public Clouds the cloud server has to validate the users’ certificates before storing the data uploaded by the users in order to prevent spam. This incurs considerable costs since numerous users may frequently upload data to the cloud server. This paper addresses this problem with a new model of identity-based RDPC (ID-RDPC) protocols. We present the first ID-RDPC protocol proven to be secure assuming the hardness of the standard computational Diffie-Hellman (CDH) problem. In addition to the structural advantage of elimination of certificate management and verification, our ID-RDPC protocol also outperforms existing RDPC protocols in the PKI setting in terms of computation and communication. The ID-RDPC protocol is applicable in the case of small organization

The existing auditing schemes makes that the auditor executes high computation to check data integrity. It might become a burden for a lot of data owner. To solve the above problem, propose a novel public auditing scheme with public verifiability and constant communication cost based on self-certified signature scheme in Self-certified Public Auditing for Data Integrity in Cloud Storage” proposed by J. Zhang and W. Zeng. Thorough analysis shows that our proposed scheme is secure and efficient. The security of our scheme is based on the fixed inversion problem (FI) of the bilinear map and the inversion of hash function.

3. Problem Definition

Though many research works on cloud storage auditing have been done in recent years, a critical security problem exposure problem for cloud storage auditing, has remained unexplored in the previous researches. All existing protocols focus only on the faults or dishonesty of the cloud storage provider, they does not consider the weak sense of security and/or low security settings at the client.

Therefore, to deal with the client’s secret key exposure for cloud storage auditing is a very important problem. But in previous auditing protocols did not consider this critical issue, and any exposure of the client’s secret auditing key would make most of the existing auditing protocols unable to work correctly.

In this paper introduce a new method that focus on how to reduce the damage of the client’s key exposure in cloud storage auditing. The goal is to design a cloud storage auditing protocol with updating secret key in each time period. The updating of secret key is done automatically and also keep the original data in cloud securely.

4. System Model

The system model for the proposed auditing system is shown in Fig. 1. The system contains four parties: the client (data owner), TPA, trusted authority and the cloud. The client produces the data for storage and uploads these data along with corresponding authenticators to the cloud. The cloud stores these data for the client and provides view and download service if the client requires. Data is furthermore divided into multiple blocks [8]. The client or TPA plays the role of auditor in our system (for simplicity consider that the client perform the auditing task). The client can periodically perform auditing to check his data in cloud are correct. The lifetime of data stored in the cloud is divided into T + 1 time periods. In this model, the client will update his secret keys for auditing in the end of each time period. The cloud is allowed to get the client’s secret key for cloud storage auditing in one certain time period that means each time period the secret key is changed. The secret key updating task is done by trusted authority. It update the key at the end of each time period.

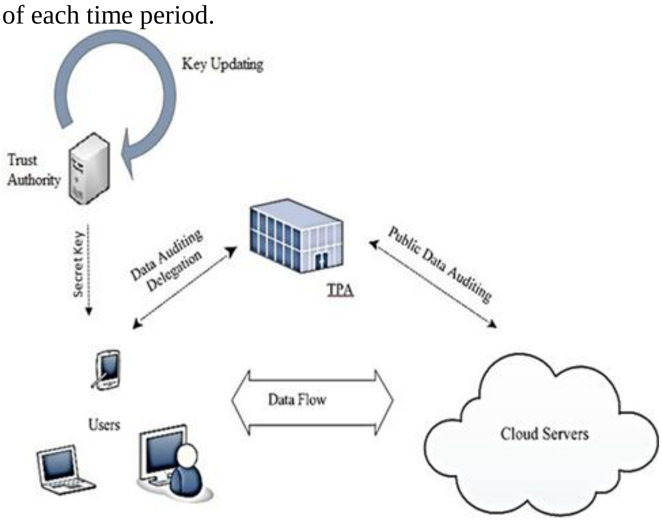

Figure 1: System model.

5. Secure Cloud Auditing with Key-Updating

The goal is to design an efficient practical auditing protocol with key-exposure resilience, in which the operational complexities like key size, computation overhead and communication overhead should be at most sub linear to T. In order to achieve the goal, use a binary tree structure to appoint time periods and associate periods with tree nodes by the pre-order traversal technique [9].

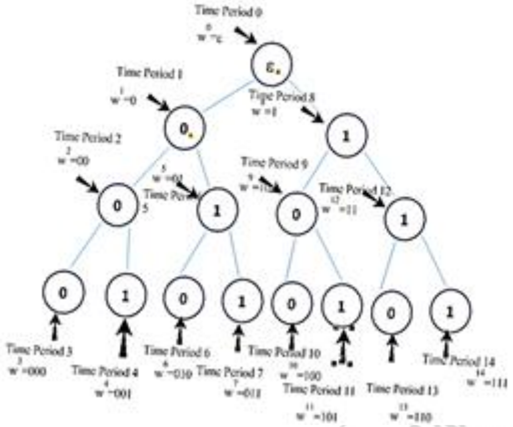

Figure 2: An example of how to associate the nodes with time periods in a binary tree with depth 4.

The secret key for client in each time period is organized as a stack. In each time period, the secret key is updated using forward-secure technique [10]. It guarantees that any authenticator generated in one time period cannot be computed from the secret keys for any other time period later than this authenticator. Computation overhead and communication overhead are only logarithmic in total number of time periods. The client can audit the integrity of the data stored in cloud by aggregated manner, i.e., without retrieving the entire data from the cloud storage system. The proposed protocol does not consider the key exposure resistance during one time period.

5.1 Algorithms for the proposed auditing protocol

The proposed auditing protocol consist of five algorithms:

- $SysSetup\ (1^k, T) \rightarrow (PK, SK_0)$: It is a probabilistic algorithm which takes as input a security parameter k and the total number of time periods T, the output is a public key PK and the initial client's secret key. This algorithm is run by the client side.
- $KeyUpdate(PK, j, SK_j) \rightarrow (SK_{j+1})$: The key update algorithm is a probabilistic algorithm. The input is the public key PK, the current period j and a client's secret key , and generates a new secret key for the next time period j + 1. This algorithm is run by the client side.
- $AuthGen(PK, j, SK_j, F) \rightarrow (\phi)$: The authenticator generation algorithm is a probabilistic algorithm which takes as input the public key PK, the current period j , the client's secret key and a file F, and generates the set of authenticators for F in the time period j. This algorithm is also run by the client side.
- $\Pr oofGen(PK, j, Chal, F, \phi) \rightarrow (P)$: This is a probabilistic algorithm which takes as input the public key PK, the time period j, a challenge Chal, a file F and the set of authenticators, and generates the proof P for F. This algorithm is run by the cloud storage system.
- $\Pr oofVerify(PK, j, Chal, P) \rightarrow ("True" \, or \, "False")$: the proof verifying algorithm it is a deterministic algorithm which takes as input the public key PK, a time period j , a challenge Chal and a proof P, and returns "True" or "False". This algorithm is run by the client side.
- *Restore* $(PK, j, SK_j', \phi) \rightarrow (\phi')$: Restore algorithm is used to retrieve the original data and stored in the cloud.

6. Results

In this system provide a secure cloud storage environment. Provide maximum secureness for the user's data. The evaluation includes security analysis and performance analysis. Security of the proposed scheme is done by analyzing storage correctness and privacy preserving property. Security analysis include storage correctness guarantee, privacy preserving guarantee, sorting out invalid responses. Storage correctness guarantee ensures that if cloud server passes the audit phase it must indeed possess the specified data intact as it is. The TPA cannot derive user's data from information during auditing; this is called as privacy preserving guarantee. The efficiency analysis on the batch auditing, is done by considering only the total number of pairing operations. However, on the practical side, there are additional less expensive operations required for batching, such as modular exponentiations and multiplications. Thus, whether the benefits of removing pairings significantly outweighs these additional operations remains to be verified.

6.1 Security Analysis

If any of the challenged data blocks m l or its data tag t l is corrupted or not up-to-date on the server, the server cannot pass the auditing because the verification equation cannot hold. The data privacy is an important requirement in the design of auditing protocol in cloud storage systems. The proposed auditing protocols are privacy-preserving. In the auditor side, it can only get the product of all the challenged data tags from the proof. The data proof in the auditing protocol is in an encrypted way by the exponentiate on the challenges. It is a discrete logarithm problem to get the linear combinations of the chosen data blocks.

6.2 Performance Analysis

Storage auditing is a very resource demanding service in terms of computational resource, communication cost and memory space. Here compare the communication cost and computation complexity between the proposed system with existing system.

In the proposed scheme, the key update workload is out-sourced to the TPA. In contrast, the client has to update the

www.ijsr.net ISSN (Online): 2319-7064

Licensed Under Creative Commons Attribution CC BY

secret key by itself in each time period in scheme. We compare the key update time on client side between the both schemes in Fig. 3. In scheme, the key update time on the client is related to the depth of the node corresponding to the current time period in binary tree. When the depth of node corresponding to the current time period is 0 or 1 (the node is internal node), the update time is about 12.6ms; when it is 2 (the node is leaf node), the update time is almost zero. In our scheme, the key update time on client side is zero in all time periods.

In our scheme, the communicational messages comprise the challenge message and the proof message. It is clear that the challenge message is linear with the size of blocks.

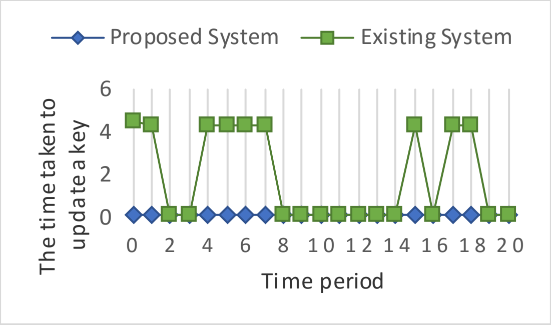

Figure 3: The key update time on client side in the proposed scheme and the existing scheme

7. Conclusion

The client's key exposure is a main problem in cloud storage auditing. Propose a new paradigm called auditing protocol with key-exposure resilience for resist the key-exposure. In such a protocol, the integrity of the data previously stored in cloud can still be verified even if the client's current secret key for cloud storage auditing is exposed.

The problem of client's secret key exposure is reduced by updating the secret key periodically. Also we can retrieve the original data even if the data is lost.

References

[1] W. Pugh. "Skip lists : a probabilistic alternative to balanced trees". Commun. ACM, 33(6):668–676, 1990.

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

[3] D. L. Gazzoni and P. S. L. M. Barreto, "Demonstrating data possession and uncheatable data transfe," Cryptology ePrint Archive, Report 2006/150, 2006.

[4] R. C. Merkle, "Protocols for Public Key Cryptosystems," Proc. IEEE Symp. Security Privacy, 1980.

[5] Smart, N.P., Warinschi, B., "Identity based group signatures from hierarchical identity-based encryption," In: Proceedings of the 3rd International Conference Palo Alto on Pairing-Based Cryptography, Pairing '09, pp. 150–170, 2009.

[6] Wang, Q., Wang, C., Li, J., Ren, K., Lou, W.: "Enabling public verifiability and data dynamics for storage security in cloud computing.,"In: Proceedings of the 14th European conference on Research in Computer Security, ESORICS'09, pp. 355–370.

[7] Y. Zhu, G.-J. Ahn, H. Hu, S. S. Yau, H. G. An, and C.-J. Hu, "Dynamic audit services for outsourced storages in clouds," IEEE Trans. Services Comput., vol. 6, no. 2, pp. 227–238, Apr./Jun. 2013.

[8] G. Ateniese, R. Di Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in Proc. 4th Int. Conf. Secur. Privacy Commun. Netw., 2008, Art. ID 9.

[9] R. Canetti, S. Halevi, and J. Katz, "A forward-secure public-key encryption scheme," in Advances in Cryptology—EUROCRYPT. Berlin, Germany: Springer-Verlag, 2003, pp. 255–271.

[10] J. Yu, R. Hao, F. Kong, X. Cheng, J. Fan, and Y. Chen, "Forwardsecure identity-based signature: Security notions and construction," Inf. Sci., vol. 181, no. 3, pp. 648–660, 2011.

[11] A.Heitzmann, B. Palazzi, C.Papamanthou, R. Tamassia., " Efficient Integrity Checking of Untrusted Network Storage," in Proc. 14th ACM Conf. Comput. Commun. Secur., 2008.

[12] C. Erway, A. Küpçü, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in Proc. 16th ACM Conf. Comput. Commun. Secur., 2009, pp. 213–222.

[13] Q. C. Wang, K. Ren, W. Lou, and J. Li, "Toward publicly auditable secure cloud data storage services," IEEE Netw., vol. 24, no. 4, pp. 19–24, Jul./Aug. 2010.

[14] K. Yang and X. Jia, "Data storage auditing service in cloud computing: Challenges, methods and opportunities," World Wide Web, vol. 15, no. 4, pp. 409–428, 2012.

[15] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy preserving public auditing for secure cloud storage," IEEE Trans. Comput., vol. 62, no. 2, pp. 362–375, Feb. 2013.

[16] K. Yang and X. Jia, "An efficient and secure dynamic auditing protocol for data storage in cloud computing," IEEE Trans. Parallel Distrib. Syst., vol. 24, no. 9, pp. 1717–1726, Sep. 2013.

[17] H. Wang, Q. Wu, B. Qin, and J. Domingo-Ferrer, "Identity-based remote data possession checking in public clouds," IET Inf. Secur., vol. 8, no. 2, pp. 114–121, Mar. 2014.

[18] J. Zhang and W. Zeng, "Self-certified Public Auditing for Data Integrity in Cloud Storage" Ninth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, Nov. 2014.

Author Profile

Niranjana S received the B. Tech degrees in Computer Science and Engineering from Kannur University in 2013 and now doing M. Tech in Computer Science and Engineering from Calicut University.

Licensed Under Creative Commons Attribution CC BY