ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Client's Key Exposure Resistance Method in Cloud Storage

K.Ashok
M.Tech Student
Dept. of CSE
Avanthi Institute of Engineering and Technology, Vizianagaram.

K.Ravindra
Associate Professor
Dept. of CSE
Avanthi Institute of Engineering and Technology, Vizianagaram.

ABSTRACT

Cloud storage auditing is viewed as an important service to verify the integrity of the data in public cloud. Current auditing protocols are all based on the assumption that the client's secret key for auditing is absolutely secure. However, such assumption may not always be held, due to the possibly weak sense of security and/or low security settings at the client. If such a secret key for auditing is exposed, most of the current auditing protocols would inevitably become unable to work. In this paper, we focus on this new aspect of cloud storage auditing. We investigate how to reduce the damage of the client's key exposure in cloud storage auditing, and give the first practical solution for this new problem setting. We formalize the definition and the security model of auditing protocolwith key-exposure resilience and propose such a protocol. In our design, we employ the binary tree structure and the preorder traversal technique to update the secret keys for the client. We also develop a novel authenticator construction to support the forward security and the property of blockless verifiability. The security proof and the performance analysis show that our proposed protocol is secure and efficient.

INTRODUCTION

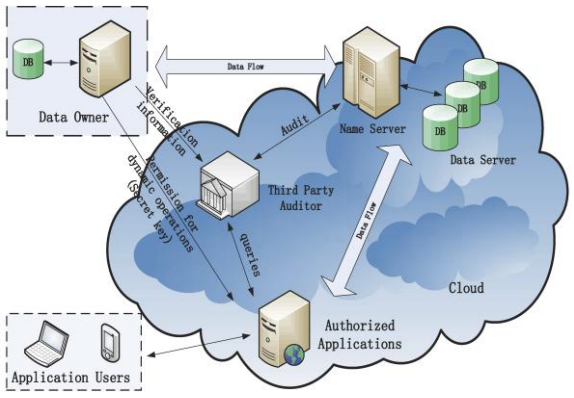
Cloud storage auditing is used to verify the integrityof the data stored in public cloud, which is one ofthe important security techniques in cloud storage. In recentyears, auditing protocols for cloud storage have attractedmuch attention and have been researched intensively these protocols focus on several different aspects of auditing,and how to achieve high bandwidth and computation efficiencyis one of the essential concerns. For that purpose, theHomomorphic Linear Authenticator (HLA) technique that supports blockless verification is explored to reduce the overheadsof computation and communication in auditing protocols,which allows the auditor to verify the integrity of the data incloud without retrieving the whole data. Many cloud storage auditing protocols like have been proposed basedon this technique. The privacy protection of data is alsoan important aspect of cloud storage auditing. In order toreduce the computational burden of the client, a third-partyauditor (TPA) is introduced to help the client to periodicallycheck the integrity of the data in cloud. However, it is possiblefor the TPA to get the client's data after it executes the auditingprotocol multiple times. Auditing protocols are designed to ensure the privacy of the client's data incloud. Another aspect having been addressed in cloud storage auditing is how to support data dynamic operations.Wa n get al. have proposed an auditing protocol supportingfully dynamic data operations including modification, insertionand deletion. Auditing protocols can also support dynamic data operations. Other aspects, such as proxy auditing, user revocation and eliminating certificate management in cloud storage auditing have also been studied. Though many research worksabout cloud storage auditing have been done in recent years,a

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

critical security problem—the key exposure problem forcloud storage auditing, has remained unexplored in previousresearches. While all existing protocols focus on the faults ordishonesty of the cloud, they have overlooked the possibleweak sense of security and/or low security settings at theclient.In fact, the client's secret key for cloud storage auditing maybe exposed, even known by the cloud, due to several reasons.Firstly, the key management is a very complex procedure which involves many factors including system policy, usertraining, etc. One client often needs to manage varieties of keysto complete different security tasks. Any careless mistake orfault in managing these keys would make the key exposurepossible. It is not uncommon to see a client choosing touse cheap software-based key management for economical factors, which may only provide limited protectionand make the sensitive secret keys vulnerable to exposure.

Secondly, the client himself may be the target and vulnerableto many Internet based security attacks. For an ordinary client,the sense of security protection can be relatively weaker, compared with the case of enterprises and organizations.Hence, it is possible for a client to unintentionally downloadmalicious software from Internet or to overlookthe timely security patch to their computer system. Both ofthese cases could give the hacker easy access to their secretkeys. Last but not the least, the cloud also has incentivesto get clients' secret keys for storage auditing, e.g., throughtrading with the aforementioned hackers. Specifically, if thecloud gets these keys, it can regenerate the fake data andforge their authenticators to easily hide the data loss incidents,e.g., caused by Byzantine failures, from the client, whilemaintaining its reputation. In the malicious case,it can even discard the client's data that are rarely accessed to save the storage space, without worrying aboutfailure to pass the auditing protocol initiated by the client.Obviously, the auditing secret key exposure could be disastrousfor the clients of cloud storage applications.Therefore, how to deal with the client's secret key exposurefor cloud storage auditing is a very important problem.Unfortunately, previous auditing protocols did not considerthis critical issue, and any exposure of the client's secretauditing key would make most of the existing auditingprotocols unable to work correctly. In this paper, we focuson how to reduce the damage of the clients key exposure incloud storage auditing. Our goal is to design a cloud storageauditing protocol with built-in key-exposure resilience. How to do it efficiently under this new problem setting brings inmany new challenges to be addressed below. First of all,applying the traditional solution of key revocation to cloudstorage auditing is not practical. This is because, wheneverthe client's secret key for auditing is exposed, the client needsto produce a new pair of public key and secret key andregenerate the authenticators for the client's data previouslystored in cloud. The process involves the downloading ofwhole data from the cloud, producing new authenticators,and re-uploading everything back to the cloud, all of whichcan be tedious and cumbersome. Besides, it cannot alwaysguarantee that the cloud provides real data when the clientregenerates new authenticators. Secondly, directly adoptingstandard key-evolving technique is also not suitablefor the new problem setting. It can lead to retrieving all of theactual files blocks when the verification is proceeded. Thisis partly because the technique isincompatible with block lessverification. The resulting authenticators cannot be aggregated,leading to unacceptably high computation and communicationcost for the storage auditing.

SYSTEM ARCHITECTURE:



ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

MODULES DESCRIPTION:

Client:

The client produces files and uploads these files along with corresponding authenticators to the cloud.The client can periodically audit whether his files in cloud are correct. The client will update his secret keys for cloud storage auditing in the end of each time period, but the public key is always unchanged.

TPA:

In order to reduce the computational burden of the client, a third-partyauditor (TPA) is introduced to help the client to periodically check the integrity of the data in cloud. However, it is possible for the TPA to get the client's data 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 having been addressed in cloud storage auditing is how to support data dynamic operations.

Cloud:

Cloud computing is a model for enabling ubiquitous, convenient, on-demand access to a shared pool of configurable computing resources. Cloud computing and storage solutions provide users and enterprises with various capabilities to store and process their data in third-party data centers. It relies on sharing of resources to achieve coherence and economies of scale, similar to a utility (like the electricity grid) over a network. At the foundation of cloud computing is the broader concept of converged infrastructure and shared services.

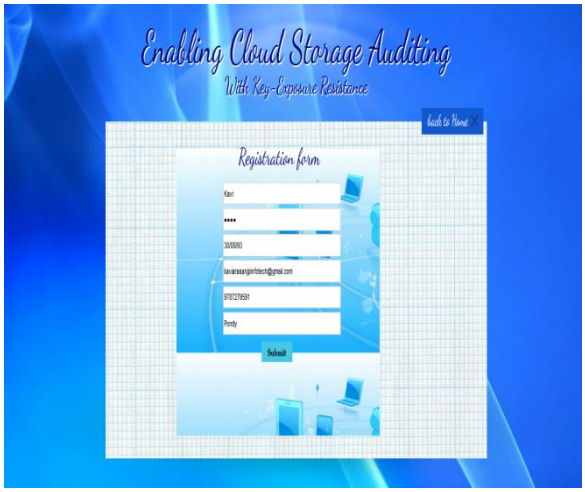
Key Exposure Resistance:

The client needs to produce a new pair of public key and secret key and regenerate the authenticators for the client's data previously stored in cloud. There is a one-time public key sharing for each file and a Time Stamp based secret key Generation. For each instance the timestamp based key exposure will be vary according to the current time stamp.

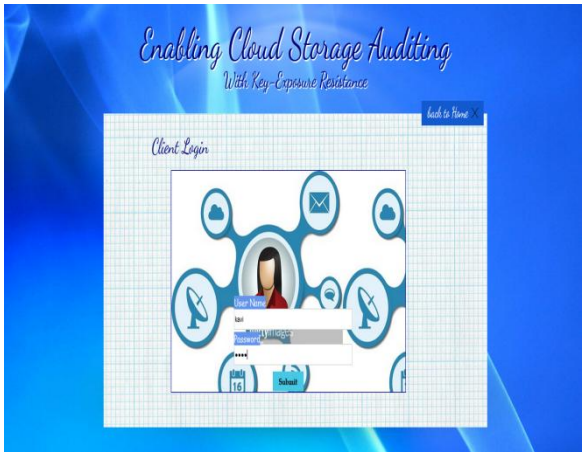
SCREEN SHOTS:

Home:

Registration:



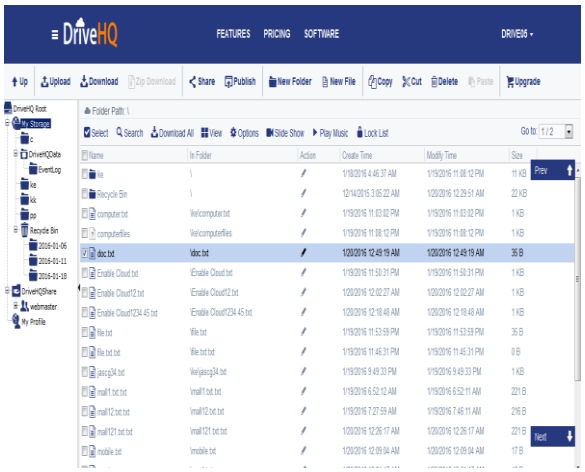
Client Login:



ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

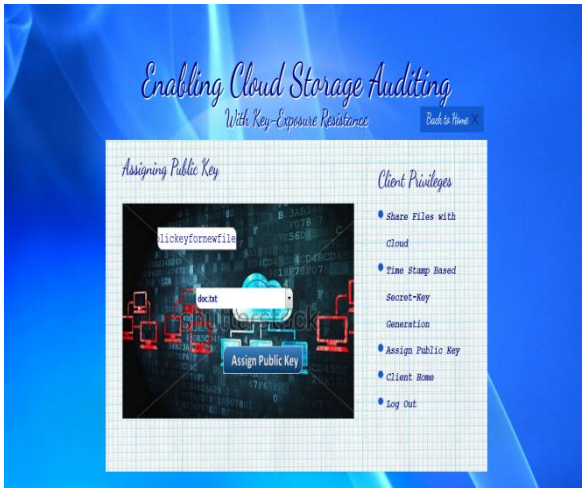
Client Home:

File Stored into Cloud:



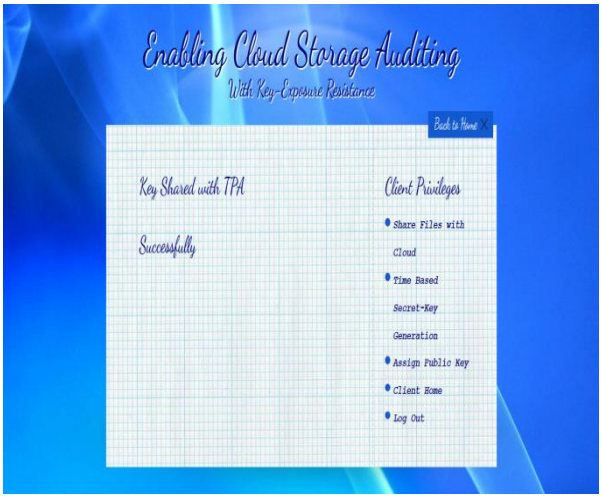
Share File in Cloud:

Assign Public Key:



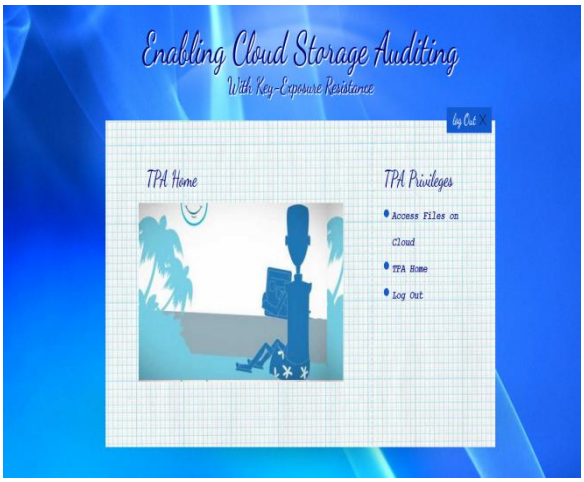
Cloud:

Key Shared with TPA:



ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

TPA Home:



CONCLUSION

In this paper, we study on how to deal with the client's key exposure in cloud storage auditing. We propose a new paradigm called auditing protocol with key-exposure resilience. In such a protocol, the integrity of the data previously stored incloud can still be verified even if the client's current secret key for cloud storage auditing is exposed. We formalize the definition and the security model of auditing protocol with key-exposure resilience, and then propose the first practical solution. The security proof and the asymptotic performance evaluation show that the proposed protocol is secure and efficient.

REFERENCES

[1] G. Ateniese et al., "Provable data possession at untrusted stores," in Proc. 14th ACM Conf. Comput. Commun. Secur., 2007, pp. 598–609.

[2] 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.

[3] F. Sebe, J. Domingo-Ferrer, A. Martinez-Balleste, Y. Deswarte, and J.-J. Quisquater, "Efficient remote data possession checking in critical information infrastructures," IEEE Trans. Knowl. Data Eng., vol. 20, no. 8, pp. 1034–1038, Aug. 2008.

[4] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "MR-PDP: Multiplereplica provable data possession," in Proc. 28th IEEE Int. Conf. Distrib. Comput. Syst., Jun. 2008, pp. 411–420.

[5] H. Shacham and B. Waters, "Compact proofs of retrievability," in Advances in Cryptology—ASIACRYPT. Berlin, Germany: Springer-Verlag, 2008, pp. 90–107.

[6] 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.

[7] Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S. S. Yau, "Efficient provable data possession for hybrid clouds," in Proc. 17th ACM Conf. Comput. Commun. Secur., 2010, pp. 756–758.

[8] 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.

[9] 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.

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

[11] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling public auditability and data dynamics for storage security in cloud computing," IEEE Trans. Parallel Distrib. Syst., vol. 22, no. 5, pp. 847–859, May 2011.

[12] 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.

[13] 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.

[14] H. Wang, "Proxy provable data possession in public clouds," IEEE Trans. Services Comput., vol. 6, no. 4, pp. 551–559, Oct./Dec. 2013.

[15] B. Wang, B. Li, and H. Li, "Public auditing for shared data with efficient user revocation in the cloud," in Proc. IEEE INFOCOM, Apr. 2013, pp. 2904–2912.

[16] 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.

[17] T. Stewart. (Aug. 2012). Security Policy and Key Management: Centrally Manage Encryption Key. [Online]. Available: http://www.slideshare. net/Tina-stewart/security-policy-and-enterprise-key-management-fromvormetric

[18] Microsoft. (2014). Key Management. [Online]. Available: http://technet.microsoft.com/en-us/library/cc961626.aspx

[19] FBI. (2012). Is Your Computer Infected with DNSChanger Malware?. [Online]. Available: http://www.fbi.gov/news/news_blog/is-yourcomputer-infected-with-dnschanger-malware