Strong Key-Exposure Resilient Auditing for Secure Cloud Storage

ABSTARCT

- Key exposure is one serious security problem for cloud storage auditing. In order to deal with this problem, cloud storage auditing scheme with key-exposure resilience has been proposed.
- However, in such a scheme, the malicious cloud might still forge valid authenticators later than the key-exposure time period if it obtains the current secret key of data owner.
- In this paper, we innovatively propose a paradigm named strong key-exposure resilient auditing for secure cloud storage,

- in which the security of cloud storage auditing not only earlier than but also later than the key exposure can be preserved. We formalize the definition and the security model of this new kind of cloud storage auditing and design a concrete scheme. In our proposed scheme, the key exposure in one time period doesn't affect the security of cloud storage auditing in other time periods. The rigorous security proof and the experimental results demonstrate that our proposed scheme achieves desirable security and efficiency

EXISTING SYSTEM

- Many cloud storage auditing schemes have been proposed up to now. These schemes consider several different aspects of cloud storage auditing such as the data dynamic update the privacy protection of user's data the data sharing among multiple clients and the multicopies of cloud data. Key-exposure resilience, as another important aspect, has been proposed recently . Indeed, the secret key might be exposed due to the weak security sense and/or the low security settings of the client. Once a malicious cloud gets the client's secret key for cloud storage auditing, it can hide the data loss incidents by forging the authenticators of fake data.

PROPOSED SYTEM

- (1) We investigate how to preserve the security of cloud storage auditing scheme in any time period other than the key-exposure time period when the key exposure happens. We propose a paradigm named strong key-exposure resilient auditing as a practical solution for this problem in this paper.
- (2) We design a concrete strong key-exposure resilient auditing scheme for secure cloud storage. A novel and efficient key update technique is used in the designed scheme. In our detailed construction, the Third Party Auditor (TPA) generates an update message from his secret key in each time period, and then sends it to the client

- (3) We formalize the definition and the security model of this new paradigm. In the security model, we consider the most powerful adversary who can query the secret keys of the client in all except one unexposed time period.

HARDWARE REQUIREMENTS

- System : Pentium IV 2.4 GHz.
- Hard Disk : 40 GB.
- Floppy Drive : 1.44 Mb.
- Monitor : 15 VGA Colour.
- Mouse : Logitech.
- Ram : 512 Mb.

SOFTWARE REQUIREMENTS

- Operating system : Windows XP/7.
- Coding Language : ASP.net, C#.net /java

Conclusion

- In this paper, we further study on how to deal with the key exposure problem in cloud storage auditing. We propose a new paradigm called strong key-exposure resilient auditing scheme for secure cloud storage. In this paradigm, the security of the cloud storage auditing not only earlier than but also later than the key exposure can be preserved. We formalize the definition and the security model of this new kind of cloud storage auditing and design a concrete scheme. The security proof and the experimental results demonstrate that the proposed scheme is secure and efficient.

Reference

- [1] F. Sebe, J. Domingo-Ferrer, A. Martinez-balleste, Y. Deswarte, and J. Quisquater, "Efficient Remote Data In-tegrity checking in Critical Information Infrastructures," IEEE Transactions on Knowledge and Data Engineering, vol. 20, no. 8, pp. 1-6, 2008.
- [2] C. Wang, K. Ren, W. Lou, and J. Li, "Toward Publicly Auditable Secure Cloud Data Storage Services," IEEE Network, vol. 24, no. 4, pp. 19-24, July/Aug. 2010.
- [3] Y. Zhu, H. Wang, Z. Hu, G. J. Ahn, H. Hu, and S. S. Yau, "Efficient Provable Data Possession for Hybrid Clouds," Proc. 17th ACM Conference on Computer and Communications Security, pp. 756-758, 2010.