Volume 5, Issue 11, November 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Study of Secure De Duplication using De key with Efficient and Reliable Convergent Key Management in Cloud Storage

[1]R. Thilagavathi[*], [2]S. Ramasamy, [3]R. K. Gnanamurthy
[1]PG Scholar, Dept of CSE, Vivekanandha College of Engineering for Women, Namakkal, Tamilnadu, India
[2]AP/CSE, Dept of CSE, Vivekanandha College of Engineering for Women, Namakkal, Tamilnadu, India
[3]Professor, Dept of ECE, SKP Engg College, Tiruvannamalai, Tamilnadu, India

Abstract-- Secure de duplication is a technique for eliminating duplicate copies of storage data, and provides security to them. To reduce storage space and upload bandwidth in cloud storage de duplication has been a well-known technique.Promising as it is, an arising challenge is to perform secure de duplication in cloud storage. This project makes the first attempt to formally address the problem of achieving efficient and reliable key management in secure de duplication.Although convergent encryption has been extensively adopted for secure de duplication, a critical issue of making convergent encryption practical is to efficiently and reliably manage a huge number of convergent keys. This project first introduces a baseline approach in which each user holds an independent master key for encrypting the convergent keys and outsourcing them to the cloud. However, such a baseline key management scheme generates an enormous number of keys with the increasing number of users and requires users to dedicatedly protect the master keys. To this end, this project proposes De key, a new construction in which users do not need to manage any keys on their own but instead securely distribute the convergent key shares across multiple servers.

Keywords----De duplication, De key, Convergent encryption, Asymmetric searchable encryption, Key Management

I. INTRODUCTION

The advent of cloud storage motivates enterprises and organizations to outsource data storage to third-party cloud providers, as evidenced by many real-life case studies. One critical challenge of today's cloud storage services is the management of the ever-increasing volume of data. According to the analysis report of IDC, the volume of data in the wild is expected to reach 40 trillion gigabytes in 2020. To make data management scalable, de duplication has been a well-known technique to reduce storage space and upload bandwidth in cloud storage. Instead of keeping multiple data copies with the same content, de duplication eliminates redundant data by keeping only one physical copy and referring other redundant data to that copy. Convergent encryption provides a viable option to enforce data confidentiality while realizing de duplication. The cipher texts can only be decrypted by the corresponding data owners with their convergent keys.

De key, which provides efficiency and reliability, guarantees for convergent key management on both user and cloud storage sides. Our idea is to apply de duplication to the convergent keys and leverage secret sharing techniques. Specifically, we construct secret shares for the convergent keys and distribute them across multiple independent key servers.

II. LITERATURE SURVEY

Paul Anderson [1] is to observe that there is a good deal of sharing between the data on typical laptops. For example, most (but not all) of the system files are likely to be shared with at least one other user. But equally importantly, it would. significantly reduce the time required for backups in most cases – upgrading an operating system, or downloading a new music file should not require any additional backup time at all if someone else has already backed-up those same files. There has been a lot of interest recently in de duplication techniques, using content-addressable storage (CAS). This is designed to address exactly the above problem. However, most of these solutions are intended for use in a local file system or SAN. This has two major drawbacks: (i) clients must send the data to the remote file system before the duplication is detected – this forfeits the potential saving in network traffic and time. And (ii) any encryption occurs on the server, hence exposing sensitive information to the owner of the service – this is usually not appropriate for many of the files on a typical laptop which are essentially personal rather than corporate4. Backing up to cloud-based storage becomes increasing popular in recent years. The main benefits of using a cloud storage are lower server maintenance cost, cheaper long term operational cost, and sometimes enhanced data safety via a vendor's own geographically diverse data replication**.** In particular, the benefits of employing a cloud-based secondary storage are:1. New cloud services can be added easily on the backup server to provide enhanced data safety and to reduce the risk of vendor lock-in.2. Upload cost to cloud storage can be reduced via data aggregation techniques.

In [5] ASE(Asymmetric searchable encryption**)** schemes are appropriate in any setting where the party searching over the data is different from the party that generates it. They refer to such scenarios as many writer/single reader (MWSR). The main advantage of ASE is functionality while the main disadvantages are inefficiency and weaker

security. Since the writer and reader can be different, ASE schemes are usable in a larger number of settings than SSE schemes. The inefficiency comes from the fact that all known ASE schemes require the evaluation of pairings on elliptic curves which is a relatively slow operation compared to evaluations of (cryptographic) hash functions or block ciphers. In addition, in the typical usage scenarios for ASE (i.e., MWSR) the data cannot be stored in efficient data structures. For most customers, this provides several benefits including availability (i.e., being able to access data from anywhere) and reliability (i.e., not having to worry about backups) at a relatively low cost. While the benefits of using a public cloud infrastructure are clear, it introduces significant security and privacy risks.

Dutch T. Meyer and William J. Bolosky [8] File systems often contain redundant copies of information: identical files or sub-file regions, possibly stored on a single host, on a shared storage cluster, or backed-up to secondary storage. De duplicating storage systems take advantage of this redundancy to reduce the underlying space needed to contain the file systems (or backup images thereof). De duplication can work at either the sub-file or whole-file level. More fine-grained de duplication creates more opportunities for space savings, but necessarily reduces the sequential layout of some files, which may have significant performance impacts when hard disks are used for storage. Alternatively, whole-file de duplication is simpler and eliminates file-fragmentation concerns, though at the cost of some otherwise reclaimable storage. They find that while block-based de duplication of their dataset can lower storage consumption to as little as 32% of its original requirements, nearly three quarters of the improvement observed could be captured through whole-file de duplication and sparseness. They also explore the parameter space for de duplication systems, and quantify the relative benefits of sparse file support. These challenges, the increase in un-structured files, and an ever-deepening and more populated namespace pose significant challenge for future file system designs.

Zooko Wilcox-O'Hearn and Brian Warner [16] Tahoe is a system for secure, distributed storage. It uses capabilities for access control, cryptography for confidentiality and integrity, and erasure coding for fault-tolerance. It has been deployed in a commercial backup service and is currently operational. The implementation is Open Source. Tahoe is a storage cloud designed to provide secure, long-term storage, such as for backup applications. It consists of users pace processes running on commodity PC hardware and communicating with one another over TCP/IP. Tahoe was designed following the Principle of Least Authority each user or process that needs to accomplish a task should be able to perform that task without having or wielding more authority than is necessary. Tahoe was developed by allmydata.com to serve as the storage backend for their backup service

Access Control:

Tahoe uses the capability access control model to manage access to files and directories. In Tahoe, a capability is a short string of bits which uniquely identifies one file or directory. Knowledge of that identifier is necessary and sufficient to gain access to the object that it identifies. The strings must be short enough to be convenient to store and transmit, but must be long enough that they are unguessable (this requires them to be at least 96 bits). Tahoe's security is based on cryptographic capabilities for decentralized access control, which have proven to be flexible enough to serve their requirements so far.

Wee Keong Ng and Huafei Zhu [9], a new notion which they call private data de duplication protocol, a de duplication technique for private data storage is introduced and formalized. Intuitively, a private data de duplication protocol allows a client who holds a private data proves to a server who holds a summary string of the data that he/she is the owner of that data without revealing further information to the server. Their notion can be viewed as a complement of the state-of-the-art public data de duplication. The security of private data de duplication protocols is formalized in the simulation-based framework in the context of two-party computations. A construction of private de duplication protocols based on the standard cryptographic assumptions is then presented and analyzed. They show that the proposed private data de duplication protocol is provably secure assuming that the underlying hash function is collision-resilient, the discrete logarithm is hard and the erasure coding algorithm can erasure up to α-fraction of the bits in the presence of malicious adversaries in the presence of malicious adversaries. To the best their knowledge this is the first de duplication protocol for private data storage.

De duplication protocol:

The server can store only a small amount of data per file in fast storage but it cannot afford to retrieve the file or parts of it from secondary storage upon every upload request. As a result, the private data de duplication scheme must allow the server to store only an extremely short information per file that will enable it to check claims from clients that they have that file without having to fetch the file contents for verification. A feasible result of private data de duplication protocols has been proposed and analyzed. They have shown that the proposed private data de duplication protocol is provably secure in the simulation-based framework assuming that the underlying hash function is collision-resilient, the discrete logarithm is hard and the erasure coding algorithm can erasure up to α-fraction of the bits in the presence of malicious adversaries.

Qian Wang and Cong Wang[14] This work studies the problem of ensuring the integrity of data storage in Cloud Computing. In particular, they 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 is indeed intact, which can be important in achieving economies of scale for Cloud Computing. While prior works on ensuring remote data integrity often lacks the support of either public auditability or dynamic data operations, achieves both. In particular, to achieve efficient data dynamics, they 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, they further explore the technique of bilinear aggregate signature to extend their main result into a multi-user 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.

In order to solve the problem of data integrity checking, many schemes are proposed under different systems and security models . In all these works, great efforts are made to design solutions that meet various requirements: high scheme efficiency, stateless verification, unbounded use of queries and retrievability of data, etc. Considering the role of the verifier in the model, all the schemes presented before fall into two categories: private auditability and public auditability. Although schemes with private auditability can achieve higher scheme efficiency, public auditability allows any- one, not just the client (data owner), to challenge the cloud server for correctness of data storage while keeping no private information.

Weichao Wang and Zhiwei Li[15] Providing secure and efficient access to large scale outsourced data is an important component of cloud computing. Authors propose a mechanism to solve this problem in owner-write-users-read applications. They propose to encrypt every data block with a different key so that flexible cryptography-based access control can be achieved. Through the adoption of key derivation methods, the owner needs to maintain only a few secrets. Analysis shows that the key derivation procedure using hash functions will introduce very limited computation overhead.

They design mechanisms to handle both updates to outsourced data and changes in user access rights. They investigate the overhead and safety of the proposed approach, and study mechanisms to improve data access efficiency. Author focus on the data outsourcing scenario investigated in this environment, the data can be updated only by the original owner. At the same time, end users with different access rights need to read the information in an efficient and secure way. Both data and user dynamics must be properly handled to preserve the performance and safety of the outsourced storage system. The proposed approach provides fine grained access control to outsourced data with flexible and efficient management. The data owner needs to maintain only a few secrets for key derivation.They analyze the computational, storage, and communication overhead of the approach. They also investigate the scalability and safety of the approach.

In proposed system using De key technique constructs secret shares on the original convergent keys (that are in plain) and distributes the shares across multiple KM-CSPs (Key Management-Cloud Service Provider). If multiple users share the same block, they can access the same corresponding convergent key. If the original (first) user of the group intimates the server with a user's (B) revocation, then the server rejects the proof of ownership submitted by that user (B). Likewise, session based de duplication is considered. Here if the user provides the session duration i.e., front date and to date, then only with the data range, proof of ownership can be allowed in server on those dates. This increases the security if the outsourced data need to be safely accessed on the given duration. In proposed system session based de duplication is considered. Here if the user provides the session duration i.e., front date and to date, then only with the data range, proof of ownership can be allowed in server on those dates. This increases the security if the outsourced data need to be safely accessed on the given duration.

III. CONCLUSION

Study of this paper De key, an efficient and reliable convergent key management scheme for secure de duplication. De key applies de duplication among convergent keys and distributes convergent key shares across multiple key servers, while preserving semantic security of convergent keys and confidentiality of outsourced data. We implement De key using the Ramp secret sharing scheme and demonstrate that it incurs small encoding/decoding overhead compared to the network transmission overhead in the regular upload/download operations.

REFERENCES

[1] P. Anderson and L. Zhang, ''*Fast and Secure Laptop Backups with Encrypted De-Duplication,*'' in Proc. USENIX LISA, 2010,pp. 1-8.

[2] M. Bellare, S. Keelveedhi, and T. Ristenpart, ''*Message-Locked Encryption and Secure De duplication,*'' in Proc. IACR Cryptology Print Archive, 2012, pp. 296-3122012:631.

[3] G.R. Blakley and C. Meadows, ''*Security of Ramp Schemes*,'' inProc. Adv. CRYPTO, vol. 196, Lecture Notes in Computer Science,G.R. Blakley and D. Chaum, Eds., 1985, pp. 242-268.

[4] J. Gantz and D. Reinsel, The Digital Universe in 2020: Big Data, Bigger Digital Shadows, Biggest Growth in the Far East, Dec. 2012.[Online]. Available: http://www.emc.com/collateral/analystreports/idc-the-digital-universe-in-2020.pdf. Peleg, ''*Side Channels in Cloud Services: Deduplication in Cloud Storage,*'' IEEE Security Privacy, vol. 8, no. 6, pp. 40-47, Nov./Dec. 2010.

[5] S. Kamara and K. Lauter, ''*Cryptographic Cloud Storage*,'' in Proc. Financial Cryptography: Workshop Real-Life Cryptograph.Protocols Standardization, 2010, pp. 136-149.

[6] D. Meister and A. Brinkmann, ''*Multi-Level Comparison of Data Deduplication in a Backup Scenario*,'' in Proc. SYSTOR, 2009,pp. 1-12.

[7] M.W. Storer, K. Greenan, D.D.E. Long, and E.L. Miller, ''*Secure Data Deduplication*,'' in Proc. StorageSS, 2008, pp. 1-10.

[8] D.T. Meyer and W.J. Bolosky, ''*A Study of Practical Deduplication,*'' in Proc. 9th USENIX Conf. FAST, 2011, pp. 1-13.

[9] W.K. Ng, Y. Wen, and H. Zhu, *''Private Data De duplication Protocols in Cloud Storage*,'' in Proc. 27th Annu. ACM Symp.Appl. Comput., S. Ossowski and P. Lecca, Eds., 2012, pp. 441-446.

[10] M.O. Rabin, ''*Efficient Dispersal of Information for Security, Load Balancing, Fault Tolerance*,'' J. ACM, vol. 36, no. 2, pp. 335-348, Apr. 1989.

[11] A. Rahumed, H.C.H. Chen, Y. Tang, P.P.C. Lee, and J.C.S. Lui, ''*A secure Cloud Backup System with Assured Deletion and Version Control*,'' in Proc. 3rd Int'l Workshop Security Cloud Comput., 2011, pp. 160-167.

[12] Y. Tang, P.P. Lee, J.C. Lui, and R. Perlman, ''*Secure Overlay Cloud Storage with Access Control and Assured Deletion,*''IEEE Trans. Dependable Secure Comput., vol. 9, no. 6, pp. 903-916, Nov./Dec. 2012.

[13] G. Wallace, F. Douglis, H. Qian, P. Shilane, S. Smaldone, M. Chamness, and W. Hsu, ''*Characteristics of Backup Workloads in Production Systems*,'' in Proc. 10th USENIX Conf. FAST, 2012, pp. 1-16.

[14] 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.

[15] W. Wang, Z. Li, R. Owens, and B. Bhargava, ''*Secure and Efficient Access to Outsourced Data*,'' in Proc. ACM CCSW, Nov. 2009, pp. 55-66.

[16] Z. Wilcox-O'Hearn and B. Warner, ''*Tahoe: The Least-Authority Filesystem*,'' in Proc. ACM StorageSS, 2008, pp. 21-26.

