Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 5.258

IJCSMC, Vol. 5, Issue. 3, March 2016, pg.461 – 466

EFFICIENT APPROACH FOR DATA RETRIEVABILITY ON CLOUD STORAGE SYSTEM

Muthulakshmi.K, B.E CSE final yr, IFET College of Engineering, Villupuram

Kumudhapriyaa.M, B.E CSE final yr, IFET College of Engineering, Villupuram

Kirubanandam.T, B.E CSE final yr, IFET College of Engineering, Villupuram

Pandiyavathi.T, M.E, Assistant Professor, CSE, IFET College of Engineering, Villupuram

Abstract: Cloud storage is a model of data storage in which the digital data is stored in logical pools. It allows users to store their data in a remote server to get rid of expensive local storage and management costs and then access data of interest anytime anywhere. We propose an enhanced dynamic proof of retrievability scheme supporting public audit ability and communication-efficient recovery from data corruptions. We split up the data into small data blocks and encode that data block using network coding. To eliminate the communication overhead for small data corruptions within a server, each encoded data block is further encoded. Based on the encoded data blocks, we utilize tree structure to enforce the data sequence for dynamic operations, preventing the cloud service provider from manipulating data block to pass the integrity check in the dynamic scenario. We also analyze the effectiveness of the proposed construction in defending against pollution attacks during data retrievability.

Keywords: Cloud storage, Data retrievability, Data integrity, Public audit, and Data dynamic.

I. Introduction

A cloud refers to a distinct IT environment that is designed for the purpose of remote provisioning scalable and measured IT resources. The term originated as a metaphor for the Internet which is, in essence, a network of networks providing remote access to a set of decentralized IT resources. Prior to cloud computing becoming its own formalized IT industry segment, the symbol of a cloud was commonly used to represent the Internet in a variety of specifications and mainstream documentation of Web-based architectures. This same symbol is now used to specifically represent the boundary of a cloud environment, as shown in Figure1.It is important to distinguish the term "cloud" and the cloud symbol from the Internet. As a specific environment used to remotely provision IT resources, a cloud has a finite boundary. There are many individual clouds that are accessible via the Internet.Whereas the Internet provides open access to many Web-based IT resources, a cloud is typically privately owned and offers access to IT resources that ismetered.Many approach has proposed implementing design diversity techniques to increase the reliability, availability and security of large-scale systems. However, none of them have explicitly linked the distribution of resources to risk and correlation between different candidate providers. The challenge would be to find an efficient and effective solution for investing in diversity while considering the risk and correlation between providers.

II. Literature Survey

Bo Chen [1] in this paper Remote Data Checking (RDC) is a technique by which clients can establish that data outsourced at untrusted servers remains intact over time. RDC is useful as a prevention tool, allowing clients to periodically check if data has been damaged, and as a repair tool whenever damage has been detected. context of a single server, RDC was later extended to verify data integrity in distributed storage systems that rely on replication and on erasure coding to store data redundantly at multiple servers. Recently, a technique to add redundancy based on network coding, which offers interesting tradeoffs because of its remarkably low communication overhead to repair corrupt servers. RDC-NC, a novel secure and efficient RDC scheme for network coding-based distributed storage systems. RDC-NC mitigates new attacks that stem from the under- lying principle of network coding. It is able to preserve in an adversarial setting the minimal communication overhead of the repair component achieved by network coding in a benign setting. They implement experimentally show that it is computationally inexpensive for both clients and servers.

Jia Xu [2] proposed that Proofs of Retrievability (POR) is a cryptographic formulation for remotely auditing the integrity of files stored in the cloud, without keeping a copy of the original files in local storage. In a POR scheme, a user Alice backups her data file together with some authentication data to a potentially dishonest cloud storage server Bob. Later, Alice can periodically and remotely verify the integrity of her data file using the authentication data, without retrieving back the data file. this incorporate a recent construction of constant size polynomial commitment scheme (Kate, Zaverucha and Goldberg, Asiacrypt ’10) into Shacham and Waters scheme. The resulting scheme requires O(λ) communication bits (particularly, 920 bits if a 160 bits elliptic curve group is used or 3512 bits if a 1024 bits modulo group is used) per verification and a factor of 1/s file size expansion. Experiment results show that scheme is indeed efficient and practical. Our security proof is based on Strong Diffie Hellman Assumption.

Emil Stefanov [3] present Iris, a practical, authenticated file system designed to support workloads from large enterprises storing data in the cloud and be resilient against potentially untrustworthy service providers. As a transparent layer enforcing strong integrity guarantees, Iris lets an enterprise tenant maintain a large file system in the cloud. In Iris, tenants obtain strong assurance not just on data integrity, but also on data freshness, as well as data

retrievability in case of accidental or adversarial cloud failures. Iris offers an architecture scalable to many clients (on the order of hundreds or even thousands) issuing operations on the file system in parallel. Iris includes new optimization and enterprise-side caching techniques specifically designed to overcome the high network latency typically experienced when accessing cloud storage. Iris also includes novel erasure coding techniques for efficient support of dynamic Proofs of Retrievability (PoR) protocols over the file system. Architecture and experimental results on a prototype version of Iris. Iris achieves end-to-end throughput of up to 260MB per second for 100 clients issuing simultaneous requests on the file system. Demonstrate that strong integrity protection in the cloud can be achieved with minimal performance degradation.

Cong Wang [4] Cloud Storage, users can remotely store their data and enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources, without the burden of local data storage and maintenance. However, the fact that users no longer have physical possession of the outsourced data makes the data integrity protection in Cloud Computing a formidable task, especially for users with constrained computing resources. Thus, enabling public auditability for cloud storage is of critical importance so that users can resort to a third party auditor (TPA) to check the integrity of outsourced data and be worry-free. To securely introduce an effective TPA, the auditing process should bring in no new vulnerabilities towards user data privacy, and introduce no additional online burden to user. 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.

III. Proposed System

We implement a technique with dynamic proof of retrievability scheme supporting public audit ability and communication efficient recovery from data corruptions. To this end, split up the data into small data blocks and encode each data block individually using network coding. Network coding and erasure codes are adopted to encode data blocks to achieve within server and cross server data redundancy, tolerating data corruption. By combining range based 2-3 tree and improved version of aggregately signature based broadcast encryption, our construction can support efficient data dynamics while defending against data replay attack.

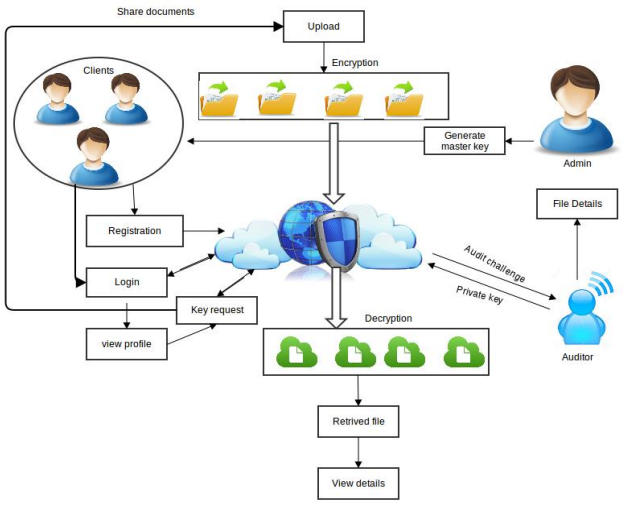

Figure 1. System Architecture

1) Key generation and File upload

Key is the set of character or number or symbol. Key is generated from the auditor. The private key is generated to the user to encrypt the uploaded file. The private key is generated from the auditor.

The master key is generated from the admin. User details are stored in the database. Key detail and user details are maintained by the separate table. After key generation user upload the file by using the master key.

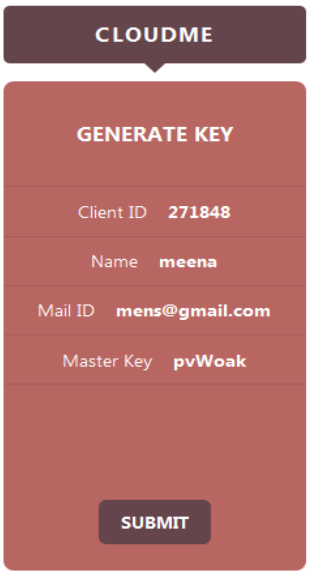

Figure 2.Key generation

The above diagram shows the user details and the master key which generated from the admin.

Algorithm used

i)Base64

Base64 is a group of similar binary-to-text encoding schemes that represent binary data in an ASCII string format by translating it into a radixx-64 representation. The term *Base64* originates from a specific MIME(Multipurpose Internet Mail Extensions) content transfer encoding.

ii)Network coding

Network coding is a technique which can be used to improve a network's throughput, efficiency and scalability as well as resilience to attacks and eavesdropping. Instead of simply relaying the packets of information they receive, the nodes of a network take *several* packets and combine them together for transmission. This can be used to attain the maximum possible information flow in a network.

2) Encryption

Base64 is used commonly in a number of applications including email via MIME, and storing complex data in XML. The particular choice of characters to make up the 64 characters required for base varies between implementations.

The process of encoding is:

1. Divide the input bytes stream into blocks of 3 bytes.

2. Divide 24 bits of each 3-bytes block into 4 groups of 6 bits.

3. Map each group of 6 bits to 1 printable character, base on the 6-bit value using the base64 character set map.

4. If the last 3-byte block has only 1 byte of input data, pad 2 bytes of zero (\x0000).

5. After encoding it as a normal block, override the last 2 characters with 2 equal signs (= =), so the decoding process knows 2 bytes of zero were padded.

6.If the last 3-bytes block has only 2 bytes of input data, pad 1 byte of zero (\x00).After encoding it as a normal block, override that last 1 character with equal signs (=), so the decoding process knows 1 byte of zero was padded.

7. Carriage return (\r) and new line (\n) are inserted into the output character stream. They will be ignored by the decoding process.

3) Splitting the data

Recent work in coding for distributed storage [8, 9] has shown that the k network overhead factor for the repair component is not unavoidable (as it was commonly believed). Given a file represented by m input blocks, b1, b2... bm, the client uses network coding to generate coded blocks as linear combinations of the original m file blocks. Each input block bi can be viewed as a column vector: bi = (bi1, bi2...biu), where bij are elements in a finite field GF (2w) and are referred to as symbols.

IV. System Implementation

The system output is mainly based on encoding method. It will be performed by using base64 algorithm. Then the data have been split into number of blacks to store the data to cloud. The reference of the block to be stored in each every block. It is used to avoid the data corruption. The loaded file to be encrypted and split into the number of blocks. It can be encrypted by using the private key generated by the auditor.

V. Conclusion

In this paper, we proposed a new dynamic proof of retrievability scheme for coded cloud storage systems. Networks coding and erasure codes are adapted to encode data blocks to achieve redundancy, tolerating data corruptions and supporting communication efficient data recovery. By combining range-based 2-3 tree and an improved version of aggregatable signature based broadcast (ASBB) encryption, our construction can support efficient data dynamics while defending against data replay attack and pollution attack. Security analysis and experimental evaluations demonstrated the practicality of our construction in coded cloud storage systems.

VI. Future work

In future, we plan to Increase the security by providing key for the encrypted block files. And giving alert to the data owner if the key is misused. Increasing the speed of data integrity. Auditing details also send to the data owners through email by weekly and monthly basis.

References

[1] Bo Chen, Reza Curtmola ,Giuseppe Ateniese and Randal Burns "Remote Data Checking for Network Coding-based Distributed Storage Systems" in Proc.IEEE Conf. High Performance Comput.Commun., Sep.2008,pp.5-13.

[2] Jia Xu, Ee-Chien Chang "Towards Efficient Proofs of Retrievability" in Proc. Int. Conf. Very large Data bases, 2007, pp. 111-122.

[3]. Emil Stefanov, Marten van Dijk AlinaOprea "Iris: A Scalable Cloud File System with Efficient Integrity Checks" in Proc. IEEE Conf. Dec.2009, pp.99-120, 2008.

[4] Cong Wang, S.M. Chow, Qian Wang, Kui Ren, and Wenjing Lou" Privacy-Preserving Public Auditing for Secure Cloud Storage" in Proc.IEE Conf. 2010, pp. 411-426.

[5] B.Krebs, "Payment Processor Breach May Be Largest Ever," http://voices.washingtonpost.com/securityfix/2009/01/payment processor breach may b.html, 2009.

[6] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peter- son, and D. Song, "Provable Data Possession at Untrusted Stores," Proc. 14th ACM Conf. Computer and Comm. Security (CCS'07), pp. 598-609, 2007.

[7] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling Public Verifiability and Data Dynamic for Storage Security in Cloud Computing," Proc. 14th European Symp. Research in Computer Security (ESORICS'09), pp. 355-370, 2009.

[8] Dimakis, A. G., Godfrey, B., Wainwright, M. J., AND Ramchandran, K. Network coding for distributed storage systems. In INFOCOM (2007).

[9] Dimakis, A. G., Godfrey, P. B., WU, Y., M. O., And Ramchandran, WainwrighT K. Network coding for distributed storage systems. IEEE Transactions on Information Theory (2010).

[10] H. Shacham and B. Waters, "Compact Proofs of Retrievability," Proc. 14th Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT'08), pp. 90-107, and 2008.

