IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 1 Issue 10, December 2014.
www.ijiset.com
ISSN 2348 – 7968

A Secure Erasure Code-based Cloud Storage System with Secure Data Forwarding: A Review

Ujwala Salunke, Dayanand Kendre, Gauri Gonjari, Dinesh Bhor, Mrs. Minakshi Vharkate

Department of Computer Science and Engineering ,MIT Acadamy Of Engineering, Pune,India

Abstract

A collection of storage servers meant to be a cloud storage system which provides a long term storage services over the internet. Data confidentiality is a major issue when the data is stored on third party's cloud. General encryption scheme protect data confidentiality. As over encrypted data only few operations are supported when encrypted using general encryption scheme which in turn results to the limitation of functionality. System which is distributed in manner and has no central authority is difficult to implement. Integration of a threshold proxy re-encryption scheme and decentralized erasure code, a secure storage system is developed. The distributed storage system supports secure and robust data storage and retrieval. It also lets a user forward his data in storage servers to another user without retrieving the data back. The proposed system can be used for military and hospital applications and for other secret data transmission.

Keywords: Decentralized erasure code; proxy reencryption; threshold cryptography; secure storage system;

1. INTRODUCTION

In recent years, many services are provided on the internet such that users can use them from anywhere at anytime because of high-speed networks and ubiquitous Internet access available. For example, the email service is very popular. Cloud is consider as an unified entity which treats the resources on the internet in the concept of cloud computing. We have to just use services without being concerned about how the computation is done and storage is managed. A cloud storage system is considered as a large distributed storage system that consists of independent storage servers for storage purpose.

Data robustness is a major issue for storage systems. Robustness can be achieved by duplicating a message so that each storage server stores an individual copy of the message. It is very robust in nature since the message can be retrieved back as long as one storage server survives. Another way is encode a message of k length to an erasure error of the codeword. We can recover the message back from the code word symbols stored individually in the storage servers by the process of decoding.

At the early years, the Network Attached Storage (NAS) and the Network File System (NFS) are used to provide extra storage devices on the network such that a user can access the storage devices via network connection. Afterward, many improvements on issues like scalability, robustness, efficiency, and security were proposed.

2. EXISTING SYSTEMS

2.1. PLUTUS

Plutus is a cryptographic storage system. It avoids placing trust in the storage system itself. The owners of the files on the storage system are responsible for distributing encryption keys. Each file block is used as a unique encryption key. All the file-block keys are read and written using file-lockbox keys. Each file-lockbox contains all the keys for a filegroup. The intention of Plutus is to create a cryptographic storage system which minimized trust in the server. Plutus is more scalable, because most of the cost is incurred by the user. Plutus is mostly successful; however there are still attacks that are not complete addressed.

Advantages of Plutus:

✓It Protects against attacks on the physical device for security purpose
✓Flexible
✓The Access policies are set arbitrarily by the users for access control
✓High server scalability
✓ Crypto, key management and distribution done by clients.

Limitations and vulnerabilities:

✓Replication is not considered.
✓Users must trust their local machine.
✓User authentication is not part of the system.
✓Access patterns are not obfuscated.
✓This scheme reduces the number of keys needed for the file system. This increases vulnerability.

2.2 FARSITE

Federated, Available and Reliable Storage for an Incompletely Trusted Environment

Farsite is a serverless, distributed file system that does not assume mutual trust among the client computers on

which it runs. If we see logically the system functions as a central file server, but there is no central server machine physically. Instead, a group of client computers establish a virtual file server that can be accessed by any client. The system provides a global name space for files as well as location-transparent access to both private files and shared public files. Thus it improves reliability relative to storing files on a desktop workstation by distributing multiple encrypted replicas of each file among a set of client machines. Files can be referenced through a hierarchical directory structure that is maintained by a distributed directory service.

Advantages of Farsite:

✓ Reliability and Availability
- Long-term persistence,
- Immediate accessibility of file data during request.
- Files replicated via simple replication

✓ Security
- Directory groups maintain an access control list (ACL) of public keys of all authorized writers.
- File key is encrypted with public keys of all authorized readers of the file. Encrypted file keys are stored with file.

Drawbacks and vulnerabilities:

✓ it uses loosely coupled, unsecured and unreliable machines to establish a secured and reliable virtual file server.

✓ FARSITE does not update at once all replicas of a file. It would be too slow.

2.3 OCEANSTORE

Architecture for Global-Scale Persistent Storage

OceanStore is a global persistent data store designed to scale to billions of users. It provides consistent as well as highly-available, and durable storage utility constructing an infrastructure comprised of untrusted servers. Any computer who wants storage can join the infrastructure, contributing storage. Users need only to be subscribed to a single OceanStore service provider, they may have storage from many different providers. The providers can automatically buy and sell capacity by themselves. The utility model thus combines the resources from federated systems to provide a quality of service higher than that achievable by any single company.

OceanStore caches data; any server can create a local replica of data object. These local replicas give faster access and robustness to network partitions. They reduce high network congestion by localizing access traffic. We must assume that any server in the infrastructure may get crash, may leak information, or may become compromised. Promiscuous caching requires redundancy as well as cryptographic techniques to protect the data from the servers upon which it is stored. OceanStore uses a Byzantine-fault tolerant commit protocol so to provide strong consistency across replicas.

Advantages of OceanStore:

✓ Security
- Encrypt all data that is not public and distribute the key to users with read permission.
- Writes are signed; checked against an access control list (ACL)

✓ Distributed Routing
- Location independent addressing, using GUID (Globally Unique ID)

Limitations of Oceanstore:

✓ Deterministic algorithm is slow as compared to probabilistic algorithm which is applied before Deterministic algorithm.

2.4 PAST

A large-scale, persistent, peer-to-peer storage utility

Peer-to-Peer (P2P) systems can be characterized as distributed systems in which all nodes have identical capabilities and responsibilities and all communication is symmetric. PAST is An Internet-based, P2P global storage utility, which aims to provide strong persistence with high availability as well as scalability and security. It is based on a self-organizing. It includes internet based network of storage nodes that cooperatively route file queries as well as store additional multiple replicas of files, and cache copies of popular files also for faster use. Storage nodes and files are each assigned uniformly distributed identifiers.

PAST is composed of nodes connected to the Internet, where each node is capable of routing client requests to insert or retrieve files. PAST nodes form a self-organizing network. Inserted files are replicated across multiple nodes for availability.

Advantages of PAST:

✓ Exploits the multitude and diversity of nodes in the Internet to achieve strong persistence and high availability

✓ No need for physical transport of storage media to protect backup and archival data

✓ No need for explicit mirroring to ensure high availability and throughput for shared data.

Limitations of PAST:

✓ Maintaining K replicas are very difficult in PAST.

✓ Load balancing is difficult to achieve in Past.

2.5 GLACIER

Managing and storing long-term digital archives is a challenge for almost every company. With Amazon Glacier, IT organizations now have a solution that removes the burden

ISSN 2348 – 7968

of digital archiving, storing and provides extremely low cost storage. Amazon Glacier stores data as archives, which are then uploaded to Glacier and organized in vaults. Customers can control access to using the AWS Identity and Access Management (IAM) service. Retrieving of data can be achieved by scheduling a job, which generally completes within 3 to 5 hours.

Amazon Glacier works with AWS services such as Amazon S3 and the different AWS Database services.

Advantages of Glacier:

✓ Glacier ensures durability of unrecoverable data in a cooperative, decentralized storage system, despite large scale correlated Byzantine failures.

✓ It does not rely on Introspection which has inherent limitation to capture all sources of correlated failures.

✓ Glacier uses raw, unreliable storage available at nodes to provide hard durability guarantees.

3. CONCLUSION

This paper presents the hypothetical study of various existing systems for cloud storage. Glacier is proved as an efficient distributed storage system by Amazon. Amazon *Glacier* is a secure, durable, and extremely low-cost *storage* service for data archiving and online backup.

ACKNOWLDEGMENT

We would like to thank Mrs. Minakshi Vharkate, Department of Computer Engineering, MIT Academy of Engineering, Alandi for their valuable comments and guidance.

References

[1] J. Kubiatowicz, D. Bindel, Y. Chen, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao, *"Oceanstore: An Architecture for Global-Scale Persistent Storage,"* Proc. Ninth Int'l Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS), pp. 190- 201, 2000.

[2] P. Druschel and A. Rowstron, *"PAST: A Large-Peer-to-Peer Storage Utility,"* Proc. Eighth Workshop Hot Topics in Operating System (HotOS VIII), pp. 75-80, 2001.

[3] A. Adya, W.J. Bolosky, M. Castro, G. Cermak, R. ceur, J. Howell, J.R. Lorch, M. Theimer, and R. Wattenhofer, *"Farsite: Federated, Available, and Reliable Storage for an Incompletely Trusted Environment,"* Proc. Fifth Symp. Operating System Design and Implementation (OSDI), pp. 1-14, 2002

[4] A. Haeberlen, A. Mislove, and P. Druschel, *"Glacier: Highly Durable, Decentralized Storage Despite Massive Correlated Failures,"* Proc. Second Symp. Networked Systems Design and Implementation (NSDI), pp. 143-158, 2005.

[5] H.-Y. Lin and W.-G. Tzeng, *"A Secure Decentralized Erasure Code for Distributed Network Storage,"* IEEE Trans. Parallel and Distributed Systems, vol. 21, no. 11, pp. 1586-1594, Nov. 2010. on Parallel and Distributed Systems, vol. 24, pp. 205-225, 2012.