



# File Operation using Secrete Sharing & Hadoop

**Mr. Sujay P. Pawar**

Assistant Professor, Department of Computer Science, D Y Patil Institute of Engineering, Ambi, Talegaon, Dist - Pune, India.  
pawarsujay.85@gmail.com

## ABSTRACT

*In today's high tech technology modern world everyone mostly store their Personal data in the Cloud which may has passwords, account numbers and other important information that may be used and misused by a miscreant, a opponent, or a court of law. These data are read, copied, and archived by Cloud Service Providers (CSPs), often without authors permission and control. Self-destructing data plays a very important role in protecting the user data's privacy. All the data stored at servers and their copies destroyed after a user-specified time, and also this data became unreadable for any user intervention. The decryption key is destructed after the user-specified time that is ttl(time-to-live) field. In proposed system, we present self-destructing data system that meets this challenge through a novel integration of secure cryptography techniques with active storage techniques based on 'hadoop'. We implement a proof-of-concept SeDas prototype. By functionality and security properties evaluate the SeDas prototype, the results conclude that SeDas is practical to use and achieve all the privacy-preserving aims described. Compared to the system without self-destructing data mechanism, performance of uploading and*

*downloading with the proposed SeDas acceptably decreases, while latency for uploading and downloading operations with self-destructing data mechanism increases.*

**Keywords:** Active storage; Cloud computing; Data privacy; self-destructing data.

## 1. INTRODUCTION

With the fast development of Cloud computing and popularization of mobile Internet, cloud services are becoming much important for peoples as a part of life. Peoples are more or less desired to store or post some personal private information on the Cloud using Internet. When people doing this, they hope that service providers will provide security policy to secure their data from hacking, so others people will not lose their privacy.

Scheme starts with a secret and then assume from it certain shares which are distributed among users (i.e., participants). The secret may be variously resolve (i.e., recovered) only by certain predetermined subgroups of users which constitute the access structure. The important category of access structure is



the  $(w, N)$ -threshold access structure in which, given  $N$  shareholders, an authorized group consist of any  $w$  or more participants and any group of at most  $w-1$  participants is an illegal group.

As peoples attracted more and more to the Internet and Cloud, privacy of personal data is at more risk. On the one hand, when database is being in operation by the current system or network, systems or network must cached, copy or archive it. These copies are essential for computer system and the network. However, people don't have knowledge about these copies and cannot control them, so these copies may leak their confidentiality. On the other hand, their privacy also can be leaked via Cloud Service Providers (CSP's), hacker's intrusion or some fair actions. These problems present dangerous challenges to protect people's privacy. Secret sharing has broad applications in this real world and can be used for situations in which access to important resources to be protected. There is old story which have motivated the secret sharing principle [8]: a group of pirates accidentally detected a map that would lead them to an enclave full of treasure. Who was going to be entrusted to keep the map? Safe result is: the map should be divided into pieces such that all pieces are needed to reconstruct the map and missing any piece would make the map which is not readable. Thus, every pirate was given one such piece. Another important

function or operation of secret sharing is e-voting where the vote of every single will be absolutely and correctly counted in the voting result but there is no way for their people (including candidates and authorities) to know whom the individual voted for [12].

Today database and networking era, sharing data secretly is also a fundamental issue in network security and can be used in key administration and multi-party secure computation [7]. Since the concept of secret sharing, along with an efficient structure to accomplish it, was proposed by Shamir in 1979[20]. Blakley also did the similar work at same time [5], there have been many papers approaching Shamir's scheme and designing new secret sharing schemes [2][7][9][10][12][15].

Secret sharing schemes can be classified into various categories according to different criteria's. In terms of multiple numbers of secrets to be divided, two classes can be identified: single secret and multiple secrets.

Our contributions are summarized as follows.

- 1) We focus on the security and multiple client handling, Map Reduce the flavor of Hadoop provide the functionality to handle multiple client efficiently.
- 2) Based on key distribution algorithm, Shamir's algorithm [1], which is used as the core algorithm to implement distributing keys in the system. We use these methods to implement a safety destruct with equal divided key (Shamir Secret Shares [1]).

- 3) We use proof-of-concept Se Das prototype to store and manage the equally divided key.
- 4) Through functionality and security properties evaluation of the system, the expected results that system is practical to use and meets all the privacy-preserving goals. The system also handles multiple clients efficiently without affecting the speed of server.
- 5) System supports security erasing files and random encryption keys stored in a hard disk drive (HDD) or solid statedrive (SSD), respectively.

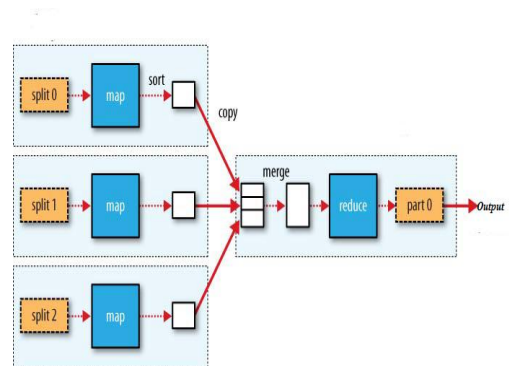
The system can be used for online shopping servers, social sites etc.

## 2. TECHNIQUES

### Map-Reduce

MapReduce is a programming model for parallel data processing. The model is simple, yet not too simple to express useful programs in. Hadoop can run MapReduce programs written in various languages like Java, Ruby, Python, and C++. Most important, MapReduce programs are inherently parallel, thus putting very large-scale data analysis into the hands of anyone with enough machines at their disposal. MapReduce comes into its own for large datasets, so let's start by looking at one.

MapReduce works by breaking the processing into two phases: the map phase and the reduce phase. Each phase has key-value pairs as input and output, the types of



which may be chosen by the programmer. The programmer also specifies two functions: the map function and the reduce function.

- The client submits the MapReduce job.

Figure 1: Map-Reduce Working

- The jobtracker, which coordinates the job run. The jobtracker is a Java application whose main class is JobTracker.
- The tasktrackers, which run the tasks that the job has been split into. Tasktrackers are Java applications whose main class is TaskTracker.
- The distributed filesystem, which is used for sharing job files between the other entities.

When the map function starts producing output, it is not simply written to disk. The process is more involved, and takes advantage of buffering writes in memory and doing some presorting for efficiency reasons.

The map output file is sitting on the local disk of the tasktracker that ran the map task (note that although map outputs always get written to the local disk of the map tasktracker, reduce outputs may not be), but now it is needed by the tasktracker that is about to run the reduce task for the partition. Furthermore, the reduce task needs the map output for its particular partition from several map tasks across the cluster. The map tasks may finish at different times, so the reduce task starts copying their outputs as soon as each completes. This is known as the *copy phase* of the reduce task. The reduce task has a small number of copy threads so that it can fetch map outputs in parallel. The default is five threads, but this number can be changed by setting the `mapred.reduce.parallel.copies` property.

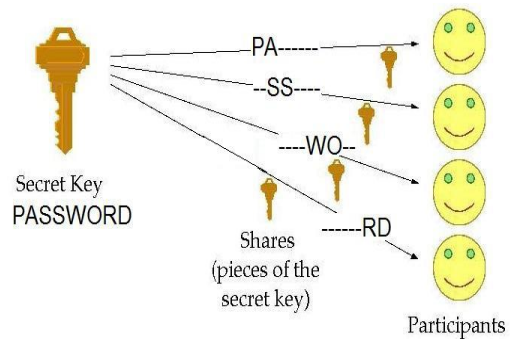
### 1.3 Shamir's Secrete Sharing

In this technique, secret key is divided in equal number of parts and distributed among the participants as shown in figure 2. The goal is to reconstruct the secret by obtaining minimum number of shares. The threshold scheme is required to reconstruct the secret that is given by  $(k, n)$ , Where  $n$  is the number of participants among which secret is distributed and  $k$  is the minimum number of participants required to

- Provides perfect security.
- It is flexible.
- It is expandable.
- Efficient distributed mechanism is used for arithmetic calculations.

### 2.3 Important Processes

Uploading: When a user uploads a file to a



storage system and stores his key in this Se Das system, he should identify the file, the key and *TTL (time-to-live)* arguments for the uploading procedure. In these codes, we assume key has been read from the file. The ENCRYPT procedure uses a common encrypt algorithm or user-defined encrypt algorithm.

Downloading: Any user who has needed permission can download data stored in the data storage system. The data should be in decrypted form before use. The whole logic is implemented Figure 2: Secrete Sharing in user's application.

In the proposed system there are multiple advantages are available. In this system we used the Hadoop technology for provide larger security of user secret or confidential data like, hacking of secret keys or passwords. The main advantage is to secure data. In this system the data is more secure than existing system. Because the data accessed by only authorized users. In this

proposed system if the hacker can try to get client confidential data, then this data made useless by destroying the private key. This type of securities provided by proposed system.

In this system the user key or password is distributed over multiple servers using Shamir's algorithm. By using this algorithm, the secret password or key is distributed over multiple server machines and using this approach the hackers can't hack the user password or secret key.

### 3. ARCHITECTURE

#### 1.3 Metadata Server (MDS)

MDS takes responsibilities of user management, server management, session management and file metadata management. Also the session can be handled using MDS.

#### Application Node

It is the client that uses the storage services provided by the service providers.

#### Storage Node

Storage node is responsible for storing files at various systems which are physically at same

or different locations. Storage node uses storage handler to handle the files.

### 4. CONCLUSION

Hadoop has been very efficient solution for companies dealing with the data in petabytes. It has solve many problems in industry related to hug data management and distributed system and it produce much more security to the database.

Data outsourcing using Shamir's secret sharing methods are compared using the Hadoop technology. The Secret Sharing methods are computationally inexpensive when compared with the normal encryption techniques. The security provided by these methods is not bounded by the computational capabilities of the existing hardware.

By taking the above points into consideration we determine that, due to the distributed nature of the hadoop system, information distribution of data is the more optimal approach for data outsourcing.

### 5. REFERENCES

- [1] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979.
- [2] Li Bai and Xukai Zou. A proactive secret sharing scheme in matrix projection method. *International Journal of Security and Networks*, 4(2):15–23, 2009.
- [3] L. Zeng, Z. Shi, S. Xu, and D. Feng, "Safevanish: An improved data self-destruction for protecting data privacy," in *Proc. Second Int. Conf. Cloud Computing Technology and*

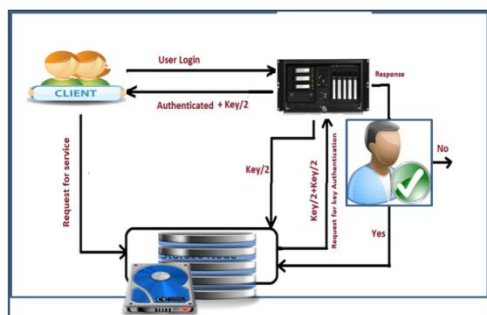


Figure 3: System Architecture





*Science (CloudCom)*, Indianapolis, IN, USA, Dec. 2010, pp. 521–528.

[4] R. Perlman, “File system design with assured delete,” in *Proc. Third IEEE Int. Security Storage Workshop (SISW)*, 2005.

[5] G. R. Blakley. Safeguarding cryptographic keys. *American Federation of Information Processing Societies Proceedings*, 48:313–317, 1979.

[6] C. Wang, Q. Wang, K. Ren, and W. Lou, “Privacy-preserving public auditing for storage security in cloud computing,” in *Proc. IEEE INFOCOM*, 2010.

[7] M. Franklin and M. Yung. Communication complexity of secure computation. *STOC*, pages 699–710, 1992.

[8] R. Gennaro. Theory and practice of verifiable secret sharing. *Ph.Dthesis, MIT*, 1995.

[9] H. Ghodosi, J. Pieprzyk, and R. Safavi-Naini. Secret sharing in multilevel and compartmented groups. *Lecture Notes in Computer Science*, 1438:367–378, 1998.

[10] J. He and E. Dawson. Multistage secret sharing based on one-way function. *Electronics Letters*, 30:1591–1592, 1994.

[11] Y. Tang, P. P. C. Lee, J. C. S. Lui, and R. Perlman, “FADE: Secure overlay cloud storage with file assured deletion,” in *Proc. SecureComm*, 2010.

[12] S. Iftene. General secret sharing based on the chinese remainder theorem with applications in e-voting. *Electronic Notes in Theoretical Computer Science*, 186:67–84, 2007.

[13] RFU Xiao, WANG Zhi-jian, WU Hao, YANG Jia-qi, WANG Zi-zhao, “How to send a Self-destructing Email, a method of self-destructing email system,” in *Proc. IEEE DOI 10.1109/BigData.Congress.2014*, pp. 304–309.

[14] Tina Miriam John, Anuradharthi Thiruvenkata Ramani, John A. Chandy, “Active Storage using Object-Based Devices”. Second International Workshop on High Performance I/O Systems and Data Intensive Computing (HiperIO’08) IEEE, 2008, pp. 472-478.

[15] K. M. Martin, J. Pieprzyk, R. Safavi-Naini, and H. Wang. Changing thresholds in the absence of secure channels. *Lecture Notes in Computer Science*, 1587:177–191, 1999.

[16] Lingfang Zeng , Shibin Chen , Qingsong Wei , and Dan Feng, ” SeDas : A Self-Destructing Data System Based on Active Storage Framework” IEEE TRANSACTIONS ON MAGNETICS, VOL. 49, NO. 6, JUNE 2013,pp:2548-2554.

[17] Shaofeng Zou, *Student Member, IEEE*, Yingbin Liang, *Member, IEEE*,



Lifeng Lai, *Member, IEEE*, and Shlomo Shamai (Shitz), *Fellow, IEEE*, "An Information Theoretic Approach to Secret Sharing", *IEEE TRANSACTIONS ON INFORMATION THEORY*, VOL. 61, NO. 6, JUNE 2015, pp:3121-3136.

[24] Tamir Tassa, Nira Dyn, and U Ci. Multipartite secret sharing by bivariate interpolation. In *33rd International Colloquium on Automata, Languages and Programming, ICALP 2006, Lecture Notes in Comput. Sci. 4052*, pages 288–299. Springer-Verlag, 2006.

[18] M.Plastoi, Curiac,D.-I, "Energy-Driven methodology for node self-destruction in wireless sensor networks", *Applied computational intelligence and informatics, 2009.SACI '09.5<sup>th</sup> international symposium*, pp:319-322.

[19] Deghaili,R.,Chehab,A.,Kayssi,A, "Trust-privacy tradeoffs in distributed systems", *Innovations in information technology[IIT],2008.IIT2008*,pp:39-43.

[20] Yu Zhang, Dan Feng, "An Active Storage System for High Performance Computing", *Advanced Information Networking and Applications 2008 AINA 2008 22nd International Conference*. pp: 644 – 651.

[21] Ron Steinfeld, Josef Pieprzyka, and Huaxiong Wang. Lattice-based threshold changeability for standard shamir secret-sharing schemes. *IEEE Transactions on Information Theory*, 53:2542–2559, 2007.

[22] D. R. Stinson, editor. *Cryptography: Theory and Practice*. CRC Press, Inc., Boca Raton, Florida, USA, 1995.

[23] Tamir Tassa. Hierarchical threshold secret sharing. *Journal of Cryptology*, 20(2):237–264, November 2009