# A Secure Data Self-Destructing Scheme in Cloud Computing

**Anna N Kurian**
*UG Student*
*Department of Computer Science & Engineering*
*MBCCET, Peermade*

**Ashna Ashraf**
*UG Student*
*Department of Computer Science & Engineering*
*MBCCET, Peermade*

**Annie Chacko**
*Head of Department*
*Department of Computer Science & Engineering*
*MBCCET, Peermade*

## Abstract

The paper presents architecture of a system that is capable of implementing: a keyless encryption method; self-destruction of data within a time frame without user's intervention; and break-glass data recovery, and in-built failover protection. It mainly aims to overcome many of the current problems within Cloud-based infrastructures, such as in the loss of private keys, and inherent failover protection. The architecture emphasis a secret share method with: an Application Platform; Proxy Servers with Routers; and a Metadata Server. These interact within a multi-cloud environment to provide a robust and secure system, and which sounds a new direction in an improved cloud computing environment. The user gets an alert about the time frame. If the user fails to collect the data, then he/she can make a request to the owner for resending the data. It aims to ensure user privacy, and reduces the potential for data loss, and also reducing denial-of-service outages within the cloud, and with in-built failover protection for stored data.

**Keywords: break-glass data recovery, failover protection, keyless encryption, multi-cloud, secret shares, self-destruct**

## I. INTRODUCTION

Distributed computing is considered as the following stride in the advancement of on-request data innovation which consolidates an arrangement of existing and new methods from research zones, for example, benefit situated structures Service Oriented Architectures (SOA) and virtualization. With the fast improvement of adaptable distributed computing innovation and administrations, it is standard for users to use distributed storage administrations to impart information to others in a companion circle, e.g., Dropbox, Google Drive and AliCloud. The shared data in cloud servers, however, usually contains client's delicate data (e.g., individual profile, financial data, health records, etc.) and needs to be very much secured. There are many security and privacy issues within Cloud-based systems that need to be addressed, as, often, existing architectures within organisations have simply been scaled-out into the Cloud, with the addition of encryption. This methodology has been very weak from many aspects, especially related to: trusted administrator access; lack of proper access control; APT; and in the loss of private keys.

Now many systems often protected with symmetric key encryption methods, in which the key is protected by a password or encrypted using public key encryption. Along with this, anyone with System Administrator can gain access to the encrypted data. The current encryption methods in the Cloud often suffer where the loss of a single encryption key can result in large-scale data loss. With the Cloud, there is always a risk of a major outage in parts of the Cloud, such as most recent ones within Azure and Amazon [2] and in many cases, this outage result to denial of service and more severely business shut down as there is no alternative methods of accessing data. Beyond this, user's privacy are usually jeopardised in the cloud as cloud service providers often without consent cache, copy and archive user's data, which can easily be retrieved, use and misuse by miscreants, competitors or court of law even when the owner seems to have deleted them. The paper proposes a secret share architecture which aims to address these problems with:

- Keyless encryption- Numerous frameworks have been ruptured by a trade-off around the departure of a private key. The putting away, handling and transmitted information inside Cloud-based framework can include breaking information into mystery offers which can be appropriated among the individuals who have rights to the information. On the off chance that any of the information components are accessed, it won't be conceivable to recoup the original information, until the other significant shares are accessible. A solid rights foundation can then actualize the uniting of the shares.
- Self-destruction- The secret sharing strategy bolsters a self-destructing information framework, where all the data and their duplicates, and in addition unscrambling keys can be destructed after a client characterize time, with no client contribution. SeDas [3][1], for instance, causes delicate data, for example, account numbers, passwords and notes, to irreversibly self-destruct, with no activity on the client's part. It is therefore appropriate in various server-based frameworks, but has an operational overhead.

All rights reserved by *www.ijste.org*

- Break-glass data recovery- The usage of secret shares supports a break-glass approach, where data can be recovered which does not comply with the required policy, and would be used in emergency cases. This has very strong applications in safety critical system or in disaster recovery.
- In-built failover protection- The utilisation of secret shares allows the striping down of data across multiple cloud-based storage systems using any k-from-n sharing policy. For example if we stripe data across five Cloud storage systems, and apply a policy of 3 from-5 then,if two of the Cloud storage systems can fail, but the original data will still be able to be recovered.
- The rest of this paper is organized as follows. In Section II, related works secure data self-destruction; cloud computing architecture; and its current method of data dissemination was carried out. We leaverged the proposed architecture in cloud based systems in Section III and also the secret sharing architecture in a multi-cloud environment and its elements. We concluded in section IV.

## II. RELATED WORKS

### *A. Secure Data Self-Destruction*

A well-known scheme for addressing the problem is the safe and secure deletion of sensitive data after expiration period when the data was used [16]. A policy graph to describe the relationship between attributes and the protection class and also proposed a policy-based secure data deletion scheme [17].The graph theory, B tree structure and key wrapping and proposed a novel approach to the design and analysis of secure deletion for persistent storage devices [18]. Concerning the properties of physical storage media, the above methods are not suitable for the cloud computing environment as the deleted data can be recovered easily in the cloud servers [19]. A data self-destructing scheme, a promising methodology which designs a Vanish system enables users to control over the lifecycle of the sensitive data. An extended Vanish system, proposed a Secure Self-Destructing scheme for electronic Data (SSDD) [20]. In the SSDD scheme, a data is encrypted into a ciphertext, which is associated and extracted to make it incomplete to resist against the traditional cryptanalysis and the brute-force attack. Then, both the decryption key and the extracted ciphertext are then distributed into a Distributed Hash Table (DHT) network to implement self-destruction after the update period of the DHT network. A lot of experiments were conducted and confirmed that the Vanish system is vulnerable to Sybil attacks by using the Vuze DHT network [21]. So the security of the SSDD scheme is also questionable.

To address this problem, a SeDas system, which is a novel integration of cryptographic techniques with active storage techniques [3]? Then leveraged the DHT network and identity-based encryption (IBE) [22] and proposed an IBE-based Secure Self-destruction (ISS) scheme [19]. In order to protect the confidentiality and privacy security of the composite documents within the whole lifecycle in cloud computing, applied the ABE(Attribute Based Encryption) algorithm to propose a secure Self-destruction scheme for composite Documents (SelfDoc) [23][24]. Recently, employed IDentity-based Timed-Release Encryption (ID-TRE) algorithm [10] and the DHT network and proposed a full lifecycle privacy protection scheme for sensitive data (FullPP), which is able to provide full lifecycle privacy protection for user's sensitive data by making it unreadable before a predefined time and automatically destructed after expiration [7]. The main aim of the above schemes is that they respectively combine different cryptographic techniques with the DHT network to provide fine grained data access control during the lifecycle of the protected data and to implement data self-destruction after expiration. However, using of the DHT network will result in the fact that the lifecycle of the protected data is limited by the update periods of the nodes in the network. How to achieve user defined data lifecycle and secure self-destruction is a worth exploring problem?

### *B. Secret Sharing Schemes*

In Perfect Secret Sharing schemes (PSS), Shamir gives a decent case of a flawless secret sharing [4]. Its efficiency is measured by two parameters: if, and only if, t-1 shares, give positively no data in regards to the hidden secret, and when the proportion of the length of the key to the length of each of the shares known as data rate, is equivalent to 1.In Information Dispersal Algorithm (IDA) Rabin suggested splitting a secret S into n pieces such that, one can obtain the secret only if $k < n$ of these pieces are available, where k is the threshold. Here, each secret $S_i$, $i \leq n$, is of size $|S|/k$, where $|S|$ is the size of the secret. The total sizes of all the secrets are:

$$(n/k) \times |S| \qquad (1)$$

The storage complexity of a secret sharing system can be significantly reduced in comparison to Shamir perfect secret sharing (PSS) scheme. But, the security flaw is that, if the data exhibits some pattern frequently, and that the attacker gets hold of $m < k$ slices, there are great possibilities for him accessing the secret S. Both split and combine algorithms operate by performing matrix multiplication over the input.

Krawczyk's Computational Secret Sharing (CSS) [5] proposed the Computational Secret Sharing (CSS) technique, secret sharing made short, which combines Rabin's IDA with Shamir's PSS. Data is first encrypted with a randomly generated key, using a symmetric encryption algorithm. Then this data is split into n fragments using Rabin's IDA with a threshold t configured. In this case, the scheme is t times more efficient than Shamir's PSS. The final step is to use Shamir's PSS to produce the shares of the randomly generated symmetric key (which is typically of the order of 64 to 256 bits) and then give one share and one fragment to each shareholder. A related approach, known as AONT-RS, applies an All-Or-Nothing transform (AONT) to the

All rights reserved by *www.ijste.org*

data as a pre-processing step to the IDA. AONT guarantees that any number of shares less than the threshold is insufficient to decrypt the data.

### *C. Keyless Encryption*

We observed that Attribute-Based Encryption is one of the important applications of fuzzy identity-based encryption [8]. ABE comes in two flavors called KP-ABE [9][11] and Ciphertext-Policy ABE (CP-ABE) [12][13]. In CP-ABE, the ciphertext is associated with the access structure while the private key contains a set of attributes. The drawback of the scheme is that security proof was only constructed under the generic group model. In KP-ABE, the idea got reversed: the ciphertext contains a set of attributes and the private key is related to the access structure. In this, when a user made a secret request, the trusted authority determined which combination of attributes must appear in the ciphertext for the user to decrypt the data. Instead of using the Shamir secret key technique [14] in the private key, this scheme used a more generalized form of secret sharing to enforce a monotonic access tree. A combining technique of KP-ABE, proxy re-encryption, and lazy re-encryption which allows the data owner to delegate most of the computation tasks involved in fine-grained data access control to the untrusted cloud servers without disclosing the underlying data contents [15].

The Key-Policy Attribute Based Encryption with Time Specified Attributes (KP-TSABE) scheme [6] each data item can be associated with a set of attributes and every attribute is associated with a specification of time interval, Decryption Attribute Time Interval, DATI, denoting that the encrypted data item can only be decrypted between the specified date and it will not be recoverable before and after that day. The data owner encrypts his/her data to share with users in the system, in which every users key is associated with an access tree and each leaf node is associated with a time instant. The access tree of each user can be defined as a unique logical expression over these DATI attributes to reflect the data item authorized to the user. In order to decrypt the ciphertext successfully, the valid attributes should satisfy the access tree where the time instant of each leaf in the user's key should belong to the DATI in the corresponding attribute in the ciphertext. Question arises here is that what happens when the key is lost? How the data become recoverable?

## III. PROPOSED ARCHITECTURE

Numerous frameworks frequently secure with symmetric key encryption strategies, where the key is ensured by a secret key or encoded utilizing open key encryption. Alongside this, anyone with System Administrator get to can access the encoded content. The present encryption strategies in the Cloud regularly endure where the passing of a solitary encryption key can bring about substantial scale information misfortune. Hence our proposed architecture is described which sounds out the keyless encryption; self-destruction; emergency data recovery.

### *A. Elements*

- Data Owner- Data Owners function is to determine the access structure, encode secrets, sends secrets to the main multi-cloud proxy server for distribution to multi-cloud service providers, and keeping the secret shares when recovered.
- Main Multi-Cloud Proxy Server with Router- This splits and distributes encoded shares to multi-cloud based on pre-determined access structure and manages the fail-over protection of shares. This gathers shares and reconstruct secret as well manages break-glass data recovery.
- Cloud server- This includes the functionality of user management, server management, session management, and file management [3].
- Data User- User gets an alert about the time frame set by the owner. If they fails to retrieve the data they have the provision to request the corresponding owner to resend the data.

At the Application platform, the data owner determines n and t values using both calls up the application to be used, selects a suitable algorithm. After a successful sign in to the system and access level determined. The encoded data is sent to the local main multi-cloud proxy server with router for dissemination to the CSPs.

The proxy splits the encoded data according to a secret-sharing scheme determined access structure, and distributes each share over the Internet to different CSPs. The retrieval of data is similar to the storage process as the metadata server helps to keep track of the siblings of the shares. The proxy retrieves enough corresponding shares from the cloud service providers. The retrieval process involves authentication to the cloud service providers. The retrieved shares are sent back to the application platform software, which decodes them and verifies their authenticity before reconstructing the data. The system is capable of a break-glass data recovery through the local multi-cloud proxy server in a case of emergency after, which a clean-up should be performed at the end of the activities for record purposes. The design incorporates unique feature in a multi-cloud environment as it uses secret sharing scheme to implement keyless encryption. This is done by breaking the secret into chunks called (k-out-of-n) threshold in such a manner that less than k shares cannot recover the secret, therefore using it for data distribution in object storage system. This is also used to implement safety destruct with equal divided shares. The incorporation of a Self-Destructive system solves the problem of cloud user's privacy as there is no other means by which the user's data can be accessed, copied, cached or used without the data owner's consent within a predetermined time-frame as all data and their copies become destructed after a user-specified time, without any user intervention [3].The self-destructive system defines two modules: a self-destruct method object; and survival time period for each secret key part. In this case, a secret sharing algorithm is used to

All rights reserved by *www.ijste.org*

implement share distribution in object storage system so as to ensure safe destruct with equally divided shares. Considering active storage framework, object-based storage interface will be used to store and manage the equal divided shares. The use and implementation of a threshold system in cloud services are deliberate act towards implementing a failover protection in the model. In a normal circumstance all the service providers are used in share storage as well as secret reconstruction, but in an extreme desperate situation, 2-out-of-5 can be made redundant.

That is if 2-out-of-5 CSPs fails, data/secret storage and reconstruction are still possible. The use of the local multi-cloud proxy server and the sub-routers (1-5) were also deliberate as they are for the implementation of a break-glass data recovery. With the sub-routers and the second multi-cloud server, a route is established to and from all the CSPs and having decided for a 3-out-of-5 access structure only 3-out-of-5 CSPs are required to store and reconstruct the secret at an emergency situation. By this feature, the concept of total business shut down as in [2] or denial of service may not exist in using this model, though the number of CSPs required is dependent on the secret sharing algorithm of choice in times of secret reconstruction. There is also an incorporation of a Break-Glass data recovery system implemented using one of the Proxy Servers. An access to Multi-Cloud proxy Server entails an access to CSPs 1, 3 and 5 and this in turn ensures a quick recovery of shares in order to reconstruct the secret as it is a quick link to all other CSPs and the access structure, such access ensures the possibility of reconstructing the secret in an emergency situation. This is an important feature as there could be a period of cloud outage as stated in [2], and in such situation, data recovery could be done from 3-out-of-5 Cloud service providers being used for data storage. Hence, if 2-out-of-the-5 cloud service providers fail, data recovery is still possible in such an extreme condition.

### *B. Architectural Model*

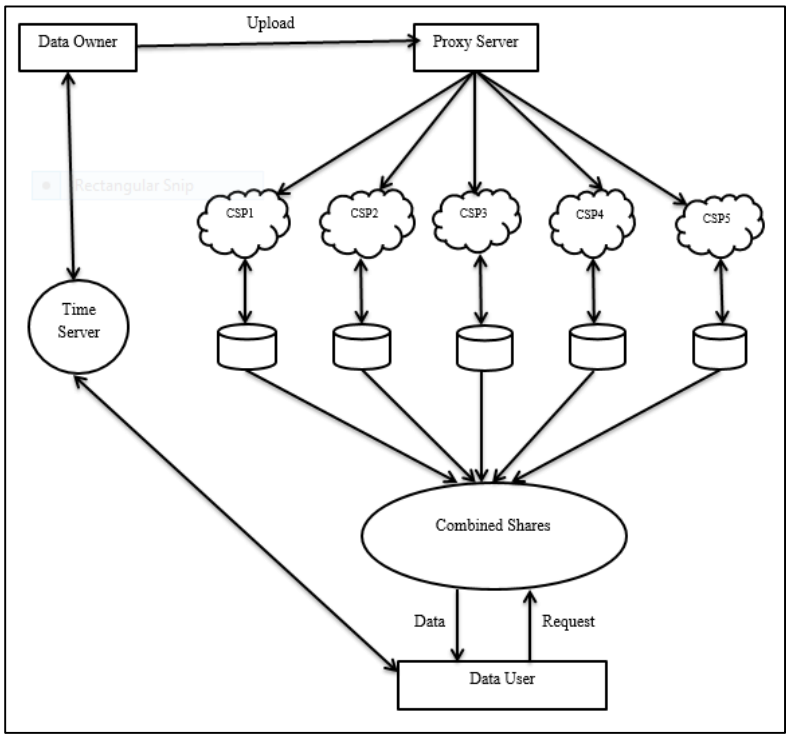


Fig. 1: Proposed architecture for secret sharing in multi-cloud environment

### *C. Implementation*

The secret sharing schemes have been used successfully in data splitting and recovery, thereby providing data security in a keyless manner. Our proposed architecture mainly focuses on the following:

1) Fast and efficient data/key distribution to multi-cloud service providers.
2) Keyless encryption and by extension data security.
3) Provide data owner's privacy by implementing SeDas, as it meets all the privacy-preserving goals [3].
4) SeDas supports security erasing files and random storage in drives (Cloud, HDD or SSD) respectively [3].
5) Backup operational mode in which the function of 5 CSPs can be assumed by 3 CSPs when 2out-of-the-5 CSPs become unavailable either through failure or scheduled down time.
6) Break-glass data recovery.

All rights reserved by *www.ijste.org*

## IV. CONCLUSION

With the development of cloud computing and wide use of mobile networks, cloud services are becoming more and more important for people's life. Data privacy has become increasingly important in the cloud environment. There are many weaknesses within Cloud-based infrastructure, including risks around the usage of public and private key encryption. The implemented architecture provides a new method of integrating privacy into the data infrastructure, along with integrating a self-destructive system, with break-glass data recovery and in-built failover protection. , though, that there is a performance impact with large file, and related to the number of shares used. In our implementation we have used AES (Advanced Encryption Scheme) for encryption and decryption. However the proposed architecture is superior to other existing system.

## REFERENCES

[1] Rao, C., Rodi, P., Palande, A., & Bhusari, V. (2015). SEDAS: A Self-Destructing Data System Based on Shamir's Secret Sharing Algorithm. International Journal Of Scientific Research And Education, 3(03).
[2] Srinivasan, S. (2014). Building trust in cloud computing: Challenges in the midst of outages. Proceedings of Informing Science & IT Education Conference (InSITE) 2014 (pp. 305312). Retrieved from http://Proceedings.InformingScience.org/InSITE2014/InSITE1 4p305-312Srinivasan0544.pdf
[3] Zeng, L., Chen, S., Wei, Q., & Feng, D. (2012). Sedas: a self-destructing data system based on active storage framework. In APMRC, 2012 Digest(pp. 1-8). IEEE.
[4] A. Shamir, "How to share a secret," Communications of the ACM, vol. 22, no. 11, p. 612–613, 1979.
[5] H. Krawczyk, "Secret Sharing Made Short," in Proceedings of the 13th Annual International Cryptology Conference on Advances in Cryptology, 1993.
[6] A secure data self-destructing scheme in cloud computing Jinbo Xiong, Student Member, IEEE, Ximeng Liu, Student Member, IEEE, Zhiqiang Yao, Jianfeng Ma, Qi Li, Kui Geng, and Patrick S. Chen IEEE transactions on cloud computing vol:pp no:99 year 2014
[7] J. Xiong, F. Li, J. Ma, X. Liu, Z. Yao, and P. S. Chen, "A full lifecycle privacy protection scheme for sensitive data in cloud computing," Peerto-Peer Networking and Applications. [Online]. Available: http://dx.doi.org/10.1007/s12083-014-0295-x
[8] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in Advances in Cryptology–EUROCRYPT 2005, ser. LNCS, vol. 7371. Springer, 2005, pp. 457–473.
[9] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in Proceedings of the 13th ACM conference on Computer and Communications Security. ACM, 2006, pp. 89–98.
[10] A. F. Chan and I. F. Blake, "Scalable, server-passive, useranonymous timed release cryptography," in Proceedings of the International Conference on Distributed Computing Systems. IEEE, 2005, pp. 504–513.
[11] Q. Li, J. Ma, R. Li, J. Xiong, and X. Liu, "Large universe decentralized key-policy attribute-based encryption," Security and Communication Networks, 2014. [Online]. Available: http://dx.doi.org/10.1002/sec.997
[12] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in Proceedings of the 28th IEEE Symposium on Security and Privacy. IEEE, 2007, pp. 321– 334.
[13] L. Cheung and C. C. Newport, "Provably secure ciphertext policy abe," in Proceedings of the 14th ACM conference on Computer and communications security. ACM, 2007, pp. 456– 465.
[14] A. Shamir, "How to share a secret," Communications of the ACM, vol. 22, no. 11, pp. 612–613, 1979.
[15] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in Proceedings of the 29th IEEE International Conference on Computer Communications. IEEE, 2010, pp. 1–9.
[16] J. Reardon, D. Basin, and S. Capkun, "Sok: Secure data deletion," in Proceedings of the 34th IEEE Symposium on Security and Privacy. IEEE, 2013, pp. 1–15.
[17] C. Cachin, K. Haralambiev, H.-C. Hsiao, and A. Sorniotti, "Policy-based secure deletion," in Proceedings of the ACM Conference Computer and Communications Security. ACM, 2013, pp. 152–167.
[18] J. Reardon, H. Ritzdorf, D. Basin, and S. Capkun, "Secure data deletion from persistent media," in Proceedings of the 2013 ACM Conference on Computer and Communications Security. ACM, 2013, pp. 271–284.
[19] J. Xiong, Z. Yao, J. Ma, F. Li, and X. Liu, "A secure self-destruction scheme with ibe for the internet content privacy," Chinese Journal of Computers, vol. 37, no. 1, pp. 139–150, 2014.
[20] G. Wang, F. Yue, and Q. Liu, "A secure self-destructing scheme for electronic data," Journal of Computer and System Sciences, vol. 79, no. 2, pp. 279–290, 2013.
[21] S. Wolchok, O. S. Hofmann, N. Heninger, E. W. Felten, J. A. Halderman, C. J. Rossbach, B. Waters, and E. Witchel, "Defeating vanish with low-cost sybil attacks against large dhts," in Proceedings of the 17th Annual Network and Distributed System Security Conference, NDSS. ISOC, 2010, pp. 1–15.
[22] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," SIAM Journal on Computing, vol. 32, no. 3, pp. 586–615, 2003.
[23] J. Xiong, Z. Yao, J. Ma, X. Liu, and Q. Li, "A secure document self-destruction scheme: An abe approach," in Proceedings of the 15th IEEE International Conference on High Performance Computing and Communications. IEEE, 2013, pp. 59–64.
[24] J. Xiong, Z. Yao, J. Ma, F. Li, X. Liu, and Q. Li, "A secure self-destruction scheme for composite documents with attribute based encryption," Acta Electronica Sinica, vol. 42, no. 2, pp. 366–376, 2014.

All rights reserved by *www.ijste.org*