International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 November 2023*

Evaluation of Proposed Secrete Sharing System based on Encryption for Distributed Cloud Data Security

Mrs. Kruti Patel [1] Dr. Kamaljit Lakhtaria [2]
[1] PhD Scholar, Dept. of Computer Science, Gujarat University, Ahmedabad, Gujarat, India-380009.
kruti.patel.k@gmail.com
[2] Associate Professor, Dept. of Computer Science, Gujarat University, Ahmedabad, Gujarat, India-380009.
kamaljit.ilakhtaria@gmail.com

Abstract: The Public Key Infrastructure (PKI) is deteriorating, partly due to a lack of comprehensive understanding of encryption mechanisms and also due to flaws in its execution. This paper presents a data storage methodology utilizing secret sharing techniques, which could address the challenges associated with PKI while accommodating innovative architectural designs incorporating features like automated failover and emergency data retrieval. The document introduces a framework which facilitating a cloud-based infrastructure with inherent privacy measures and failover capabilities. To evaluate the performance impact of secret sharing architecture, the paper describes a series of experiments exploring the overhead of this method.

This paper introduces a system architecture capable of implementing: a keyless encryption approach; automatic data expiration within a predefined timeframe; and emergency data retrieval with integrated failover mechanisms. It seeks to address various issues encountered in current Cloud-based infrastructures, such as key loss and inherent failover challenges. To evaluate the most suitable secret sharing method for this architecture, the document describes a variety of experiments examining the performance implications of the most pertinent secret sharing techniques.

Keywords: *secret shares, distributed cloud, key management, secrete sharing, cloud computing, encryption*

1. INTRODUCTION

Cloud Computing has undergone a significant transformation within Information Technology, notably seen in the transition from private networks to virtualized ones, as well as from private cloud infrastructures to public ones. However, these transitions have not substantially altered security practices, often resorting to the addition of encryption keys or the adoption of multi-factor authentication methods. Moreover, concerns persist regarding the public cloud's susceptibility to large-scale outages and other security vulnerabilities.

A significant risk in migrating existing systems to the Cloud lies in the reliance on PKI (Public Key Infrastructure) for data security. This reliance introduces potential vulnerabilities and a lack of comprehensive understanding of encryption techniques. Many encryption methods, including the RSA algorithm, utilize key pairs to safeguard symmetric keys used for data encryption in the Cloud. While these methods are generally considered secure against major vulnerabilities, the loss of the private key poses significant data loss risks. Public cloud systems are particularly susceptible to data loss due to private key compromise, exacerbated by the proliferation of Advanced Persistent Threats (APTs), certificate cracking, and insider threats.

The future of the Internet demands built-in data protection mechanisms, whether through robust protective measures like sticky policies or through fragmentation techniques that secure data fragments and enforce strict policies for data reconstruction, without relying solely on traditional encryption methods.

2. SECRETE SHARING METHODOLOGY

Securing data in Cloud-based storage systems presents a significant hurdle[3], with existing architectures often falling short in adequately managing access privileges to such data. The insider threat, particularly from individuals like System Administrators, coupled with unforeseen implementation issues, undermines the integrity of Cloud-based systems on multiple fronts. While a secret sharing scheme can offer high efficiency, it lacks robust security. A truly secure secret sharing scheme should distribute shares in a manner where possessing fewer than shares provides no additional insight into the secret compared to having zero shares. For instance, consider a secret sharing scheme where the secret phrase "password" is divided into the following shares (in addition to RS code shares):

“pa------”, “--ss----”, “----wo--”, “------rd”

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 November 2023*

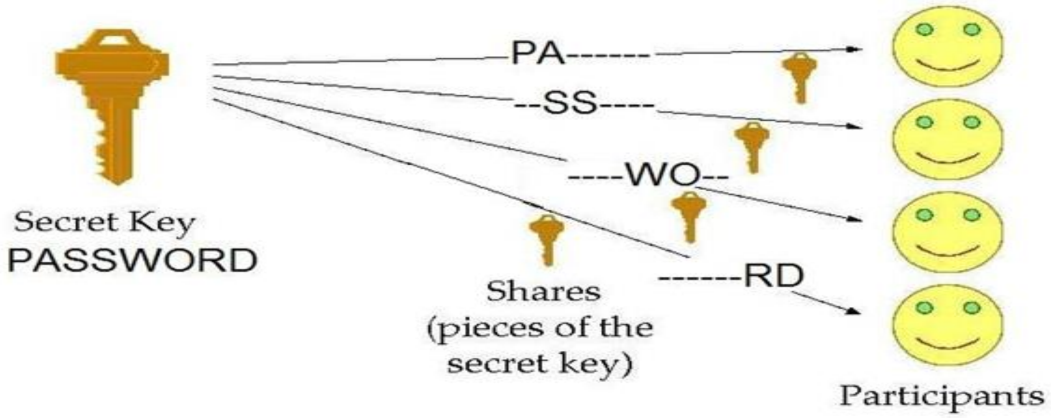

Figure 1: secrete key sharing among participants

A person with 0 of these shares might only discern that the password comprises eight letters, necessitating them to sift through 208 billion combinations (26_8) to guess the password. Conversely, with one share, they would only need to consider 308 million combinations (to guess solely the six letters—26_6), and so on as more shares become available. RS code falls short as a secure secret sharing scheme since it permits a player with fewer than secret-shares to partially identify some of the original data.

3. SECRET SHARING SCHEMES ADVANTAGES

Shamir's Polynomial Secret Sharing (PSS)[2] presents a promising solution for secret sharing in cloud storage, offering numerous advantages:

1. Secure: Individuals possessing fewer than required shares gain no additional insight into the secret compared to those with zero shares, ensuring robust security.
2. Extensible: Even with a fixed value of shares, new shares can be dynamically introduced or removed without impacting existing shares, enhancing flexibility and scalability.
3. Dynamic: The polynomial can be modified, allowing for the creation of new shares without altering the original secret, facilitating efficient management and updates.
4. Flexible: In organizational settings where hierarchy plays a crucial role, it is feasible to allocate varying numbers of shares to each participant based on their importance [5], providing a flexible and tailored approach to access control.

4. SECURITY LIMITATIONS OF SECRET SHARING SCHEMES

Recent advancements in information and communication technology infrastructure have led to a rapid expansion of electronic data exchange. Consequently, both public and private institutions, along with various industries, frequently outsource vast electronic databases to storage facilities. Cloud computing technology enables users to interact with these centers without requiring knowledge of their internal workings. However, centralizing all data in one location creates a single point of failure, triggering concerns about privacy and availability, particularly regarding disaster preparedness and recovery. Secret sharing, a cryptographic technology, offers a solution to address both privacy and availability concerns simultaneously [11].

Nevertheless, Dautrich and Ravishankar's study [12], titled "Security Limitations of Using Secret Sharing for Data Outsourcing," exposes the vulnerabilities of relying solely on secret sharing. They refute claims made by previous works [13], [14], [15] suggesting that the security of a scheme remains intact as long as a prime p and a vector X used by the secret sharing algorithm are kept private. Instead, they describe and implement an attack that reconstructs all secret data when only $k + 2$ secrets are known initially. Their experiment successfully recovered a hidden 256-bit prime for $k \leq 13$ servers or an 8192-bit prime for $k \leq 8$ in under 500 seconds.

Furthermore, Tompa and Woll [16], in their paper titled "How to Share a Secret with Cheaters," have identified a vulnerability in the Shamir threshold scheme, exposing it to potential attacks by cheaters. They analyze the impact of an active adversary who masquerades as a participant but intentionally submits a false share during the reconstruction phase. For instance, if a participant Pi submits a false share λi instead of the correct share $f(xi)$, it prevents an honest participant from discovering the correct secret. This failure to detect the incorrect reconstruction also deprives other participants of the opportunity to realize the error. Consequently, the adversary can exploit this situation to learn the correct secret by leveraging knowledge of $f(xi)-\lambda i$ [18].

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 November 2023*

5. PROPOSED SYSTEM ARCHITECTURE

The proposed method will build a more reliable, decentralized light weight key management technique with secret sharing with fragmented original data which provides more efficient data security in cloud systems with validation and renewal of shares.

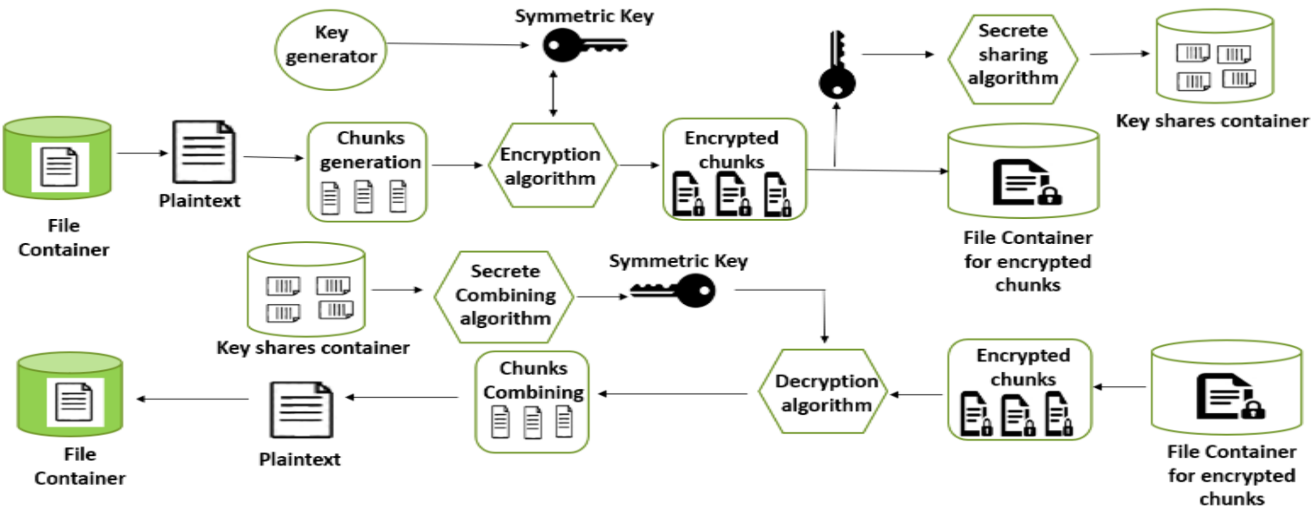

Figure 2: Proposed system share generation and Share recovery

As per the diagram of above proposed system initially user's file will be generated in to number of chunks. Then with the use of symmetric Key which is already generated by Key generator Encryption performed. An AES-256 algorithm performed on each generated chunk. Then all of those chunks will be encrypted. All Encrypted chunks will be stored in storage container. Encrypted key gets converted into multiple shares through secrete sharing algorithm. Each share is stored on different data center of cloud providers. Hence user's file secretly stored with cloud.

When user demands to get original stored file then first of all shares are getting combined and generate symmetric decryption key. Then each Encrypted chunks are assigned to decryption algorithm. Then AES-256 decryption algorithm will decrypt chunks. Afterwards Chunks are combined through merging algorithm. Then user will get original file through chunks Combination. So, this is the way how secrete key will be stored in fragmentation and encryption for data security as well. Key share Process time involves time taken to split the file into chunks using a pre-defined chunk size, fragment encryption time, key share creation and writing times while Key share Recover time involves time taken to recover key shares from folders, key recreation time, fragment decryption and file recombination times.

6. TESTS RESULTS AND EVALUATIONS

Three different sets of experiments were performed: The first experiment was based on share creation and recovery with respect to different share policy and different file sizes. The second experiment is based on total process time and recover time of file with respect to varied size fragments and also having fixed sized fragments. The third experiment is taken for key share generation process and recovery process time with respect to varied size fragments and also having fixed sized fragments. In file sharing, files of different sizes are created into share and stored in folders. When the files are needed, the several shares are recovered from the folders and the file recreated. Each file involved in the process is created into shares using M-out-of-N threshold secret sharing scheme and the shares stored in folders. While in key sharing files of different sizes are broken into chunks; each chunk is encrypted using AES of 256-bits key length then stored in folder, the encryption key is thereafter shared, stored in folders as well.

When the files are needed, the shares are recovered from the folders for each key based on policy and the key recreated, using each key to decrypt a chunk as retrieved from the folder and the file recombined. The issue of confidentiality and integrity in the use of secret sharing scheme has been validated by many works in secret sharing schemes such as Abdallah and Salleh [27], Buchanan et al. [28]. Since proposed scheme concentrated on data availability. Here time may varied while taking result due to different data center location which automatically choose by proposed implemented system.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 November 2023*

In Experiment One, Test results 1 and 2 are taken. In Test result 1, the time taken is calculated to create shares of data against the 2 from 5, 3 from 5, and 4 from 5 share policies as shown in Table 1. In Test result 2, the time taken is calculated to recover shares of data against the 2 from 5, 3 from 5, and 4 from 5 share policies as shown in Table 2. Figures 3 and 4 show a normal curve with an increasing size of Threshold (M) and file size.

In Experiment Two, Test results 3 and 4 are taken. In Test result 3, the total overhead time cost is measured with respect to File Sizes in the 2 from 5 share policy for a 1 KB fragment size as shown in Table 3. In Test result 4, the total overhead time cost is measured with respect to File Sizes in the 2 from 5 share policy for a fixed size fragment as shown in Table 4. For these experiments, we have selected a 15% fixed fragment size of the File size. Figure 5 shows that the curve increases very slightly until the file size reaches 1 MB, after which it starts gradually increasing with respect to file size. Figure 6 shows that the curve increases very slightly until the file size reaches 10 MB, after which it starts gradually increasing with respect to file size. Although all times can't be fixed because there is no direct relation between file size and key share policy.

In Experiment Three, Test results 5 and 6 are taken. In Test result 5, the total overhead time cost is measured with respect to key share generation and recovery in the 2 from 5 share policy for a 1 KB fragment size as shown in Table 5. In Test result 6, the total overhead time cost is measured with respect to key share generation and recovery in the 2 from 5 share policy for a fixed size fragment as shown in Table 6. For these experiments, we have selected a 15% fixed fragment size of the File size. Figure 7 shows the Key Share Creation and Recovery time curve using a 1 KB fragment in the 2 from 5 share policy. It suddenly starts increasing after the file size reaches 1 MB. Figure 8 shows the Key Share Creation and Recovery time curve using a fixed fragment size in the 2 from 5 share policy. It gradually increases with stable file size increments. The time taken with a fixed size fragment is different from that with a varied size fragment. Although it doesn't relate directly, for analysis, we can refer to the plotted graphs.

Test Results 1: Time taken to Create shares of data against Share Policy

	Policy:	2 from 5	3 from 5	4 from 5
S/N	File Size	Share Creation Time (Sec)	Share Creation Time (Sec)	Share Creation Time (Sec)
1	1 KB	0.107219	0.109521	0.248711
2	10 KB	0.812554	1.023654	1.365488
3	100 KB	1.584442	2.052688	2.458622

Table 1: Share creation against policy

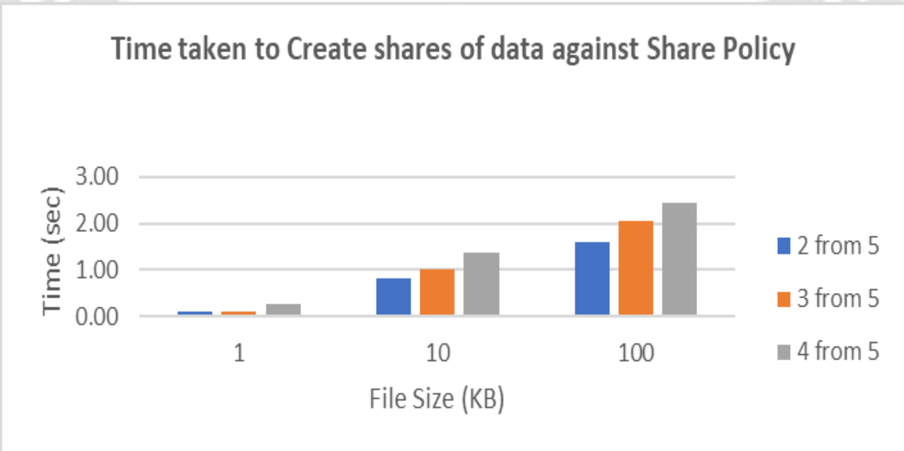

Figure 3: Time taken to Create share against Policy

Test Results 2: Time taken to Recover shares of data against Share Policy

	Policy:	2 from 5	3 from 5	4 from 5
S/N	File Size	Share Recovery	Share Recovery	Share Recovery

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 November 2023*

		Time (Sec)	Time (Sec)	Time (Sec)
1	1 KB	0.004251	0.008421	0.018125
2	10 KB	0.062568	0.096584	0.102548
3	100 KB	0.072541	0.095647	0.135412

Table 2: Share Recovery against policy

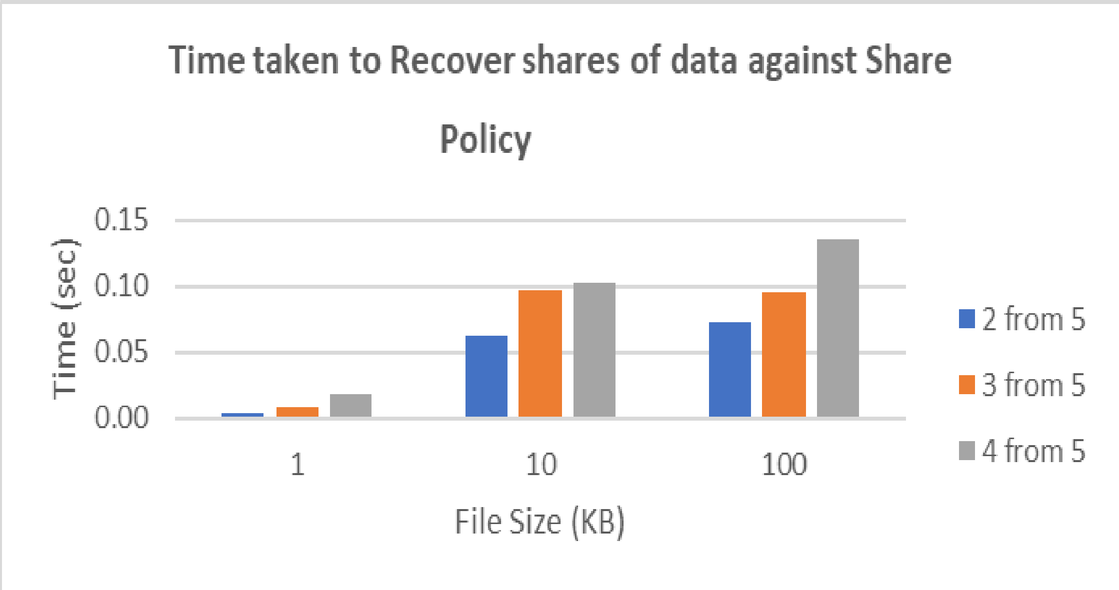

Figure 4: Time taken to Recover share against Policy

Test Results 3: File Sizes against Time Taken in 2 from 5 share policy for 1 KB fragment size

S/N	Policy:	2 from 5				
		1 KB fragment size				
	File Size	File Split Time (sec)	Fragment Encrypt Time (sec)	Fragment Decrypt Time (sec)	File Combine Time (sec)	OverHead Cost (sec)
1	1 KB	0.008541	0.019475	0.015552	0.011746	0.055314
2	10 KB	0.045699	0.032558	0.458213	0.253680	0.790150
3	100 KB	0.253684	0.325471	0.632547	0.325551	1.537253
4	1 MB	2.362584	4.458127	6.752510	2.352541	15.92576
5	10 MB	70.20558	123.2654	141.2557	70.55255	405.2793
6	100 MB	210.2565	2010.255	785.2554	1425.559	4431.326

Table 3: Varied file sizes using 1KB fragment size in 2 from 5 share policy

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 November 2023*

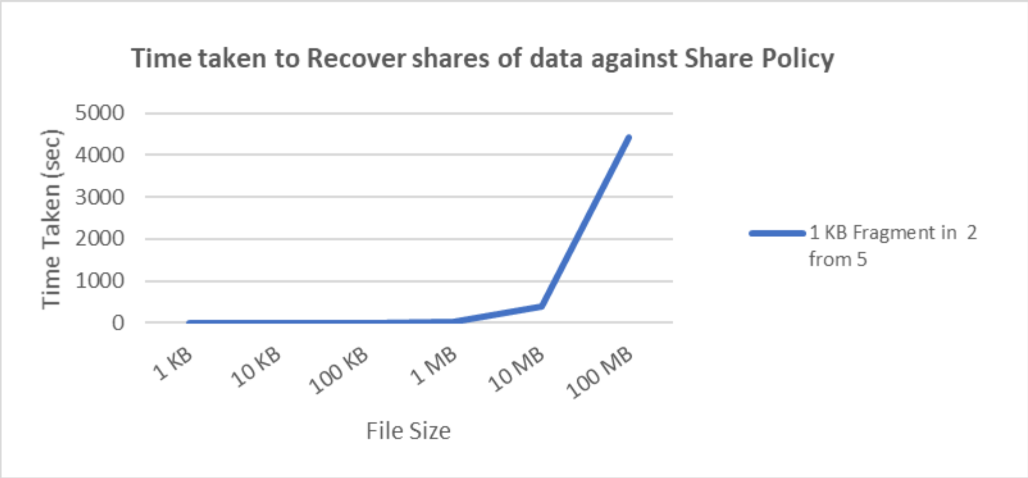

Figure 5: Time taken to Create share against Policy for 1 KB size fragment

Test Results 4: File Sizes against Time Taken to recombine file in 2 from 5 share policy for fragment size 15% of file size

S/N	Policy:	2 from 5				
		Fragment size 15 % of File size				
	File Size	File Split Time(sec)	Fragment Encrypt Time(sec)	Fragment Decrypt Time(sec)	File CombineTime (sec)	OverHead Cost(sec)
1	1 KB	0.002158	0.005143	0.014258	0.001699	0.023258
2	10 KB	0.038561	0.007584	0.045813	0.002537	0.094495
3	100 KB	0.042568	0.008541	0.054124	0.005682	0.110915
4	1 MB	0.069854	0.009569	0.059841	0.005841	0.145105
5	10 MB	0.036952	0.085241	0.084126	0.015487	0.221806
6	100 MB	0.352684	0.985412	0.745812	0.352658	2.436567

Table 4: Varied file sizes using equal number of fragments in 2 from 5 share policy

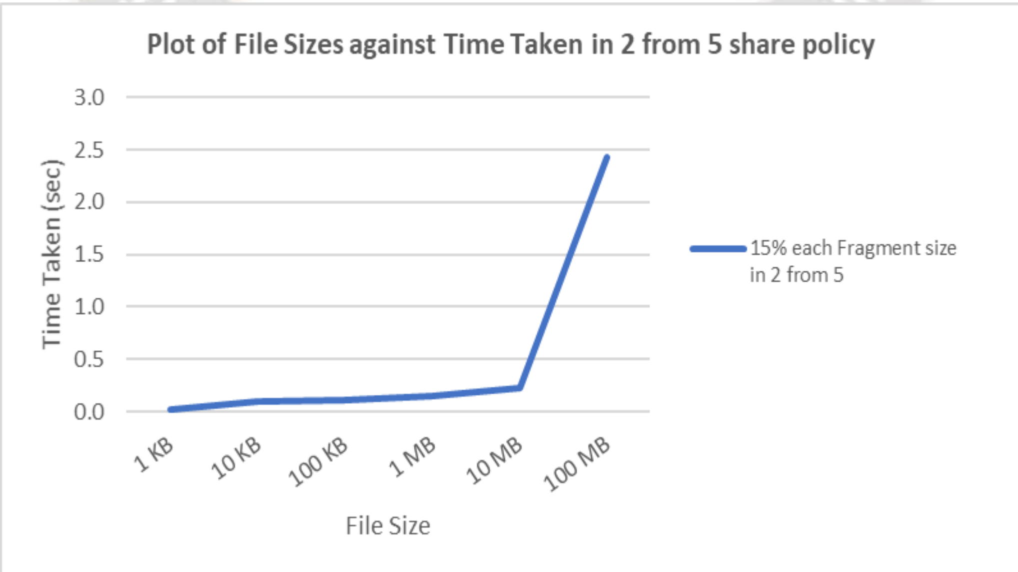

Figure 6: Varied file sizes using equal number of fragments in 2 from 5 share policy

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 November 2023*

Test Results 5: File Sizes against Time Taken to process and recover secret key using 2 from 5 key share policy for 1 KB fragment size

	Policy:	2 from 5				
S/N		1 KB fragment size				
	File Size	Key Share Create Time(sec)	Key Share storage Time(sec)	Key Share Recall Time(sec)	Key Share Recovery Time(sec)	OverHead Cost (sec)
1	1 KB	0.002541	0.008743	0.002542	0.007854	0.021680
2	10 KB	0.035851	0.054712	0.014825	0.008541	0.113930
3	100 KB	0.425883	0.412548	0.029854	0.009528	0.877813
4	1 MB	3.458741	8.254621	0.068747	0.012548	11.79466
5	10 MB	102.2568	218.2568	0.698511	0.265841	321.4780
6	100 MB	1254.257	1842.256	16.36528	2.365841	3115.244

Table 5: Key Share Creation and Recovering using 1KB fragment in 2 from 5 share policy

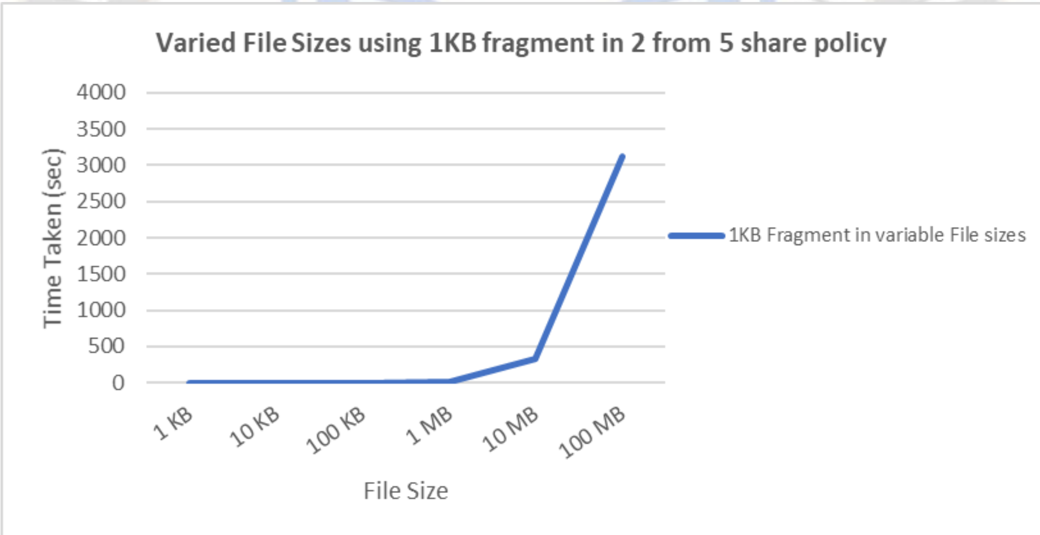

Figure 7: Key Share Creation and Recovering using 1KB fragment in 2 from 5 share policy

Test Results 6: File Sizes against Time Taken to process and recover secret key using 2 from 5 key share policy for Fragment size 15 % of File size

	Policy:	2 from 5				
S/N		Fragment size 15 % of File size				
	File Size	Key Share Create Time(sec)	Key Share storage Time(sec)	Key Share Recall Time(sec)	Key Share Recovery Time(sec)	OverHead Cost(sec)
1	1 KB	0.004526	5.258413	0.003656	0.001259	5.267854
2	10 KB	0.006854	5.895413	0.006584	0.006395	5.915246
3	100 KB	0.007581	6.784599	0.007854	0.007854	6.807888

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 November 2023*

4	1 MB	0.007965	7.254699	0.008541	0.008954	7.280159
5	10 MB	0.008257	7.659842	0.036987	0.004854	7.709940
6	100 MB	0.008699	8.526941	0.054781	0.006987	8.597409

Table 6: Varied file sizes using equal number of fragments Size in 2 from 5 share policy

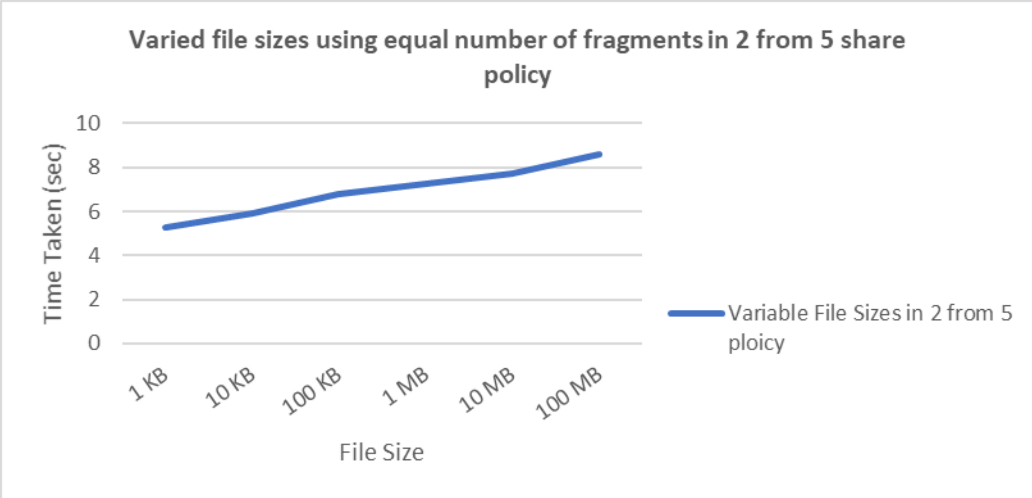

Figure 8: Varied file sizes using equal number of fragments in 2 from 5 share policy

7. CONCLUSION

In all the findings presented, it is clear that utilizing a fragmented secret sharing system is the superior choice for managing big data infrastructure compared to using a threshold secret sharing scheme alone. The latter has proven impractical for scaling large data infrastructure due to the inherent properties of finite field arithmetic. The objective of the experiment is to identify all factors that contribute to performance overhead, thereby compromising overall system performance in both File and Key Sharing methods. As we intend to apply these methods further in both network and cloud environments, we will focus on eliminating the identified factors that contribute to performance overhead. This approach has demonstrated scalability with big data infrastructure.

Experiments conducted using secret sharing schemes have demonstrated resilience in the face of failures, as not all hosts are required to reconstruct data after splitting. However, a significant drawback remains the impact of latency on performance. This issue is exacerbated as data size increases and the distance between hosts grows, thus leading to our research. Lessons learned indicate that using the Key Share method rather than the Data Share method, in conjunction with an appropriate fragment and share policy, is the only way to scale large data infrastructure.

With these lessons and validations, we aim to eliminate all factors identified as capable of adding substantial overhead to the system. Fabian and Fabian [22], Ermakova and Fabian [23], and Alsolami and Boult in [37] all argue that the secret sharing scheme is suitable for data sharing but failed to demonstrate its capability to maintain production when file sizes increase exponentially, thus limiting its application in large-scale data infrastructure. By applying this thesis' evaluation framework on scalability, as defined above, the overall evaluation with other similar methods showed that the proposed method was able to provide a more scalable alternative by combining data fragmentation using optimal fragment size with a secret sharing scheme in key management.

REFERENCES

[1] Kaefer, G (2010). Cloud Computing Architecture. A presentation by SIEMENS in 4th generation datacenter IEEE spectrum, http://www/ct.siemens.com

[2] A. Shamir, 'How to share a secret', *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[3] Beimel, A., Chee, Y. M., Guo, Z., Ling, S., Shao, F., TY., Wang, H. & Xing, C. (ed.),(2011). Secret-Sha Schemes: A Survey‖, Springer, 6639, pp. 11 -46.

[4] K. Kapusta, G. Memmi, and H. Noura, 'An Efficient Keyless Fragmentation Algorithm for Data Protection', *ArXiv170509872 Cs*, May 2017.

[5] M. Russ, 'Secret Sharing Schemes PowerPoint PPT Presentation'. 2012.

[6] M. Klein, 'How the Cloud Changes Disaster Recovery, Industry Perspective', Jul. 2011.

[7] H. Kashiwazaki, ‘Practical uses of cloud computing services in a Japanese university of the arts against aftermath of the 2011 Tohoku earthquake’, in *Proceedings of the 40th annual ACM SIGUCCS conference on User services*, 2012, pp. 49–52.

[8] W. J. Buchanan, D. Lanc, L. Fan, G. Russell, and others, ‘The Future Internet: A World of Secret Shares’, *Future Internet*, vol. 7, no. 4, pp. 445–464, 2015.

[9] M. Nojoumian and D. R. Stinson, ‘Brief announcement: secret sharing based on the social behaviors of players’, in *Proceedings of the 29th ACM SIGACT-SIGOPS symposium on principles of distributed computing*, 2010, pp. 239–240.

[10] R. Shor, G. Yadgar, W. Huang, E. Yaakobi, and J. Bruck, ‘How to Best Share a Big Secret’, in *Proceedings of the 11th ACM International Systems and Storage Conference*, 2018, pp. 76–88.

[11] T. Takagi and K. Morozov, ‘MEXT Secret Sharing and Cloud Computing Workshop Overview’, 2011.

[12] J. L. Dautrich and C. V. Ravishankar, ‘Security limitations of using secret sharing for data outsourcing’, in *Data and Applications Security and Privacy XXVI*, Springer, 2012, pp. 145–160.

[13] M. A. Hadavi and R. Jalili, ‘Secure data outsourcing based on threshold secret sharing; towards a more practical solution’, in *VLDB 2010 PhD Workshop, Singapore*, 2010, pp. 54–59.

[14] D. Agrawal, A. El Abbadi, F. Emekci, A. Metwally, and S. Wang, ‘Secure data management service on cloud computing infrastructures’, in *New Frontiers in Information and Software as Services*, Springer, 2011, pp. 57–80.

[15] X. Tian, C. Sha, X. Wang, and A. Zhou, ‘Privacy preserving query processing on secret share-based data storage’, in *Database Systems for Advanced Applications*, 2011, pp. 108–122.

[16] M. Tompa and H. Woll, ‘How to share a secret with cheaters’, *J. Cryptol.*, vol. 1, no. 3, pp. 133–138, 1989.

[17] A. Abdallah and M. Salleh, ‘Secret sharing scheme security and performance analysis’, in *Computing, Control, Networking, Electronics and Embedded Systems Engineering (ICCNEEE), 2015 International Conference on*, 2015, pp. 173–180.

[18] G. J. Simmons, ‘How to (really) share a secret’, in Proceedings on Advances in cryptology, 1990, pp. 390–448.

[19] G. R. Blakely, ‘Safeguarding cryptographic keys’, in *Proc. AFIPS*, 1979, vol. 48, pp. 313–317.

[20] Q. Zhang, S. Li, Z. Li, Y. Xing, Z. Yang, and Y. Dai, ‘CHARM: A Cost-Efficient Multi-Cloud Data Hosting Scheme with High Availability’, *IEEE Trans. Cloud Comput.*, vol. 3, no. 3, pp. 372–386, Jul. 2015.

[21] M. Thangapandiyan and P. M. R. Anand, ‘Robust CHARM: an efficient data hosting scheme for cloud data storage system’, *Autom. Control Comput. Sci.*, vol. 51, no. 4, pp. 240–247, Jul. 2017.

[22] B. Fabian, T. Ermakova, and P. Junghanns, ‘Collaborative and secure sharing of healthcare data in multi-clouds’, *Inf. Syst.*, vol. 48, pp. 132–150, 2015.

[23] T. Ermakova and B. Fabian, ‘Secret sharing for health data in multi-provider clouds’, in *Business Informatics (CBI), 2013 IEEE 15th Conference on*, 2013, pp. 93–100.

[27] A. Abdallah and M. Salleh, ‘Analysis and comparison the security and performance of secret sharing schemes’, *Asian J. Inf. Technol.*, vol. 14, no. 2, pp. 74–83, 2015.

[28] W. Buchanan, D. Lanc, E. Ukwandu, L. Fan, and G. and, ‘The Future Internet: A World of Secret Shares’, *Future Internet*, vol. 7, no. 4, pp. 445–464, 2015.

[29] C. E. Shannon and W. Weaver, ‘The mathematical theory of communication. 1949’, *Urbana Univ Ill. Press*, 1963.

[30] B. Schneier, *Applied cryptography: protocols, algorithms, and source code in C*. John Wiley & sons, 2007.

[31] W. J. Buchanan, D. Lanc, L. Fan, G. Russell, and others, ‘The Future Internet: A World of Secret Shares’, *Future Internet*, vol. 7, no. 4, pp. 445–464, 2015.

[32] M. Nojoumian and D. R. Stinson, ‘Brief announcement: secret sharing based on the social behaviors of players’, in *Proceedings of the 29th ACM SIGACT-SIGOPS symposium on principles of distributed computing*, 2010, pp. 239–240.

[33] N. Al Ebri, J. Baek, and C. Y. Yeun, ‘Study on Secret Sharing Schemes (SSS) and their applications’, in *Internet Technology and Secured Transactions (ICITST), 2011 International Conference for*, 2011, pp. 40–45.

[34] M. Nojoumian and T. C. Lethbridge, ‘A new approach for the trust calculation in social networks’, in *E-Business and Telecommunication Networks*, Springer, 2006, pp. 64–77.

[35] Sian-Jheng Lin and Wei-Ho Chung, ‘An Efficient (n, k) Information Dispersal Algorithm for High Code Rate System over Fermat Fields’, *Commun. Lett. IEEE*, vol. 16, no. 12, pp. 2036–2039, 2012.

[36] P. Morillo, C. Padró, G. Sáez, and J. L. Villar, ‘Weighted threshold secret sharing schemes’, *Inf. Process. Lett.*, vol. 70, no. 5, pp. 211–216, 1999.

[37] F. Alsolami and T. E. Boult, ‘CloudStash: using secret-sharing scheme to secure data, not keys, in multi-clouds’, in *Information Technology: New Generations (ITNG), 2014 11th International Conference on*, 2014, pp. 315–320.

[38] S. Rajagopalan, B. Cully, R. O'Connor, and A. Warfield, 'SecondSite: disaster tolerance as a service', in *ACM SIGPLAN Notices*, 2012, vol. 47, pp. 97–108.

[39] T. Makkena and T. Rao, 'A Shamir Secret Based Secure Data sharing between Data owners', 2014.

[40] T. Takagi and K. Morozov, 'MEXT Secret Sharing and Cloud Computing Workshop Overview', 2011.

[41] J. L. Dautrich and C. V. Ravishankar, 'Security limitations of using secret sharing for data outsourcing', in *Data and Applications Security and Privacy XXVI*, Springer, 2012, pp. 145–160.

[42] M. A. Hadavi and R. Jalili, 'Secure data outsourcing based on threshold secret sharing; towards a more practical solution', in *VLDB 2010 PhD Workshop, Singapore*, 2010, pp. 54–59.

[43] D. Agrawal, A. El Abbadi, F. Emekci, A. Metwally, and S. Wang, 'Secure data management service on cloud computing infrastructures', in *New Frontiers in Information and Software as Services*, Springer, 2011, pp. 57–80.

[44] X. Tian, C. Sha, X. Wang, and A. Zhou, 'Privacy preserving query processing on secret share-based data storage', in *Database Systems for Advanced Applications*, 2011, pp. 108–122.

[45] M. Tompa and H. Woll, 'How to share a secret with cheaters', *J. Cryptol.*, vol. 1, no. 3, pp. 133–138, 1989.

[46] Z. Li, L. O'brien, H. Zhang, and R. Cai, 'On a catalogue of metrics for evaluating commercial cloud services', in *Grid Computing (GRID), 2012 ACM/IEEE 13th International Conference on*, 2012, pp. 164–173.

[47] A. K. Bardsiri and S. M. Hashemi, 'Qos metrics for cloud computing services evaluation', *Int. J. Intell. Syst. Appl.*, vol. 6, no. 12, p. 27, 2014.

[48] S. Narani, 'Social Secret Sharing for Resource Management in Cloud', *ArXiv Prepr. ArXiv13021185*, 2013.