Webology, Volume 18, Special Issue on Artificial Intelligence in Cloud Computing
April, 2021

Proposed Cloud Based Distributed Key Management System Based On Secret Sharing With Encryption

Mrs. Kruti Patel [1] , Dr. Kamaljit Lakhtaria [2]

[1] Dept. of Computer Science, Gujarat University, Ahmedabad, Gujarat, India-380009.

[2] Dept. of Computer Science, Gujarat University, Ahmedabad, Gujarat, India-380009.

Abstract: As there is high demand for IT there are multiple scope of big data which securely used by multi cloud infrastructure with use of secure data sharing techniques on cloud. Though multiple secrete sharing methods has been invented but still there is always gap in knowledge which based on evaluation of methods respective to scalability and Key management. This paper represents an evaluation of a proposed method that combines data fragmentation with use of Shamir's secrete sharing scheme which known as Fragmented Encrypted Secret Sharing System (FESSS).The proposed method applies data fragmentation with optimum fragment size and also each fragment will be encrypted with use of AES 256-bit algorithm. Afterwards each encrypted fragment will be dispersal to cloudlets. The key which is used for encryption in FESSS is managed by secrete sharing technology. After evaluating proposed system, it is observed that FESSS method showed that the less overhead cost fragmented method based on irrespective of files size and share policy combination is far better. The major drawback is that it is quite difficult to find corrupted and lost fragments during file recovery.

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

1. INTRODUCTION

Information Technology (IT) resources which already available in the cloud has made the adoption of attractive in-service sectors [1] as a rapid result of growth in knowledge economy. However, with the growth of data regarding types and sizes, concerns have been raised on how best to transmit data securely as well as share and make them available without interrupt which is irrespective of the size and type.

Adi Shamir [2] and George Blakely [3] did classic publications in 1979 on how to share data securely without using encryption key known as keyless encryption by defining a method that breaks data (secret) into a number of shares and certain number of these that can come together to recover the secret with less of offered shares, later on it known as Secret Sharing Scheme. The number of shares created is equivalent to the number of participants, and the number that is required to recover the secret is known as the threshold, this is known as share policy.

This proposed FESSS scheme uses two main protocols of secret share creation and share recovery. The application has proved to be secure and efficient in secrete sharing and recovering data in a cloud distributed system. In fact, the use of secret sharing scheme also has some limitations, of which are the inability to provide data operations at large-scale data size and the effects of changing share policies on system overheads. Keyless encryption implies break the data into shares in such a manner that each share of the data exists in a meaningless manner and original data will get recovered only when certain defined number known as

threshold or more can come together. A share policy implies a defined threshold and maximum number of shares to be made from a data. The original data recovery is only possible when the threshold shares or you can say number of shares equal to the total number of shares are come together using an secrete sharing algorithm.

2. FILE FRAGMENTS BASED ON CLOUD SYSTEMS

In the face of these current realities, we present an evaluation of a method for sharing large-scale data infrastructure in multi-clouds using a combination of data fragmentation and secret share scheme. A system that can provide consistent data availability, high-level scalability and security, as well as maintaining data integrity within cloud-based architecture know as Fragmented Encrypted Secret Share System (FESSS). It creates fragments from a file, encrypts each fragment and applies secret sharing methods as used in cryptography to create robust and secure keyless key management system in a multi-clouds data distribution system.

The process involves the user providing the file(s) as well as choosing desired share policy for each operation, while the system provides appropriate optimum fragment size and number of cloudlets that will participate in the operation. It goes forward by breaking the file into chunks using the chosen fragment size, encrypting each chunk with different AES-256-bit key generated by a random key generator and then creates shares out of the encryption key based on user's chosen share policy. The shares, as well as the encrypted fragments are stored with selected Cloud service Providers and when the file is required, the key shares are recovered using the same key share policy in relations to the defined threshold and as well as the encrypted fragments. Each recovered key is therefore used to decrypt corresponding encrypted fragment, and with the fragments decrypted serially, the original file is recombined and file deliver to the file owner.

3. SECRETE SHARES

Shor et al. [4] suggests that the optimal way of sharing big data in multi-cloud environments is in the combination of data encryption and use of efficient secret sharing scheme in managing encryption key. in the light of this, four experiments were performed to measure scalability, resilience and key management of FSSS and evaluate same with the evaluation frameworks developed by this thesis on the areas of scalability and resilience. The first experiments used defined fragment blocks to break file of varied block sizes into fragments. The second experiments used an optimum fragment size defined as 15% of file size. Both were used in a combination of different secret sharing policies and our results showed that defining an optimum fragment size of 15% of file size produced less overhead than the first experiment.

4. SECRETE SHARING SCHEME

The deviation from key-based to keyless encryption was introduced by Adi Shamir and George Blakely in 1979 [2, 3] in two different seminal papers, each presented to the world a different means of securing cryptographic keys. Their works focused on splitting the key into meaningless shares in such a way that it will take only a certain number of the broken keys (shares) known as a threshold to come together and reconstruct the key and any number less than the threshold cannot. This concept was later known to be Secret Sharing Scheme. This scheme focuses on the techniques used in striping and distribution of data among many participants in such a way that a certain number of the participants, known as the threshold, can come together and recover the original data while a certain number less than the threshold cannot [10].

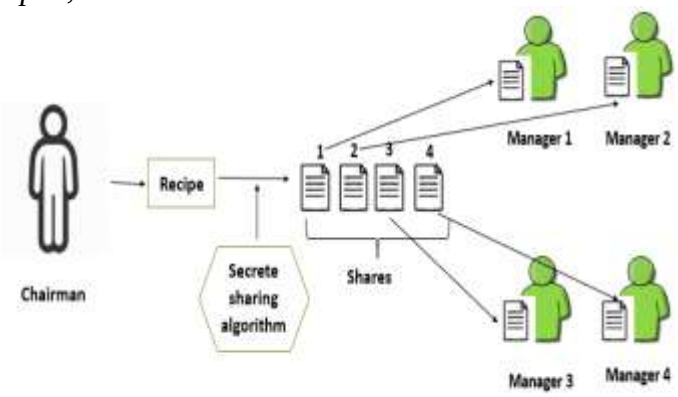

Fig. 1: Diagram of Secrete Sharing

The Shamir Secret sharing scheme is an ideal scheme and known as a perfect scheme [2], while many provide computational security such as [11]. Krawczyk [11] is of the opinion that an (n,m) −secret sharing scheme is a randomized protocol that stripes a secret S and disseminate same as shares to an n participants in such a way that only an m participants shares are capable of recovering the original secret for m, $1 \leq m \leq n$, while $m - 1$ shares cannot give any information on the secret as presented by [2, 3]. We take for instance a beverage company and the owner want to make their recipe a top secret.

Their intention is to prevent their managers from learning the recipe and in so doing decided to use an algorithm to break their recipe into shares (Si) in such a way that a certain number of the shares (M) will be enough to recover the recipe out of the total number distributed to managers (N). After breaking the recipe into shares, the shares were distributed to say four of their top managers knowing that at least two from the four are needed to get the recipe back. This total number of shares made of the recipe is equivalent to the number of players (Managers) and the minimum number required to get the recipe back is known as a threshold.

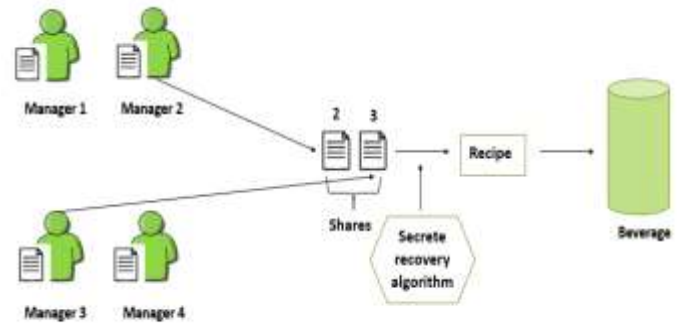

Fig. 2: Diagram of Secrete Recovery

On a last note, on secret sharing scheme is Social Secret sharing scheme (SSSS), which combines the features of Weighted Threshold Secret Sharing Scheme (WTSSS) [2], [41], [42], [43], [44] and that of Proactive Secret Sharing Scheme (PSSS) [45] in its design and concepts. We will be exploring its design

principle of Sharing-Tuning-Recovery method in our future works while leveraging on this design to provide a self-organizing system as proposed with a high level of data scalability in a multi-cloud architecture.

The Shamir's secrete sharing implies providing perfect secrecy as shares less than the threshold cannot recover or learn of the secret, but a trade-off with performance. The larger the file size, the larger each share is and an increase in the number of participants in the sharing algorithm leads to higher storage overhead of the shares. With this, Shamir's scheme is unsuitable for sharing large-scale data infrastructure according to [39], [31], [46], [5], [19], [13], [12], [18], [47] and [17].

5. SYSTEM COMPONENTS

The proposed system FESSS is made up of main five components:

1. User Management: Login details, file size options, share policy options, metadata creation.

2. File Fragmentation: fragments creation, key generation, encryption, share creation, share dispersion, storage and then followed by numbers 4 and 5 below.

3. File Recreation: Login details, metadata retrievals, encrypted fragments recovery, key share recovery, key recreation, fragments decryption, file recreation, checksum and storage, followed by numbers 4 and 5 below.

4. Cloud Behavioral Computations: It depends on whether had multi cloud structure or not.

5. Agent – Analyses and future behavioral predictions, clean-ups, self-organization if needed (future works).

6. CLOUD STORGE SECRETE SHARING ADVANTAGES

1. Secure: Anyone with fewer than t shares have no extra information about the secret than someone with zero shares.

2. Extensible: When n is fixed; new shares can be dynamically added or deleted without affecting the existing shares.

3. Dynamic: With this it is possible to modify the polynomial and construct new shares without changing the secret.

4. Flexible: In organizations where hierarchy is important, it is possible to supply each of the participants a different number of shares according to their importance.

7. CLOUD BASED KEY MANAGEMENT SYSTEM

The security of stored data in cloud is crucial in cloud computing as Wang et al. [26] posit that such necessitates the need for the design of a key management scheme that is reliable for safe computing in the cloud. Rao [27] agrees that key management is not standardized optimally in the cloud. Rao and Selvamani [28] are of the view that having only authorized users to have access to the decryption key is the best management. While Zissis and Lekkas [29] suggest that having a trusted third party is the way to go. In order to achieve this objective as opined, two major methods suffice – keyless and In-house-key-storage management system. In keyless system [4], [18], [17], [25] and [78] used secret sharing scheme, but Resch

and Plank [52] dispersed encryption key with the encrypted data in cloudlets, while [87], [30] proposed key aggregate key management system in managing In-house-key-storage. In all, it shows how important key management is in cloud-based data storage. From the methods presented, the keyless type provides a system that prevents key loss, theft and leakages and as such is resilient, reliable and as well provide confidentiality and availability.

8. METHOD OF KEY SHARING AND RECOVERY

The base algorithms – Sharing and Recovering are the concept as presented originally by Adi Shamir [2] and hence a perfect secret sharing scheme.

In first experiment, the overhead cost is the sum of time taken to create key shares; write key shares to storage devices; recover key shares from storage devices based on defined threshold for key recovery. The time taken are quite infinitesimal but a reference to them is necessary for comparison with that of second experiment, which was done using different cloud service providers presenting a real-life situation to share creation; share writing; share recovery based on a prevailing threshold and key recovery.

In second experiment the overhead cost for key sharing and recovery are based on time taken in key sharing, share writing to cloudlets which is made up of upload and download times, share recovering from downloads and secret key recovering.

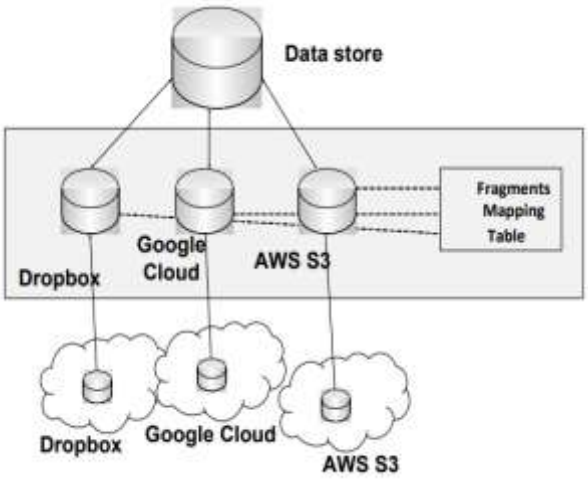

Fig. 3: FESSS Encrypted Fragments Mappings to Cloudlets

9. 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.

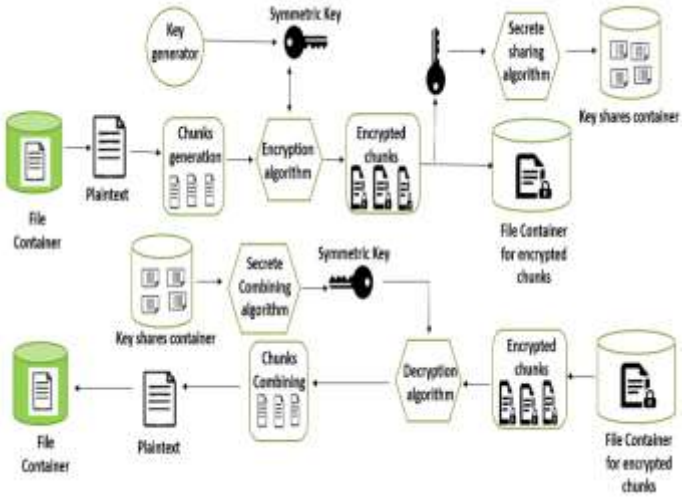

Fig. 4: Proposed system share generation and Share recovery

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 out of those chunks are 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 FESSS.

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's decryption algorithm will generate decrypted chunks. Afterwards Chunks are combined through merging algorithm. Then user will get original file through chunks Combination.

10. PROPOSED SYSTEM ALGORITHM

The proposed system is evaporated in two ways. The first one for File upload system and second for File download system.

10.1 FILE UPLOAD SYSTEM

Step:1 User upload file
Step:2 File gets fragmented into number of chunks as per the size of file
Step:3 Key generators will generate symmetric key for encryption
Step:4 Each fragmented chunks get encrypted with symmetric key
Step:5 Encrypted chunks will be stored in storage container
Step:6 Encrypted key gets converted into multiple shares through secrete sharing algorithm
Step: 7 Each share is stored on different data center of cloud providers

10.2 FILE DOWNLOAD SYSTEM

Step: 1 Each share is getting combined and generate symmetric decryption key
Step: 2 Each Encrypted chunks are assigned to decryption algorithm
Step: 3 Decryption algorithm will generate decrypted chunks
Step: 4 Chunks are combined through merging algorithm

Step: 5 Plaintext is generated through chunks combination
Step: 6 User can download original file

11. CONCLUSION AND FUTURE WORK

Use of secret sharing scheme has some inherent limitations, of which are the inability to provide data operations at large-scale data size and the effects of changing share policies on system overheads. The works of Abdallah and Salleh in [19] and [27] provided extensive information on all these without any known solutions and these are what FESSS provided using fragmented secret share system through defined optimum fragment size. By developing two evaluation frameworks on scalability and resilience, this thesis defined scalability as the ability to continue production even when file size increases exponentially and resilience of secret sharing as the ability to continue production in multi-cloud environments during adverse cloud failures. These FESSS's results and future works suggest are the better approach than existing methods.

REFERENCES

[1] H. Hassan, 'Organizational factors affecting cloud computing adoption in small and medium enterprises (SMEs) in service sector', Procedia Compute. Sci., vol. 121, pp. 976–981, Jan. 2017.
[2] A. Shamir, 'How to share a secret', Commun. ACM, vol. 22, no. 11, pp. 612–613, 1979.
[3] G. R. Blakely, 'Safeguarding cryptographic keys', in Proc. AFIPS, 1979, vol. 48, pp. 313–317.
[4] 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.
[5] E. Ukwandu, W. J. Buchanan, and G. Russell, 'Performance evaluation of a fragmented secret share system', in Cyber Situational Awareness, Data Analytics
And Assessment (Cyber SA), 2017 International Conference On, 2017, pp. 1–6.
[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] OnlineTech, 'Disaster Recovery White Paper.', 2013.
[9] Y. Gu, D. Wang, and C. Liu, 'DR-Cloud: Multi-cloud-based disaster recovery service', Tsinghua Sci. Technol., vol. 19, no. 1, pp. 13–23, Feb. 2014.
[10] M. Russ, 'Secret Sharing Schemes PowerPoint PPT Presentation'. 2012.
[11] H. Krawczyk, 'Secret sharing made short', in Advances in Cryptology—CRYPTO'93, 1993, pp. 136–146.
[12] 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.
[13] 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.
[14] 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.
[15] 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.
[16] T. Loruenser, A. Happe, and D. Slamanig, 'ARCHISTAR: towards secure and robust cloud-based data sharing', in Cloud Computing Technology and Science (CloudCom), 2015 IEEE 7th International Conference on, 2015, pp. 371–378.
[17] 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.
[18] K. Kapusta, G. Memmi, and H. Noura, 'An Efficient Keyless Fragmentation Algorithm for Data Protection', ArXiv170509872 Cs, May 2017.
[19] 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.
[20] R. Koikara, D.-S. Kim, E.-J. Yoon, A. Paul, and K.-Y. Yoo, 'Towards Security in Multi-clouds Using Secret Sharing', pp. 1557–1558, 2017.
[21] D. Pal, P. Khethavath, J. P. Thomas, and T. Chen, 'Multilevel threshold secret sharing in distributed cloud', in International Symposium on Security in Computing and Communication, 2015, pp. 13–23.
[22] H. Xiong, X. Zhang, D. Yao, X. Wu, and Y. Wen, 'Towards End-to-end Secure Content Storage and Delivery with Public Cloud', in Proceedings of the Second
ACM Conference on Data and Application Security and Privacy, New York, NY, USA, 2012, pp. 257–266.
[23] A. Bessani, M. Correia, B. Quaresma, F. André, and P. Sousa, 'DepSky: dependable and secure storage in a cloud-of-clouds', ACM Trans. Storage TOS, vol. 9, no. 4, p. 12, 2013.
[24] 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.
[25] T. Loruenser, A. Happe, and D. Slamanig, 'ARCHISTAR: towards secure and robust cloud-based data sharing', in Cloud Computing Technology and Science (CloudCom), 2015 IEEE 7th International Conference on, 2015, pp. 371–378.
[26] Y. Wang, Z. Li, and Y. Sun, 'Cloud computing key management mechanism for cloud storage', 2015.
[27] B. T. Rao, 'A study on data storage security issues in cloud computing', Procedia Comput. Sci., vol. 92, pp. 128–135, 2016.
[28] R. V. Rao and K. Selvamani, 'Data Security Challenges and Its Solutions in Cloud Computing', Procedia Comput. Sci., vol. 48, pp. 204–209, Jan. 2015.
[29] D. Zissis and D. Lekkas, 'Addressing cloud computing security issues', Future Gener. Comput. Syst., vol. 28, no. 3, pp. 583–592, 2012.
[30] P. Gharjale and P. Mohod, 'Efficient public key cryptosystem for scalable data sharing in Cloud storage', in Computation of Power, Energy Information and Commuincation (ICCPEIC), 2015 International Conference on, 2015, pp. 0325–0329
[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 SIGACTSIGOPS symposium on principles of distributed computing, 2010, pp. 239–240.
[33] M. Nojoumian, D. R. Stinson, and M. Grainger, 'Unconditionally secure social secret sharing scheme', Inf. Secur. IET, vol. 4, no. 4, pp. 202–211, 2010.
[34] M. Nojoumian and D. R. Stinson, 'Social secret sharing in cloud computing using a new trust function', in Privacy, Security and Trust (PST), 2012 Tenth Annual International Conference on, 2012, pp. 161–167.
[35] O. Farràs, T. Hansen, T. Kaced, and C. Padró, 'On the Information Ratio of NonPerfect Secret Sharing Schemes', Cryptology ePrint Archive, Report 2014/124, 2014. https://eprint. iacr. org/2014/124. pdf, 2015.
[36] C. Asmuth and J. Bloom, 'A modular approach to key safeguarding', IEEE Trans. Inf. Theory, vol. 30, no. 2, pp. 208–210, 1983.
[37] E. F. Brickell, 'Some ideal secret sharing schemes', in Advances in Cryptology— EUROCRYPT'89,

1989, pp. 468–475.
[38] G. J. Simmons, 'How to (really) share a secret', in Proceedings on Advances in cryptology, 1990, pp. 390–448.
[39] B. Buchanan, Cryptography. River Publishers, 2017.
[40] M. O. Rabin, 'Efficient dispersal of information for security, load balancing, and fault tolerance', J. ACM JACM, vol. 36, no. 2, pp. 335–348, 1989.
[41] J. Benaloh and J. Leichter, 'Generalized secret sharing and monotone functions', in Proceedings on Advances in cryptology, 1990, pp. 27–35.
[42] 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.
[43] A. Beimel, T. Tassa, and E. Weinreb, 'Characterizing ideal weighted threshold secret sharing', in Theory of Cryptography, Springer, 2005, pp. 600–619.
[44] S. Yoo, P. Park, J. Shin, and J. Ryou, 'Key sharing scheme based on one weighted threshold secret sharing', in Advanced Communication Technology (ICACT), 2013 15th International Conference on, 2013, pp. 317–320.
[45] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung, 'Proactive secret sharing or:How to cope with perpetual leakage', in Advances in Cryptology—CRYPT0'95, Springer, 1995, pp. 339–352.
[46] E. Ukwandu, W. J. Buchanan, L. Fan, G. Russell, and O. Lo, 'RESCUE: Resilient Secret Sharing Cloud-Based Architecture', in Trustcom/BigDataSE/ISPA, 2015 IEEE, 2015, vol. 1, pp. 872–879.
[47] S. Takahashi and K. Iwamura, 'Secret sharing scheme suitable for cloud computing', in Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference on, 2013, pp. 530–537.
[48] 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.
[49] 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.
[50] 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.
[51] H. Lahkar and M. R, 'Towards High Security and Fault Tolerant Dispersed Storage System with Optimized Information Dispersal Algorithm', Int. J. Adv. Res. Comput. Sci., vol. 5, no. 6, Jul. 2014.
[52] J. . Resch and J. . Plank, 'AONT-RS: Blending security and performance in dispersed storage systems', Proc USENIX FAST '11, 2011.