AN ENHANCED CODING BASED SECRET SCHEME FOR FILE SHARING IN BIG DATA

[1]J.Vishnupriya, [2]Dr.N.Puviarasan
[1]Mphil. Research scholar, [2] Associate Professor
[1]Department of Computer and Information Science
[1]Annamalai University, Tamilnadu, India

ABSTRACT: Transmission and storage of information should be performed safely and effectively. Information should be handled the stored and submitted by fast and wide-ranging network connections. Errors should be corrected that occur during the process. Syndrome decoding is a method of correcting errors. Secret sharing is an important topic in cryptography. For the syndrome decoding we want only two columns. They are syndromes and coset leaders. This table is called syndrome- decoding. In this work, Firstly, the nonzero coset leaders which are to be minimal are minimal access sets here. The components in coset leader can recover the secret by combining their shares. The order of the components is too important. Because, the position of any component is changed, then the secret cannot recover. This means the access structure of this share scheme is very strong and reliable. Thus this scheme explores some relations between secret sharing schemes based on the binary linear code which is a single error correcting and syndrome decoding. Then we obtain some results using this relation.

Index Terms- Big Data, Syndrome Decoding, Coset Leaders, NonZero Coset Leaders, Binary Linear Code

I. INTRODUCTION

Huge Data is turning into another time in information investigation. The measure of information is expanding exponentially and along these lines, a various applications, for example, cloud or appropriated stockpiling framework, are acquainted with diminish the weight of information the executives for information proprietors. Be that as it may, alongside the information usage of such frameworks, there are a great deal of security challenges in which the most widely recognized dangers are the information spillage and devastation. To secure against the dangers, mystery sharing plan is a perfect strategy which has been utilized all the more prevalently in conveyed frameworks. Mystery sharing plan is utilized for disseminating a mystery among a gathering of members with the assistance of a merchant. Every member holds an offer of the mystery. The mystery must be reproduced when there are sufficient number of offers joining together. Each offer can't be utilized alone to extricate significant data. A utilization instance of mystery sharing plan. In correspondence specialist organizations (CSP), the information gathered from client's regular gain significantly. Putting away, overseeing and backing up these Big Data are troublesome assignments for any CSP. Accordingly, they will in general use cloud or disseminated stockpiling frameworks to store such enormous information. For saving the security of the delicate information (e.g., client data), mystery sharing plan is a promising methodology. The mystery information is encoded and appropriated to a lot of members in a way that it must be reproduced from approved.

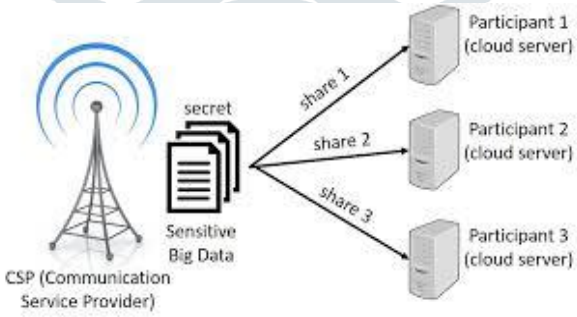

Figure 1: CSP Secret Sharing Scheme

1.1 SCOPE

Visual data discovery tools will be developing 2.5 times quicker than rest of the Business Intelligence (BI) advertises. By 2018 investing into this enabler of end-user self-service will become a requirement for all enterprises. Over the following five years spending on cloud-based Big Data and analytics (BDA) solutions will grow on three times quicker than spending for on-premise solutions. Hybrids on/off premise deployments will turn become equirement. Shortage of skilled staff will continue. In

the U.S. alone there will be 181,000 deep analytics exert in 2018 and five times that many positions requiring related skills in data Administration and interpretation. By 2017 unified data platform architecture will become into the foundation of BDA strategy. The unification will occur across data information management, analysis, and search technology. Growth in applications incorporating advanced and predictive analytics, including machine learning, will quicken in 2015. This application will grow 65% faster than application without predictive functionality. 70% of large partnership already purchases external data and 100% will do such by 2019. In parallel more partnership will begin to monetize their data by selling them or providing value-added content.

Adoption of technology to continuously analyze streams of events will accelerate in 2015 as it is applied to Internet of Things (IoT) analytics, which is expected upon to grow at a five-year compound yearly growth rate (CAGR) of 30%. Decision administration platforms will expand at a CAGR of 60% through 2019 in response to the need for greater consistency in decision making and basic decision making process knowledge retention.

1.2 HADOOP-BIG DATA

Because of the appearance of new innovations, gadgets, and correspondence implies like interpersonal interaction destinations, the measure of data delivered by humankind is developing quickly consistently. The measure of data created by use from the earliest starting point of time till 2003 was 5 billion gigabytes. In the event that you heap upon the information as circles it might whole football field. A similar number was made in each two days in 2011, and in at regular intervals in 2013. This rate is as yet developing colossally. In spite of the fact that this data created is significant and can be helpful when handled, it is being dismissed.

- HDFS
- HDFS RESOURCE

HDFS is organized out comparatively to a ordinary Unixfilesystem aside that information stockpiling is distributedacross a few machines. It isn't expected upon as a substitution to a customary filesystem, but instead as a filesystem-like layer for vast appropriated frameworks to utilize. It has in fabricated components to deal with machine blackouts and is streamlined for throughput as opposed to dormancy.

Datanode - where HDFS actually stores the data information, there are usually quite a few of these.

Namenode -the 'master' machine. It controls the entire Meta data form cluster. Eg - what blocks document makes up a file and what datanodes those blocks are stored on?

Secondary Namenode - NOT a backup namenode, but is a separate service that keeps a copy of both the edit logs and filesystem image, merging them periodically to keep the size reasonable. This is soon being deprecated in favor of the checkpoint node and the backup node, but the functionality remains similar (if not the same).
For more information about the design of *HDFS (Hadoop Distributed File System),* you should read through apache software documentation page. In particular the streaming and data access section has some really simple and informative graphs on how data read/writes actually happen.

1.3 MapReduce

The second fundamental piece of Hadoop is the MapReduce layer. This is construct up of two sub components: one of the most important parts of a MapReduce work is what happens between map and reduce, there are 3 different stages; Partitioning, Sorting, and Grouping. In the default configuration, the object of these intermediate steps is to ensure this behavior; that the values for every key are grouped together ready for the reduce () function. APIs are also assuming if you want to tweak how these phase work (like if you want to perform a secondary sort).

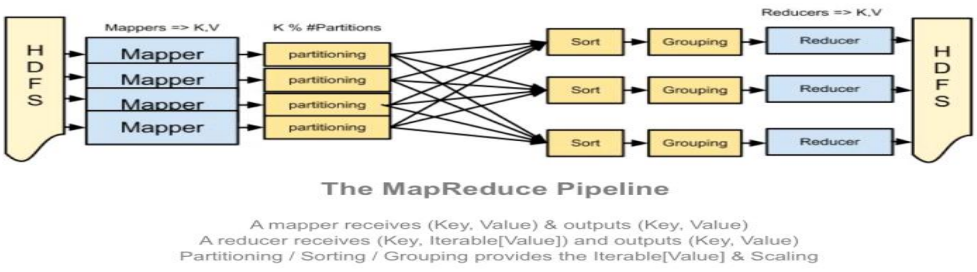

Figure 2: The MapReduce Pipeline

What's really extremely about this API is that there is no dependency between any two of the same task. To do its job a map () task does not need to know about other map task, and similarly a single reduce () task has the entire context it needs to aggregate for any specific key, it does not share any state with other reduce tasks.

II. LITERATURE SURVEY

O. Farras, T. Hansen, T. Kaced, and C. Padro A secret sharing scheme is non-perfect if a few subsets of members that can't recoup the mystery esteem have incomplete data about it. The information proportion of a secret sharing scheme is the proportion between the maximum length of the off share and the length of the secret. This work is devoted to the search of bound on the data information of non-perfect secret sharing scheme. To this end, we broaden the known connections between polymatroids and perfect secret sharing scheme to the non-perfect case. In the order to study about non-perfect secret sharing scheme in all generality statement, we describe their structure through their entrance function, a real function that the measure of the data information that each subset of participants obtains about the secret value. We prove that there exists a secret sharing scheme for each access function. Uniform access on function, that is, the ones whose qualities depend just on the quantity of participants, generalize up the threshold access structures. Our principle result is to determine the optimal data information ratio of the uniform access function. Additionally, we present a development of linear secret sharing to optimal information ratio for the rational uniform access function.

R. Matsumoto We introduces a coding conceptual criterion for Yamamoto's strong security of the ramp secret sharing scheme. After that, by using it, we convey the strong security of the strongly multiplicative ramp secret sharing proposed by Chen et al. in 2008.

Y. Wang, LDPC codes, LT codes, and digital fountain techniques have received consequential attention from both academics and industry in the past few years. By employing the fundamental ideas of efficient Belief Propagation (BP) decoding process in LDPC and LT codes, this paper designs the BP-XOR codes and use them to structure three category of secret sharing schemes called BP-XOR secret sharing schemes, pseudo-BP-XOR secret sharing schemes, and LDPC secret sharing schemes. By establishing the identity between the edge-colored graph model and degree-two BP-XOR secret sharing schemes, we are expert to design novel perfect and ideal 2-out-of-n BP-XOR secret sharing schemes. By employing techniques from array code design, we are also expert to design other (n; k) threshold LDPC secret sharing schemes. In the efficient (pseudo) BP-XOR/LDPC secret sharing schemes that we will establish, only linear number of XOR (exclusive-or) operations on binary strings are required for both secret distribution phase and secret reconstruction aspect.

For a comparison, we should note that Shamir secret sharing schemes require O (n log n) field operations for the secret distribution phase and O (n2) field operations for the secret reconstruction aspect. Further, our schemes achieve the optimal update complexity for secret sharing schemes. By update complexity for a secret sharing scheme, we mean the average number of bits in the participant's shares that needs to be adapting when certain bit of the master secret is changed. The extremely efficient secret sharing schemes discussed in this paper could be used for huge data storage in cloud environments achieving privacy and reliability without employing encryption techniques.

J. Li, X. Chen, M. Li, J. Li, P.P.C. Lee, and W. Lou Data deduplication is a technique for remove duplicate copies of data, and has been extensive used in cloud storage to reduce storage space and upload bandwidth. Promising as it is, an arising challenge is to execute secure deduplication in cloud storage. Although convergent encryption has been extensively acquire for secure deduplication, a critical case of making convergent encryption practical is to efficiently and reliably manage a huge number of convergent keys. This paper makes the first attempt to formally address the problem of achieving reliable and efficient key management in secure deduplication. We first introduce a baseline approach in which every user holds an independent master key for encrypting the convergent keys and outsourcing them to the cloud. However, such a baseline key management scheme generates an enormous number of keys with the increasing number of requires users and user to dedicatedly protect the master keys.

To this end, we propose Dekey, a new construction in which users do not need to manage any keys on their own but alternatively securely distribute the convergent key shares across multiple servers. Security analysis demonstrates that Dekey is secure in terms of the definitions indentify in the proposed security model. As a verification of concept, we implement Dekey using the Ramp secret sharing scheme and demonstrate that Dekey incurs restricted overhead in realistic environments

III. PROPOSED SYSTEM

In this work, Firstly, the nonzero coset leaders which are to be minimal are minimal access sets here. The components in each coset leader can recover the secret by combining their shares. Also, the arrange of the components is too important. Because, if the position of any component is changed, then the secret cannot recover. This means the access structure of this scheme is very reliable and strong. Thus this scheme explores some relations between secret sharing schemes based on the binary linear code which is a single error correcting and syndrome decoding. Then we obtain some results using this relation.

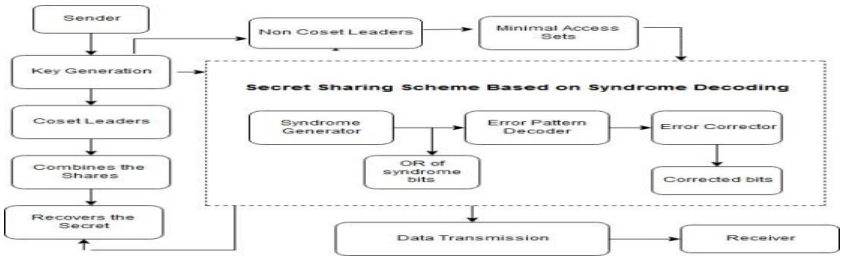

Figure 3: Secret Sharing Based On Syndrome Decoding

1V. IMPLEMENTATION

4.1 INPUT DESIGN

The input design is the link between the information system and the client. It comprises the creating specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable shape form for processing can be achieved by assessing the computer to read data from a written or printed document or it can happen by having people keying the data directly into the system. The design of input required on controlling the amount of input focuses, controlling the errors, avoiding extra, avoiding delay, steps and keeping the process simple. The input is designed in such a route so that it provides security and ease of use with retaining the privacy. Information Input Design considered about the following things.

4.2 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end client and presents the information clearly. In any system results of processing are communicated to the client and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and the hard copy output. It is the most important and direct source data information to the client. Efficient and intelligent output design improves the system's relationship to help client decision-making.

Designing computer output should proceed in an organized, well thought out manner way; the right output must be developed while ensuring that each output element is designed so that people will find the system can use effectively easily and easily . When analysis design computer output they should identify the specific output that is needed to be meet the requirements.

4.3 SOFTWARE RISK:

Risk is an expectation of loss a potential problem that may or may not occur in the potential. It is generally caused time to lack of information, control or due. A capability of suffering from loss in software development process is called a software risk. Loss can be anything, increase in production cost, development of poor quality software, not being capable to complete the project on time. Software risk exists because the potential is uncertain and there are many known and unknown things that cannot be incorporated in the project plan. A software risk can be of two types (1) external risks that are within the control of project manager (2) internal risks that are beyond the control of the project manager.

4.3.1 Software Risk Identification

In order to identify the risks that your project may be content to, it is important to first study the problems faced by previous projects. Study the project scheme properly and check for all the possible areas that are exposed to some or the other type of risks. The best ways of analyzing a project scheme is by converting it to a flowchart and examine all essential areas. It is important to conduct deal brainstorming sessions to identify the known unknowns that can affect the project. Any decision taken related to technical, operational, political, legal, social, external or external factors should be evaluated properly.

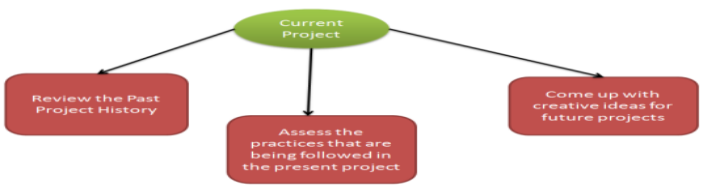

Figure 4: Software Risk Identification

In this period of Risk management you have to define processes that are important for risk identification. All the details of the risk such as individual Id, date on which it was identified, description and so on should be clearly mentioned.

4.3.2 Software Risk Analysis

Software Risk analysis is a very important aspect of risk management. In this phase the risk is identified and then categorized. After the categorization of risk, the level, likelihood (percentage) and impact of the risk is analyzed. Likelihood is defined in percentage after examining what are the chances of risk to occur due to various technical conditions. These technical conditions can be:

- Complexity of the technology
- Technical knowledge possessed by the testing team
- Conflicts within the team
- Teams being distributed over a large geographical area
- Usage of poor quality testing tools

V. Result

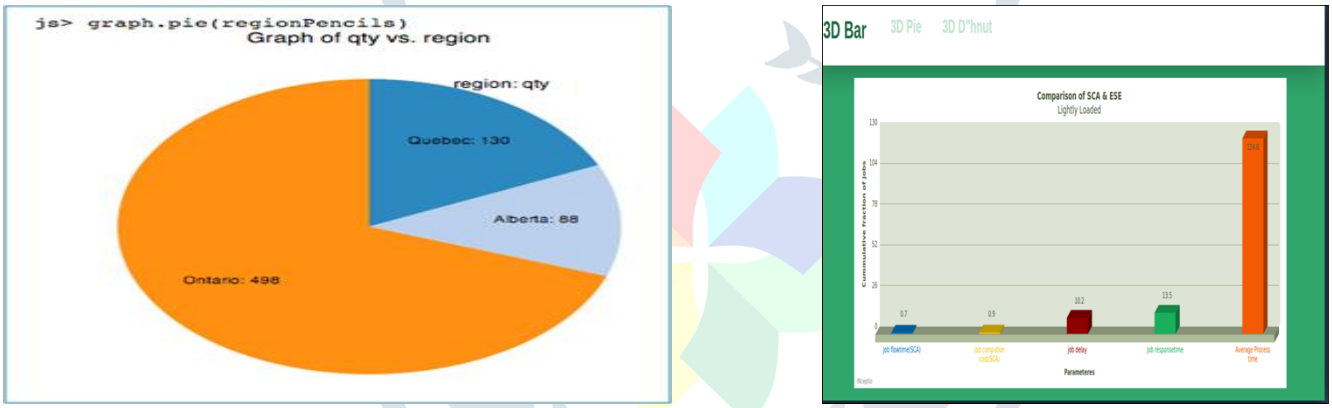

Figure 5: Pie Chart and Bar Graph

VI. Conclusion

Thus a new agglomerative hierarchical clustering algorithm is presented in this paper. It is implemented by Map Reduce framework. The approach divides the original documents' vectors' set into separation with the method of initial classification, and then distributes the separation to different data nodes of Map Reduce framework. Finally it processes the documents' vectors in each data node with accepted agglomerative hierarchical clustering algorithm. Benefit from the paralleled procession in each data node, the efficiency of clustering is enhancing by 86.5%. The speed-up ratio of our framework, which subsists of 20 computers, is up to 7.414. Compared with traditional k-means and AHC algorithm the accuracy especially the recollect rate of our new approach is improved. Result of experiments conveys that our new algorithm implemented on Map Reduce framework can apply in large-scale dataset clustering satisfactory.

REFERENCES

[1] K. Kurosawa, K. Okada, K. Sakano, W. Ogata, and T. Tsujii, "Non perfect secret sharing schemes and matroids", Workshop on the Theory and Application of Cryptographic Techniques (EUROCRYPT'93), vol. 765, pp. 126-141, 1993.

[2] B. Fortescue, and G. Gour, "Reducing the Quantum Communication Cost of Quantum Secret Sharing", IEEE Trans. Inf. Theory, no. 58, no. 10, pp. 6659-6666, 2012.

[3] R. Matsumoto, "Strong Security of the Strongly Multiplicative Ramp Secret Sharing Based on Algebraic Curves", IEICE Trans. On Fundamentals, vol. E98-A, no. 7, pp.1576-1578, 2015.

[4] J. Kurihara, S. Kiyomoto, K. Fukushima, and T. Tanaka, "A fast (3,n)-threshold secret sharing scheme using exclusive-OR operations", IEICE Trans. on Fundamentals, vol. E91-A, no. 1, pp. 127-138, 2008.

[5] J. Kurihara, S. Kiyomoto, K. Fukushima, and T. Tanaka, "A new (k; n)-threshold secret sharing scheme and its extension", 11th conf. on Information Security (ISC'08), pp. 455-470, 2008.

[6] L. Chunli, X. Jia, L. Tian, J. Jing, and M. Sun, "Efficient Ideal Threshold Secret Sharing Schemes Based on EXCLUSIVE-OR Operations", 4th Conf. on Network and System Security (NSS'10), pp. 136-143, 2010.

[7] Y. Wang, and Y. Desmedt, "Efficient Secret Sharing Schemes Achieving Optimal Information Rate", Inf. Theory Workshop (ITW), pp. 516-520, 2014.

[8] J. Kurihara, S. Kiyomoto, K. Fukushima, and T. Tanaka, "A fast (k-L-N)-Threshold Ramp secret sharing scheme", IEICE Trans. On fundamentals, doi:10.1587/transfun.E92.A.1808, 2009.

[9] M. Kurihara, and H. Kuwakado, "Secret Sharing Schemes Based on Minimum Bandwidth Regenerating Codes", Symposium on Inf. Theory and its Applications (ISITA'12), pp. 255-259, 2012.

[10] J. Liu, H. Wang, M. Xian, and K. Huang, "A Secure and Efficient Scheme for Cloud Storage against Eavesdropper", 15th Conf. on Information and Communication Security (ICICS'13), pp. 75-89, 2013.

[11] O. Farras, T. Hansen, T. Kaced, and C. Padro, "Optimal Non-perfect Uniform Secret Sharing Schemes", 34th Cryptology Conf. on Advances in Cryptology (CRYPTO'14), pp. 217-234, 2014.

[12] J. Li, X. Chen, M. Li, J. Li, P.P.C. Lee, and W. Lou, "Secure Deduplication with Efficient and Reliable Convergent Key Management", IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 6, pp. 1615-1625, 2014.

[13] Y. Wang, "Privacy-Preserving Data Storage in Cloud Using Array BP-XOR Codes", IEEE Trans. Cloud Comput. vol. 3, no. 4, pp. 425- 435, 2015.

[14] A. Shamir, "How to share a secret", Communication of the ACM, vol. 22, no. 11, pp. 612-613, 1979.

[15] G.R. Blakley, "Safeguarding cryptographic keys", AFIPS National Computer Conf., vol. 48, pp. 313-317, 1979.

[16] G.R. Blakley, and C. Meadows, "Security of ramp schemes", CRYPTO on Advances in Cryptology, pp. 242-269, 1984.

[17] H. Yamamoto, "On secret sharing systems using (k; L; n) threshold scheme", IEICE Trans. on Fundamentals, vol. J68-A, no. 9, pp.945-952, 1985.

[18] N. CAI, and W. Raymond, "Secure network coding", IEEE Int. Symposium Inf. Theory, 2002.

[19] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "XORs in the air: practical wireless network coding", IEEE Trans. Netw., vol. 16, no. 3, pp. 497-510, 2008.

[20] Z. Yu, Y. Wei, B. Ramkumar, and Y. Guan, "An Efficient Scheme for Securing XOR Network Coding against Pollution Attacks", 28th Conf. on Computer Communication (INFOCOM'09), pp. 406-414, 2009.

[21] A. Khreishah, I.M. Khalil, P. Ostovari, and J. Wu, "Flow-based XOR Network Coding for Lossy Wireless Networks", IEEE Trans.
Wireless Commun. vol. 11, no. 6, pp. 2321-2329, 2012.

[22] A. Kalantari, G. Zheng, Z. Gao, Z. Han, and B. Ottersten, "Secrecy Analysis on Network Coding in Bidirectional Multibeam Satellite Communications", IEEE Trans. Inf. Forensics Security, vol. 10, no. 9, pp. 1862-1874, 2015.

[23] D. Slepian, and J. Wolf, "Noiseless coding of correlated information sources", IEEE Trans. Inf. Theory, vol. 19, no. 4, pp. 471-480, 1973.

[24] S. Cheng, Slepian-Wolf Code Designs. Available: http:// tulsagrad.ou.edu/samuel cheng/ information theory 2010/ swcd.pdf, 2010.

[25] R. Ahlswede, N. Cai, S. Li, and W. Yeung, "Network information flow", IEEE Trans. Inf. Theory, vol. 46, no. 4, pp. 1204-1216, 2000.