IJCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Volume-5, Issue-10 E-ISSN: 2347-2693

Identification of Duplicate Chunks Using Content Approach

Gagandeep Kaur*, Mandeep Singh Devgan

[1*]Department of Information Technology, Chandigarh Engineering College, Mohali, India
[2]Department of Information Technology, Chandigarh Engineering College, Mohali, India

**Corresponding Author: kaurgagandeeparora@gmail.com*

Available online at: www.ijcseonline.org

Received: 21/Sep/2017, Revised: 30/Sep/2017, Accepted: 15/Oct/2017, Published: 30/Oct/2017

Abstract- In this article the implementation of the functions for identification of duplicate chunks based on block, file and content approach have been discussed. The main core of the Deduplication algorithms is chunking and hashing functions. It is also referred as Deduplication granularity. The analysis of these three methods show that the content approach for deduplication is bit slow but the accuracy is good as compared to file and block strategies. It can be seen that the content method of identifying duplicate chunks is about 0.2-0.3% slower but its accuracy is higher by 1-2 % when duplicate finding method of block and file are considered. This work is useful for building duplicate content –aware applications. Especially, when it is used for checking multiple patterns, matching paraphrased content and plagiarism. The proposed methods here can be used for inline as well in the post processing type of Deduplication and it can be extended to include the concept of background and foreground processing.

Keywords- Data Deduplication, Duplicate Chunks, Hashing, Execution Time, Polynomial Chunking.

I. INTRODUCTION

The multi-user environment [1] the user creates a lot of duplicate content/files that unnecessarily waste, a lot of storage space and resources on the cloud (bandwidth, CPU, Hard Disk space etc.). Due to plethora of many Deduplication applications [2] available in the market, it is hard to find an appropriate solution for particular conditions. This is due to the fact that systems of storage are dependent on multiple dynamic factors. These factors become hard to analyze when the storage mechanism is cloud based and the objective is to provide high quality of services [3]. The quality of service in storage is off course measured in terms of speed at which the data can be written and read and how efficiently the system is able to save the space along with multi –tier security. This is necessary also because the storage consolidation in case of direct attached is normally inefficient and when virtualization [4] is taken most of the resources are shared, hence require special attention for managing it. Currently, we are in stage where many applications are already deployed but were not designed for working in shared storage environment .Some applications may behave in a greedy manner and eat up the space, bandwidth and other resources. Then, there are applications that are computational intensive and may starve other applications wanting more I/O resources. Hence, the need for adding mechanism that would save storage [5] , bandwidth and CPU resources Deduplication in one the mechanism by which resources can be optimized properly. But, it will be useful tool if the performance of it is not measured and is not secure. Apparently, it is clear that end to end storage performance matters and it has impact on the I/O workload also. In current in context, the all the storage usage is measure in terms of input/output operations per second or simply IOPs. Typically a cloud server [6] or cluster will have a configurable policy that gives a range of IOPs that can happen in a particular “Flow “. By flow we means a file handler or workload opened by the virtualized system. These policies may be throttle based which normally set minimum threshold or QoS based [3] which cover other aspects also the monitoring. No matter what the policy is the ultimate goal is to save storage with secure I/O operations and Deduplication solutions ‘s purpose is similar in nature with help of chunking functions [7].

Organization of the Paper:

The introductory section gave a brief information on the Data Deduplication concepts and issues involved in this area. In the next section (Related Work), we discuss the various architectures of Deduplication solutions along the parameters that impact the performance. The third section mentions about the gaps and challenges identified after conducting the review. Fourth Section discusses the implementation of work after defining the scope of work. The implementation section gives step wise explanation of the methods employed to accomplish the scope of work. Finally, the paper gives the

Limitation section, followed by conclusion. Last, but not least the future scope of the current research work.

II. RELATED WORK

The following section discusses the various approaches in deduplication a contemporary issues in building block, file and content base deduplication.

Yinjin fu et al [8] According to these authors, the data residing in the cloud can come from users playing around with applications such as MS word or it may come from a specialized cloud service such as interconnected sensor data stream. Even if, Deduplication is applied in many cases a disproportionally large percentage of storage space is occupied by a very small number of large files with very low chunk-level redundancy after file-level dedupe. Secondly, the optimal combination of chunking and hash fingerprinting methods may reduce system overheads on resource-limited personal computing devices but still it needs further attention. This paper discussed the implementation of data duplication algorithm that is application index –aware context for solving such problems. The experiments performed on the prototype shows that the performance of the algorithm in question is better as compared to the previous algorithm in terms of reducing the overhead, shorting the backup window and storage capacity efficiency .This is mainly done by discovering high chunk-level redundancy . The performance of the system was measured in terms of Deduplication efficiency as a mathematical relation of chunk size factor and chunking method.

Atul Katiyar et al [9] in this work the authors have gone beyond a implementing a data duplication application to finding the similarity between the video files. By doing this the researchers were able to configure what proportion of noise during replica detection can be bore .The results claimed in the work shows that by trade off of CPU for storage, a 45% bargain in storage space could be attained, in comparison to 8% returned by system level de-duplication for a dataset collected from video sharing sites on the Web. The authors also present analysis of various tunable factors of the system to optimally tune the system for and quality, compression and performance. The process described in the article shows that the first step taken was generation of video signature, which is later on used for computing the similarity between the various videos.

Waraporn Leesakul et al [10] in this research work, Deduplication work is one the component of the cloud system discussed and implemented. The authors explain in the beginning that due to inherent nature of the cloud, the cloud storage remains in a dynamic mode and the backups needs to be more frequently. And at the same time there is a need to maintain an environment that consumes minimum resources specially the storage. For this the authors designed a method that can help built a dynamic scheme for removing duplicate and redundant data.

Jilin Zhang et al [11] the entire focus of this paper is to discuss the methods by which the sensor data and the image data can management. The formats “VMWare Virtual Machine Disk Format” (VMDK), “Virtual Desktop Infrastructure” (VDI), “QEMU Copy On Write2” (QCOW2), and RAW image formats are more suitable for the IM-dedup system. The authors have used open stack, an open source cloud infrastructure building source code to build and experiment the Deduplication process. The system have a heavy duty servers and client architecture and the concept of Finger Printing has been used to maintain the indexes and references /indexes of the data Chunking is used in single node configuration to maintain and find similarity between the images . During their research, it was also corroborated the virtual image size also impacts the process of Deduplication. Basically, this combined Deduplication technology with the original basic functions of open stack.

Sonam Mandal et al [12] in this research work the authors take advantage of block layer Deduplication operations to boost the speed of the Deduplication algorithm due to the fact that file system helps better in managing the files better and it makes the implementation of the Deduplication operations easier. The authors are using the concept of Hints. Which can be defined as a trigger or condition at which the Deduplication should happen or not at block level? These two hints include: PREFETCH and NODUP. According to the claims of the authors the results of the experiments show that adding the NODEDUP hint to applications like Deduplication can improves the performance by up to 5.3× during the copying unique data. This hint can be extended to other applications, such as those that compress or encrypt. Adding the PREFETCH hint to applications such as Deduplication also improves the copying time by as much as 1.8× because the caching of hashes . Then we do not need to access the metadata of the device to fetch them for writing. Adding hints to macro workloads like File bench’s Fileserver workload also improved throughput by as much as 4.5×.

A. Ragini et al [13] In this research work the authors have worked on problem in which a cloud user is trying to upload a file that already exist on the storage medium. In simple words a duplicate file. To avoid such situations the authors introduced the concept of ALG De-dupe theme on backup services conjointly. This way when a user is unnecessarily trying to transfer the already existing knowledge or a file on consumer editor from the cloud atmosphere the ALG De-dupe algorithms simply transfer the file from consumer editor itself rather than downloading from Cloud surroundings.

Bo Mao et al [14] there are many uses of Deduplication, Data recovery and restoring the data is one of the functions it performs. The authors of this paper have worked on the problem of increasing the speed of the “read operations “during the restoring the data for disaster or for copying purpose from source to another. The main reason for the tardiness is that files or blocks of files are located in a distributed fashion which leads to degradation of the disk read /write operations. To address these issues the authors worked on a SAR, an SSD (solid-state drive)-Assisted Read scheme, that effectively exploits the high random-read performance properties of SSDs and the unique data-sharing characteristic of Deduplication-based storage systems by storing in SSDs the unique data chunks with high reference count, small size, and non-sequential characteristics. In this way, many read requests to HDDs are replaced by read requests to SSDs, thus significantly improving the read performance of the Deduplication based storage systems in the cloud. The extensive trace-driven and VM restore evaluations on the prototype implementation of SAR show that SAR outperforms the traditional Deduplication-based and flash-based cache schemes significantly, in terms of the average response times.

Zhe Sun et al [15] in this work the authors have on “Deduplication –less approach”, which simply means using other than conventional chunking method. The system consist of two –tier and front –end Deduplication component. That have Deduplication algorithm and the backend consist of mass storage system (HDFS), which is primary used for Big Data. The work has been commission on Virtual Server machines for ensuring cloud like simulation. As per the claims of the researcher the systems performs quite well, especially for engineering –oriented application, data and cloud network.

Zhuan Chen et al [16] this is one of the few papers that talks about the saving the data on Flash memory and managing the data using Deduplication application. The data related to sensor, mobile data and cloud are sometimes stored on the flash memories also. Hence, management of I/O read and write operations need flash specific optimization as flash memory is usually constrained. The system implemented by these authors is fault tolerant, efficiency, and the data can be without duplicates for saving space. The implementation consist of a soft updates-style metadata write ordering that maintains storage data consistency without consistency-induced additional I/O. And their experiments various on mobile and server workloads—packages shows lot of advantage in terms of performance results show that Order Merge Dedup can realize 18–63% write reduction on workloads that exhibit 23– 73% write content duplication.

Tao Jiang et al [17] . The authors of this paper have attempted to improve the method cross –client data Deduplication called “MLE2”. The improvement is in terms of reducing the over heads in (non- interactive) zero proof knowledge algorithms use for security. For this, they have used an interactive protocol based on static or dynamic decision trees. The advantage gained from it is, by interacting with clients, the server will reduce the time complexity of Deduplication equality test from linear time to efficient logarithmic time over the whole data items in the database.

III. GAPS & SCOPE OF WORK

Base on the above literature survey. It can be inferred that the current solutions in Deduplication need to undergo change for improving storage mechanism [18] for saving space and there more methods than conventional methods of chunking for building Deduplication applications. An iterative progress can be only be done further after analyzing the existing solutions architectural designs and performances. The purpose of this work is to evaluate the performance of the Deduplication algorithm in various conditions that includes in sync and a sync mode. For a deep view of the work compression factors also be monitored for their impact on the overall performance of the system in different modes .This article however focuses this work on the use of MySQL database as a backend for the work .The system will be content-aware systems and will incorporate methods for improving de-duplication, or single instancing, in storage operations.

IV. IMPLEMENTATION

The propose Deduplication system here is based on well proven methods and workflow. The improved architecture of the application is based on the open source API/ projects. The application is configurable and improvement of the Deduplication on the level of security is required and the evaluation of the application is necessary for further improvement. The table 1 gives the parameters and configurable variables that we will used to build the Deduplication algorithm .The table not only give the value but description also.

Initially, cloud based storage infrastructure has made using a simulator approach. All there ways in which the cloud storage can be made (Public, Private, and Hybrid) has been simulated for gathering cloud based data for storage. In our context, Hybrid cloud [19] with user management module will be activated, which will have Cloud service providers (CSPs) providing data storage services to the Cloud User ‘u’ with specific their package defined in terms of price , storage capacity and other facilities . The storage is file based on object based storage mechanism. There is not Key management Scheme activated [20] for maintaining the security and is out of scope for analysis. Activation of De-Duplication Algorithm along file and block level parameters is done to begin with. In the Deduplication process, unique chunks of data, or byte patterns, are going are identified and stored in MySQL database during a process of analysis. As the analysis continues, other chunks are compared to the stored copy and whenever a match occurs, the redundant chunk [21] is replaced with a small reference that points to the stored chunk. Given that the same byte pattern may occur dozens, hundreds, or even thousands of times (the match frequency is dependent on the chunk size), the amount of data that must be stored or transferred can be greatly reduced.

Table 1. Configurable Parameters of the Improved Deduplication Algorithm

S. No.	**Parameter**	**Description**	**Value(s)**
1.	Delay write flush interval	It is the minimum interval between which the writing of data is done when the resources are easily available.	2400
3	Read cache dirty block	It is an operation that allows reading of data from the cache. The unit of reading data may be file, block or an element.	1024
4	Compression parameter	The system allows zlib type of compression for the said deduplication system. Another parameter which is used in this research work is “level of compression”.	Zlib3
5	Force flush threshold	This operation basically brings flush operation into its act by forcing. In simple words the flush operation is given top priority by the system and other operations loose the write to run in a current memory.	131072
6	Block size	It is referred as the physical record in storage and in deduplication process it is size which gives the information on how many bytes /blocks will be processed. It is configurable parameter in the application. In some systems, chunks are defined by physical layer constraints (e.g. 4KB block size in WAFL). In some systems only complete files are compared, which is called single-instance storage or SIS. The most intelligent (but CPU intensive) method to chunking is generally considered to be sliding-block. In sliding block, a window is passed along the file stream to seek out more naturally occurring internal file boundaries.	4096
7	File size	This parameter gives the size on physical disk of the file. On Disk file size may be less than the actual file size because Deduplication process moved the contents of the file to a common chunk store and replaced the original file with an NTFS reparse point stub and metadata information.	25 KB—200KB
8	File type	The kind of file that it can index, process and DE duplicate	doc, txt, rtf, pdf
10	Snapshot directory	Snapshots taken at different stages of the process of Deduplication gives us mechanism to save the current data and save us from space over utilization.	D:\snaps
11	Delay write cache	It is an object that maintains list of objects that need to be written on a storage medium whose role is to provide cache to the Deduplication system. The main purpose of this object is to identify a condition where resources are ideal and disk writing is possible. Before it writes it means to check file lock mechanism also in multiuser environment.	True or False
12	Cache-sort-interval	Minimum interval time after which the files cached are put under sorting operation. This operation may sort file as per size , date or title etc.	2500
13	Log file directory ,log file Log file-debug = info	It is a directory where information relating to the working and errors is stored in a file.	D:\logs\log.txt
14	Clear cache type	There are typically three kinds of cache namely, MRU (most recently used) LRU (least recently used) RR (random replication)	MRU

A. *Stepwise Explanation*

- Initialization of the parameters

For initiating data Deduplication process many variables need to be configured for its proper working. The system first needs to know the source and the target directories and how to do error handling (fault-tolerance) is done during the process of Deduplication. The system needs to create connection between client and server and identify the storage devices involved. The applications consist of client server architecture where the server creates connection with the client based on configurable IP, ports and storage devices. For this selection of protocol and other threshold are either configured statically or are computed dynamically after assignment of default value. All such factors and variables are mentioned in Table 1.

- Connect Client socket storage space

After the initialization the protocol data unit (PDU) transmission starts with (authentication may be LDAP based) login request and scanning of target iSCSI storage medium begins. If the login is successful a connection is established and files residing in the storage medium are indexed and the process of Deduplication is initialized.

B. *Deduplication Algorithms*

- Algorithm I

```
1. Initialize parameters with default values. Parameters with respect mode, block size, storage location etc.
2. Start I/O buffer readers, make a list of files and read first 1024 bytes from each file, using while loop.
3. Continue to read until reach EOF as per configure block size
   3.1 Invoke Hash Function and for each block size generate
       Initialize data structure for storing hash values for each block
For each block, find the first block and mark it root, For each root /parent node find store/ values of has for each node using BST
For each Tree of Blocks {
       If hash is true
      Compute Node
      Add(Link List Data Structure)
     Add End Link (Singly Linked List)
   Else
  Add the node in BST
    End if
}
4. Identify the duplicate block using hash value.
4.1 Computer Accuracy of the algorithm and execution time
```

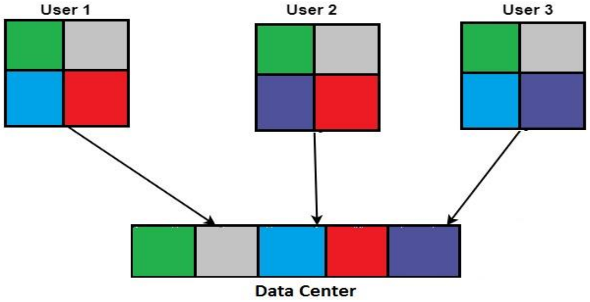


Figure 1. Schematic View of Block Level Diagram

- 4.2.2 Algorithm II

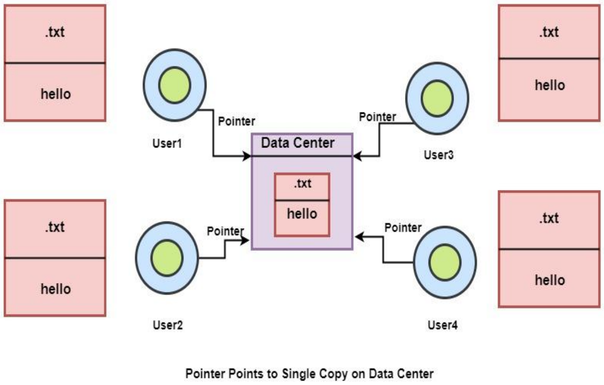


Figure 2. Schematic View of Deduplication Algorithm

```
1.     Initialize various resources and identify the file objects to be indexed for Deduplication process.
2.   For each 'f' file in file System.
Generate Pointer and has with respect to each byte block.

       End

1.   For each p Unique Pointer ,
        Identify similar bock using hash key
        Hash Key = Already Exist.
        If(Similarity ==True)
        Add Pointer To block(p)
       Else
        Ignore
       End For.

Compute Deduplication Accuracy.
```

- Algorithm III

The algorithm I and II can be configured for constant and variable block size. Similarly, the content based strategy can also be configured for variable or fix size content. However, in this research work our focus is find the performance with respect to execution of Deduplication function and accuracy with which duplicate block are found by each process. The contemporary literature shows that basic sliding methods and multi-threshold methods are popular. In this current work a content aware Deduplication application has been built and to make the process efficient and accuracy, sliding window along with signatures of the chunk using polynomial and modulus arithmetic has been implemented. The algorithm Computes (a mod b) using synthetic division where a and b represent polynomials in Galois field (2^k). As the loop iterates through the data/content chunks it computes these polynomials that act as signature of the content block. The data is stored as per block. But each block must lie in a computable sector of the content. The information is stored in the MySQL database as follows.

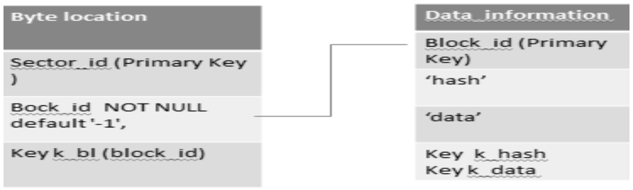


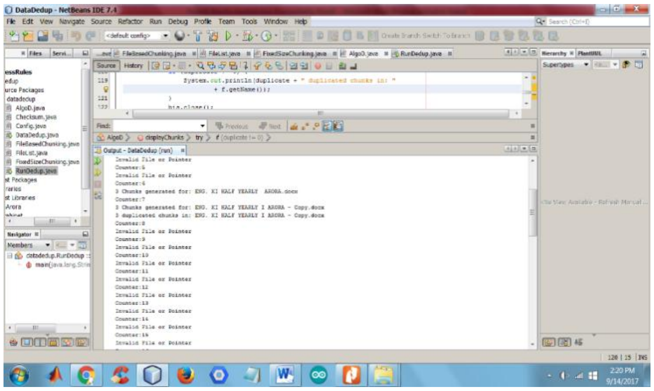
Figure 3. MySQL Data Base Structure for storing information on Hash & data.

- Receive Input/ Output Stream

The data Deduplication application now start login debug information of system, application and devices involved transmission in input and output streams. The system creates a data store which maintains the files under process and also keeps temporary pointers to the duplicate data chunks. At this stage storage type is also checked and compression parameters are initialized for further processing.

C. Evaluation

Chunking Output- In this chunking and identify the duplicated data chunks are implemented and are evaluated. It can be seen, the chunking function took 844 seconds to execute 40 file, out of which many were invalid files and in valid file it was able to detect the duplicate chunks also. Chunking Function [7] took 844 mill seconds to execute. (40 File were processed) when the average size of the files is 25 KB.



```
System.out.println(duplicate + " duplicated chunks in: "
        + f.getName());
}
bis.close();
```

```
Invalid File or Pointer
Counter:5
Invalid File or Pointer
Counter:6
3 Chunks generated for: ENG. XI HALF YEARLY  ARORA.docx
Counter:7
3 Chunks generated for: ENG. XI HALF YEARLY I ARORA - Copy.docx
3 duplicated chunks in: ENG. XI HALF YEARLY I ARORA - Copy.docx
Counter:8
Invalid File or Pointer
Counter:9
Invalid File or Pointer
Counter:10
Invalid File or Pointer
Counter:11
Invalid File or Pointer
Counter:12
Invalid File or Pointer
Counter:13
Invalid File or Pointer
Counter:14
Invalid File or Pointer
Counter:15
Invalid File or Pointer
```

Figure 4. Chunking Function Output

1	Number of Files	Average Size of File	Time in mill seconds (File)	Time in mill seconds (Block)	Time in mill seconds (content)
2	40	25 KB	814	820	844
3	50	25 KB	862	882	902
4	60	30 KB	980	920	1010
5	70	35 KB	1129	1119	1209
6	80	40 KB	1610	1540	1690
7	90	45 KB	1711	1737	1777
8	100	50 KB	1799	1787	1807

- Accuracy

Block	File	Content
88.2 %	88.91 %	90.89 %

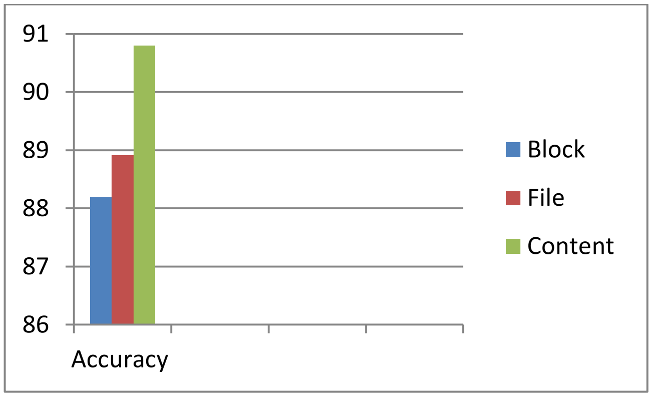


Figure 5. Accuracy of the all the three methods.

It is clear that the proposed algorithm is taking much less time in processing the chunks as it is based on the custom and highly tunable parameters. The analysis of these three methods show that the content approach for Deduplication is better as compared to file and block in certain conditions . It can be seen that the proposed method of identifying duplicate chunks is about -0.2-0.3% slower but its accuracy is higher in terms of execution of the of chunking function working, especially when it is used for checking multiple pattern matching and plagiarism .Our algorithm works faster as it avoids modulo n operation (% in C like languages) use mask n - 1, where n is 2^k, include the operations for the hash table lookup and produces good hash with a non-prime moduli also.

V. LIMITATIONS

This work has ignored the impact of various security functions that may be affecting the performance of the overall system. Key management consist of multiple steps that also have significant impact on the working of the system. The Key management schemes that are based on "challenge" or zero proof algorithms have additional steps of Proof of Ownership and Athematic validity.). The data is also prone to attacks due to the fact that it is stored on the geo-distributed cloud servers. In such arrangement, of key management is usually done using their 'challenges'. In such key management scheme, the ownership and arithmetic validity of keys keeps changing dynamically. There is a need for reducing and improving the overhead of such key management schemes. There is a scope of improvement in reducing computational time, turnaround time and overhead in methods of computations of Key Schema management by using methods such as Nikhilam Sutra. .The steps also need to analyze for the sake of full performance evaluation.

VI. CONCLUSION

In this research work, we were able to conduct a review of the different kinds of Deduplication scenarios. And, it was found that there are mainly three possible levels (File, Block, and Content) in finding the duplicate content. Content Deduplication method has application in finding similarity index used in checking plagiarism in the paper. It was also found that finding a duplicate block or simply a duplicate file may take little time as compare to the content based duplication finding method but need of the hour is content .Moreover , it is clear that approach of Deduplication here work well be the content agnostic data .as it does not require any application data formats. By using backend (MySQL) as storage medium to store hash and other indexing meta information works well where that data is not too big in volume .In fact , the use of MySQL positively impact performance and fault tolerance.

VII. FUTURE SCOPE

The degree of issues related to the implementation of Crypto Algorithms in terms of mathematics is not that difficult as compared to embracing and applying to current technological scenarios. Decentralized Anonymous Credentials validity and arithmetic validity is need of the hour and human sensitivity to remain safe is critical. In certain cases, the need to eliminate trusted credential issuers can help to reduce the overhead with-out compromising the security level whole running Deduplication process. Many algorithms for exponentiation do not provide defense against side-channel attacks when Deduplication process is run over a network. An attacker observing the sequence of squaring and multiplications can (partially) recover the exponent involved in the computation. Many methods compute the secret key based on Re-cursive method, which has more overhead as compared methods that are vectorized. Some of the vectorized implementations of such algorithms can be improved by reducing the number of steps with one line computational methods, especially when the powers of the exponent are smaller than 8. There is a scope of improvement in reducing computational overhead in methods of computations of arithmetic Validity methods by using methods such as Nikhilam Sutra, Karatsuba.

REFRENCES

[1] K. Ren, C. Wang and Q. Wang, "Security Challenges for the Public Cloud," *IEEE Internet Computing,* vol. 16, pp. 69-73, 2012.

[2] Y. Fu, H. Jiang and N. Xiao, "AA-Dedupe: An Application-Aware Source Deduplication Approach for Cloud Backup Services in the Personal Computing Environment," in *2011 IEEE International Conference on Cluster Computing*, 2011, pp. 112-120.

[3] J. Malhotra, J. Bakal and L. G. Malik, "Caching: QoS Enabled Metadata Processing Scheme for Data Deduplication," in *Proceedings of the International Congress on Information and Communication Technology: ICICT 2015, Volume 2*, Springer Singapore, 2016, pp. 545-553.

[4] J. Xiao, Z. Xu and H. Huang, "Security implications of memory deduplication in a virtualized environment," in *2013 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2013, pp. 1-12.

[5] D. Harnik, B. Pinkas and A. S.-. Peleg, "Side Channels in Cloud Services: Deduplication in Cloud Storage," *IEEE Security Privacy,* vol. 8, pp. 40-47, 2010.

[6] J. Stanek, A. Sorniotti and E. Androulaki, "A Secure Data Deduplication Scheme for Cloud Storage," in *Financial Cryptography and Data Security: 18th International Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014, Revised Selected Papers*, Springer Berlin Heidelberg, 2014, pp. 99-118.

[7] Y. C. Moon, H. M. Jung, C. Yoo and Y. W. Ko, "Data Deduplication Using Dynamic Chunking Algorithm," in *Computational Collective Intelligence. Technologies and Applications: 4th International Conference, ICCCI 2012, Ho Chi Minh City, Vietnam, November 28-30, 2012, Proceedings, Part II*, Springer Berlin Heidelberg, 2012, pp. 59-68.

[8] Y. Fu, H. Jiang and N. Xiao, "Application-Aware Local-Global Source Deduplication for Cloud Backup Services of Personal Storage," *IEEE Transactions on Parallel and Distributed Systems,* vol. 25, pp. 1155-1165, 2014.

[9] A. Katiyar and J. Weissman, "ViDeDup: An Application-Aware Framework for Video De-duplication," in *HotStorage*, 2011.

[10] W. Leesakul, P. Townend and J. Xu, "Dynamic data deduplication in cloud storage," in *Service Oriented System Engineering (SOSE), 2014 IEEE 8th International Symposium on*, 2014, pp. 320-325.

[11] J. Zhang, S. Han, J. Wan, B. Zhu, L. Zhou, Y. Ren and W. Zhang, "IM-Dedup: An Image Management System Based on Deduplication Applied in DWSNs," *International Journal of Distributed Sensor Networks,* vol. 9, 2013.

[12] S. Mandal, G. Kuenning, D. Ok, V. Shastry, P. Shilane, S. Zhen, V. Tarasov and E. Zadok, "Using Hints to Improve Inline Block-layer Deduplication," in *FAST*, 2016, pp. 315-322.

[13] A. Ragini and V. Nararaj, "Exploiting The Chunk Redundancy In Cloud Backup Using Alg-De-Duplication Technique," pp. 18-20,

2015.

[14] B. Mao, H. Jiang, S. Wu, Y. Fu and L. Tian, "Read-performance optimization for deduplication-based storage systems in the cloud," *ACM Transactions on Storage (TOS)*, vol. 10, p. 6, 2014.

[15] S. Zhe , S. Jun and Y. Jianming, "A novel approach to data deduplication over the engineering-oriented cloud systems," *Integrated Computer-Aided Engineering,* vol. 20, pp. 45-57, 2013.

[16] Z. Chen and K. Shen, "OrderMergeDedup: Efficient, Failure-Consistent Deduplication on Flash," in *FAST*, 2016, pp. 291-299.

[17] T. Jiang, X. Chen and Q. Wu, "Secure and Efficient Cloud Data Deduplication With Randomized Tag," *IEEE Transactions on Information Forensics and Security,* vol. 12, p. 3, 2017.

[18] J. Hur, D. Koo, Y. Shin and K. Kang, "Secure data deduplication with dynamic ownership management in cloud storage," *IEEE Transactions on Knowledge and Data Engineering,* vol. 28, pp. 3113-3125, 2016.

[19] S. Mishra and P. Sharma, "Hybrid Cloud Data Security Model Using Splitting Technique," *International Journal of Computer Sciences and Engineering* , vol. 4, no. 6, 2016.

[20] Y. Zhou, D. Feng and W. Xia, "SecDep: A user-aware efficient fine-grained secure deduplication scheme with multi-level key management," in *2015 31st Symposium on Mass Storage Systems and Technologies (MSST)*, 2015, pp. 1-14.

[21] Y. Tan, H. Jiang and D. Feng, "CABdedupe: A Causality-Based Deduplication Performance Booster for Cloud Backup Services," in *2011 IEEE International Parallel Distributed Processing Symposium*, 2011, pp. 1266-1277.

Author's Profile

Gagandeep Kaur is a student in Department of Information Technology at Chandigarh Engineering College, Landran (Mohali). She completed her B.Tech in North West Institute of Engineering & Technology, Dhudike (Moga) in year 2014.

Mandeep Singh Devgan is an Assistant Professor in Department of Information Technology at Chandigarh Engineering College, Landran (Mohali). He completed his B.Tech in year 2006. He received his M.Tech degree in Computer Science and Engineering in year 2010. He has been in teaching profession for the last 11 years. His areas of Interest are cloud computing, Network Security, Login Authenticity.