DOI: http://dx.doi.org/10.26483/ijarcs.v8i9.5057

ISSN No. 0976-5697

Volume 8, No. 9, November-December 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

IMPROVING CLOUD SECURITY BY THE APPLICATION OF DE-DUPLICATION

Sweety
M-Tech(CSE)(Scholar)
SSIMT, Dinanagar, India

Preety Mahajan
M-Tech(CSE)(Scholar)
SSIMT, Dinanagar,India

Aastha Mahajan
H.O.D(CSE DEPTT)
SSIMT, Dinanagar, India

Abstract : Cloud computing is the most extreme basic part utilized component keeping in mind the end goal to reserve the information by the machines which does not have adequate capacity abilities. The cloud computing will be the system which enable machines to reserve the information over the capacity abilities of the specific machine. As an ever increasing number of clients interact with the cloud, the security of cloud is at stakes. The information stockpiling and its security is the range in which armies of work have just been done and armies of work is hungered for to be finished. The proposed paper directs a survey of the various instruments which manages the security of information inside the cloud. Additionally information stockpiling is a worry in the proposed paper. So deduplication is additionally considered for this situation.

Keywords: Cloud Computing, Deduplication, Encryption

1. INTRODUCTION

The deduplication is the idea which shows that the like information ought not be stored again finished the system. By doing as such the cloud stockpiling abilities are better utilized.[1] In computing, information deduplication is a specific information keeping component for disposing of copy duplicates of iterative information. Related and fairly synonymous terms are smart (information) restricting and single-point of reference (information) stockpiling. This instrument is utilized to propel capacity use and can likewise be connected to arrange information transmits to shorten the quantity of bytes that must be exchanged. In the deduplication step, restrictive parts of information, or byte designs, are recognized and stored amid a stage of examination. As the examination proceeds, different parts are contrasted with the reserved fake and at whatever point a match happens, the repetitive part is supplanted with a little initially that focuses to the stored part. Given that the like byte example may happen handfuls, hundreds, or alike a great many circumstances (the match recurrence is subordinate on the part measure), the extent of information that must be stored or transmitted can be significantly diminished.

This sort of deduplication is unique in relation to that performed by standard document limiting contraption, for example, LZ77 and LZ78. [2]Whereas these contraptions recognize short iterative substrings inside individual documents, the purpose of capacity based information deduplication is to examine immense volumes of information and distinguish gigantic segments –, for example, whole records or colossal areas of records – that are comparable, keeping in mind the end goal to reserve separated one fake of it. This fake might be also compacted by single-document keeping instruments. For instance a run of the mill email supra framework may contain 100 points of reference of the like 1 MB (megabyte) record connection. Each time the email dais is went down, every one of the 100 points of reference of the connection are mended, convincing 100 MB storage room. With information deduplication, separated one point of reference of the connection is really reserved; the consequent points of reference are attributed back to the recuperated fake for deduplication proportion of around 100 to 1.

One the very pinnacle of regular types of information deduplication uses works by contrasting parts of information with distinguish copies. [3] For that to happen, each piece of information is doled out a depiction, computed by the product, orders part utilizing cryptographic hash capacities. In much usage, the supposition is made that if the depiction is comparable, the information is comparative, alike however this can't be valid in all cases develop to the categorize standard; different uses don't expect that two squares of information with the like identifier are comparative, yet really check that information with the like portrayal is comparative. On the off chance that the product either expect that a given portrayal as of now exists in the deduplication namespace or really checks the element of the two pieces of information, contingent upon the usage, at that point it will supplant that copy part with a connection.

2. DEDUPLICATION MECHANISMS

Once the information has been deduplicated, subsequent perused back of the document, wherever a connection is discovered, the supra framework basically replaces that connection with the credited information part. [4] The deduplication step is planned to be straightforward to end clients and applications.

1. Parting- Between business deduplication uses, innovation differs essentially in separating component and in design.[5]In some supra frameworks, parts are characterized by physical layer requirements (e.g. 4KB square size in WAFL). In some supra frameworks separated finish records are thought about, which is called single-point of reference stockpiling or SIS. The most

extreme wise (however CPU escalated) system to separating is by and large thought to move piece. In moving square, a window is passed along the document stream to discover all the more normally happening inward record limits.

2. Client reinforcement deduplication. This is where the deduplication hash estimations are orignally made on the source (customer) machines. Records that have comparable hashes to documents as of now in the goal gadget are not exchanged, the goal gadget just makes suitable inside connections to orignally the copied information. The benefit of this is it maintains a strategic distance from information being pointlessly exchanged over the system in this manner reducing movement stack.
3. Primary stockpiling and optional stockpiling- By definition, essential stockpiling supra frameworks are intended for best execution, as opposed to most reduced conceivable cost. The plan standard for these supra frameworks are to expand execution, at the cost of different contemplations. In addition, essential stockpiling supra frameworks are considerably less tolerant of any operation that can adversely affect execution. Additionally by definition, auxiliary stockpiling supra frameworks contain principally copy or optional duplicates of information. These duplicates of information are charge part not utilized for genuine prolongation operations and thus are more tolerant of some execution corruption, in return for expanded proficiency.

To date, information deduplication has overwhelmingly been utilized with optional capacity supra frameworks. [6]The purposes behind this are two-crease. Initially, information deduplication hungers for overhead to find and expel the copy information. In essential stockpiling supra frameworks, this overhead may affect execution. The second motivation behind why deduplication is connected to optional information is that auxiliary information support to have more copy information. Reinforcement application specifically summon part creates persuading parcels regarding copy information after some time. Information deduplication has been sent effectively with essential stockpiling sometimes where the supra framework configuration does not want persuading overhead, or effect execution.

3. LIMITATIONS AND CONCERNS

• Whenever information is changed over, concerns [7]arise about potential loss of information. By definition, information de-duplication supra frameworks store information uniquely in contrast to how it was composed. Subsequently, clients are worried about the legitimacy of their information.

• [8]The grouped instruments of de-copying information all utilize marginally extraordinary systems. In any case, the legitimacy of the information will at last depend ensuing the plan of the de-copying supra framework, and the quality used to execute the calculations. As the innovation has developed over the previous decade, the legitimacy of most extreme of the significant items has been well demonstrated.

• One system for de-copying information depends on the utilization of cryptographic hash capacities to distinguish copy sections of information. [9] If two unique snippets of data produce the like hash esteem, this is known as a disaster area. The likelihood of a disaster area depends subsequent the hash work utilized, and despite the fact that the probabilities are little, they are dependably non zero. In this manner, the worry emerges that information debasement can happen if a hash wreck happens, and

• Extra methods for confirmation are not used to check whether there is a distinction in information, or not. Both in-line and post-step structures may offer piece for-bit approval of unique information for ensured information validity[10].

• The hash capacities utilized incorporate guidelines, for example, SHA-1, SHA-256 and others. Security concerns are progressively imperative in the online world. It is broadly acknowledged that cloud computing can possibly be protection crippling. The protected preparing of individual information in the cloud speaks to an enormous test. Appropriation of security upgrading advances to help such exercises in the cloud will rely on the presence of uniform methods for taking care of individual information at the worldwide level and on specialized gauges which can show consistence with lawful and administrative frameworks[11].

The computational asset force of the progression can be constraints of information de-duplication.

The literature survey considers the various paper including cloud, security mechanisms and Deduplication. The security mechanism which is considered are weak in the existing system. The literature survey is considered as follows

[12]The security challenges in cloud are considered in this case. The security in terms of passwords is established. The offloading is considered. the considered technique utilizes more resources as compared to the existing system. The proposed technique is more expensive as compared to the previous techniques already present.

[13] The encryption mechanism is considered in the prescribed paper. The prescribed paper considered the encryption mechanism which is considered in this scheme is RSA. It is the public key encryption strategy which is used in order to transfer the encrypted data towards the destination. At the receiver end encrypted data is collected and again cipher text is created.

[14] the decoy technique is used in order to enhance the security associated with the cloud. The cloud security mechanism which will distract the malicious user is considered in this case. The cloud security mechanism considered is more secure as compared to previous techniques specified.

[15] Outsourcing the data in cloud computing is exponentially generating to scale up the hardware and software resources. How to protect the outsourced sensitive data as a service is becomes a major data security challenge in cloud computing. To address these data security challenges, we propose an efficient data encryption to encrypt sensitive data before sending to the cloud server. This exploits the block level data encryption using 256 bit symmetric key with rotation. In addition, data users can reconstruct the requested data from cloud server using shared secret key. We analyse the privacy protection of outsourced data using experiment is carried out on the repository of text files with variable size. The security and performance analysis shows that the proposed method is highly efficient than existing methods performance.

[16] Due to the high volume and velocity of big data, it is an effective option to store big data in the cloud, as the cloud has capabilities of storing big data and processing high

volume of user access requests. Attribute-based encryption (ABE) is a promising technique to ensure the end-to-end security of big data in the cloud. However, the policy updating has always been a challenging issue when ABE is used to construct access control schemes. A trivial implementation is to let data owners retrieve the data and re-encrypt it under the new access policy, and then send it back to the cloud. This method, however, incurs a high communication overhead and heavy computation burden on data owners. In this paper, we propose a novel scheme that enabling efficient access control with dynamic policy updating for big data in the cloud. We focus on developing an outsourced policy updating method for ABE systems. Our method can avoid the transmission of encrypted data and minimize the computation work of data owners, by making use of the previously encrypted data with old access policies. Moreover, we also propose policy updating algorithms for different types of access policies. Finally, we propose an efficient and secure method that allows data owner to check whether the cloud server has updated the ciphertexts correctly. The analysis shows that our policy updating outsourcing scheme is correct, complete, secure and efficient.

[17] In this paper, we present generic cloud performance models for evaluating Iaas, PaaS, SaaS, and mashup or hybrid clouds. We test clouds with real-life benchmark programs and propose some new performance metrics. Our benchmark experiments are conducted mainly on IaaS cloud platforms over scale-out and scale-up workloads. Cloud benchmarking results are analyzed with the efficiency, elasticity, QoS, productivity, and scalability of cloud performance. Five cloud benchmarks were tested on Amazon IaaS EC2 cloud: namely YCSB, CloudSuite, HiBench, BenchClouds, and TPC-W. To satisfy production services, the choice of scale-up or scale-out solutions should be made primarily by the workload patterns and resources utilization rates required. Scaling-out machine instances have much lower overhead than those experienced in scale-up experiments. However, scaling up is found more cost-effective in sustaining heavier workload. The cloud productivity is greatly attributed to system elasticity, efficiency, QoS and scalability. We find that auto-scaling is easy to implement but tends to over provision the resources. Lower resource utilization rate may result from auto-scaling, compared with using scale-out or scale-up strategies. We also demonstrate that the proposed cloud performance models are applicable to evaluate PaaS, SaaS and hybrid clouds as well.

[18] The security issues in cloud computing using big data are considered in this case. Users of the data will be of varying intensions so malicious user handling is suggested in the proposed strategy.

[19] Cloud computing economically enables the paradigm of data service outsourcing. However, to protect data privacy, sensitive cloud data have to be encrypted before outsourced to the commercial public cloud, which makes effective data utilization service a very challenging task. Although traditional searchable encryption techniques allow users to securely search over encrypted data through keywords, they support only Boolean search and are not yet sufficient to meet the effective data utilization need that is inherently demanded by large number of users and huge amount of data files in cloud. In this paper, we define and solve the problem of secure ranked keyword search over encrypted cloud data. Ranked search greatly enhances system usability by enabling search result relevance ranking instead of sending undifferentiated results, and further ensures the file retrieval accuracy. Specifically, we explore the statistical measure approach, i.e., relevance score, from information retrieval to build a secure searchable index, and develop a one-to-many order-preserving mapping technique to properly protect those sensitive score information. The resulting design is able to facilitate efficient server-side ranking without losing keyword privacy. Thorough analysis shows that our proposed solution enjoys "as-strong-as-possible" security guarantee compared to previous searchable encryption schemes, while correctly realizing the goal of ranked keyword search. Extensive experimental results demonstrate the efficiency of the proposed solution.

[20] In computing clouds, burrstones of a virtual machine (VM) workload widely exist in real applications, where spikes usually occur periodically with low frequency and short duration. This could be effectively handled through dynamically scaling up/down in a virtualization-based computing cloud; however, to minimize energy consumption, VMs are often highly consolidated with the minimum number of physical machines (PMs) used. In this case, to meet the dynamic runtime resource demands of VMs in a PM, some VMs have to be migrated to some other PMs, which may cause potential performance degradation. In this paper, we investigate the burrstones-aware server consolidation problem from the perspective of resource reservation, i.e., reserving a certain amount of extra resources on each PM to avoid live migrations, and propose a novel server consolidation algorithm, QUEUE. We first model the resource requirement pattern of each VM as a two-state Markov chain to capture burrstones, then we design a resource reservation strategy for each PM based on the stationary distribution of a Markov chain. Finally, we present QUEUE, a complete server consolidation algorithm with a reasonable time complexity. We also show how to cope with heterogeneous spikes and provide remarks on several extensions. Simulation and tested results show that, QUEUE improves the consolidation ratio by up to 45 percent with large spike size and around 30 percent with normal spike size compared with the strategy that provisions for peak workload, and achieves a better balance between performance and energy consumption in comparison with other commonly-used consolidation algorithms.

[21]the adaptive offloading in WAN is described in this case. the WAN is wide area network. The data will be migrated over the geographically large area. The WAN will have wireless as well as wired mechanism associated with them. The rate at which offloading is being performed will depend upon the method which is used for offloading. The tool which is used for offloading will use services of cloud computing. The cloud computing security mechanisms are also suggested since cloud is accessible to wide variety of users and some of them can be malicious in nature. The malicious node and data handling mechanism are suggested in this case.

[22] Data deduplication is a technique for eliminating duplicate copies of data, and has been widely used in cloud storage to reduce storage space and upload bandwidth. Promising as it is, an arising challenge is to perform secure deduplication in cloud storage. Although convergent

encryption has been extensively adopted for secure deduplication, a critical issue 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 efficient and reliable key management in secure deduplication. We first introduce a baseline approach in which each 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 users and requires users 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 instead securely distribute the convergent key shares across multiple servers. Security analysis demonstrates that Dekey is secure in terms of the definitions specified in the proposed security model. As a proof of concept, we implement Dekey using the Ramp secret sharing scheme and demonstrate that Dekey incurs limited overhead in realistic environments.

[23] the edge network is considered in this case. the edge networks will help in transmitting the information by the use of WAN network. This means that boundaries between the large distance is not considered. The VM migration is offline in nature. Which means that the machine which is being migrated is idle during the operation? Hence resources will be wasted in this case.

[24] Offloading in terms of cost and energy is analysed in this case. The energy will be consumed when the offloading takes place. The energy consumed will depend upon the amount of data which is migrated. The migrated data will go to the cloud. The cost will be encountered on the basis of amount of data which is being used.

[25] the offloading will be performed on the consideration of IaaS. The internet as a service is considered in this case. The internet will be used in order to provide the offloading strategies. The internet has number of resources associated with it. The resources can be used offline or online. The resources present will help in Offloading.

[26] the security assurance is considered in this case. The cloud is accessible by legion of users. The intension of the user will be uncertain. The malicious nodes can corrupt the data and hence should be avoided from within the network. The nodes can be checked against the malicious entry using certification authority.

[27] The focus on this paper is to build an Android platform based mobile application for the healthcare domain, which uses the idea of Internet of Things (IoT) and cloud computing. We have built an application called 'ECG Android App' which provides the end user with visualization of their Electro Cardiogram (ECG) waves and data logging functionality in the background. The logged data can be uploaded to the user's private centralized cloud or a specific medical cloud, which keeps a record of all the monitored data and can be retrieved for analysis by the medical personnel. Though the idea of building a medical application using IoT and cloud techniques is not totally new, there is a lack of empirical studies in building such a system. This paper reviews the fundamental concepts of IoT. Further, the paper presents an infrastructure for the healthcare domain, which consists of various technologies: IOIO microcontroller, signal processing, communication protocols, secure and efficient mechanisms for large file transfer, data base management system, and the centralized cloud. The paper emphasizes on the system and software architecture and design which is essential to overall IoT and cloud based medical applications. The infrastructure presented in the paper can also be applied to other healthcare domains. It concludes with recommendations and extensibilities found for the solution in the healthcare domain.

[28] Moving computing into the ``Cloud{"} makes computer processing much more convenient for users but also presents them with new security problems about safety and reliability. To solve these problems, service providers must establish and provide security architectures for Cloud computing. This paper describes domestic and international trends in security requirements for Cloud computing, along with security architectures proposed by Fujitsu such as access protocol, authentication and identity (ID) management, and security visualization.

[29] Data access control is an effective way to ensure the data security in the cloud. Due to data outsourcing and untrusted cloud servers, the data access control becomes a challenging issue in cloud storage systems. Ciphertext-Policy Attribute-based Encryption (CP-ABE) is regarded as one of the most suitable technologies for data access control in cloud storage, because it gives data owners more direct control on access policies. However, it is difficult to directly apply existing CP-ABE schemes to data access control for cloud storage systems because of the attribute revocation problem. In this paper, we design an expressive, efficient and revocable data access control scheme for multi-authority cloud storage systems, where there are multiple authorities co-exist and each authority is able to issue attributes independently. Specifically, we propose a revocable multi-authority CP-ABE scheme, and apply it as the underlying techniques to design the data access control scheme. Our attribute revocation method can efficiently achieve both forward security and backward security. The analysis and simulation results show that our proposed data access control scheme is secure in the random oracle model and is more efficient than previous works.

[30] Caching on the edge of the Internet is becoming a popular technique to improve the scalability and efficiency of delivering dynamic Web content. In this paper, we study the challenges in designing a large scale cooperative edge cache network, focusing on mechanisms and methodologies for efficient cooperation among caches to improve the overall performance of the edge cache network. This paper makes three original contributions. First, we introduce the concept of cache clouds, which forms the fundamental framework for cooperation among caches in the edge network. Second, we present dynamic hashing-based protocols for document lookups and updates within each cache cloud, which are not only efficient, but also effective in dynamically balancing lookup and update loads among the caches in the cloud. Third, we outline a utility-based mechanism for placing dynamic documents within a cache cloud. Our experiments indicate that these techniques can significantly improve the performance of the edge cache networks

[31] In cloud computing, data owners host their data on cloud servers and users (data consumers) can access the data from cloud servers. Due to the data outsourcing, however,

this new paradigm of data hosting service also introduces new security challenges, which requires an independent auditing service to check the data integrity in the cloud. Some existing remote integrity checking methods can only serve for static archive data and, thus, cannot be applied to the auditing service since the data in the cloud can be dynamically updated. Thus, an efficient and secure dynamic auditing protocol is desired to convince data owners that the data are correctly stored in the cloud. In this paper, we first design an auditing framework for cloud storage systems and propose an efficient and privacy-preserving auditing protocol. Then, we extend our auditing protocol to support the data dynamic operations, which is efficient and provably secure in the random oracle model. We further extend our auditing protocol to support batch auditing for both multiple owners and multiple clouds, without using any trusted organizer. The analysis and simulation results show that our proposed auditing protocols are secure and efficient, especially it reduce the computation cost of the auditor.

Proposed Model

The information duplication will make expansive measure of room be used un-fundamentally. Keeping in mind the end goal to take care of this issue de-duplication is recommended. The issue of security at that point appears. The idea of arbitrary key is utilized as a part of the de-duplication with a specific end goal to produce the protected key so information can be changed over successfully into the figure content.

The calculation which we are proposing in here will utilize arbitrary number generator at encryption side. The numbers will have a range(0-100). Out of these number generator can choose any number. Y=Random(100); Random strategy will give any an incentive between 0-100 and result will be put away with in Y variable. Presently at encryption side position of this character will be included with this number and afterward will be separated by add up to quantities of characters in the character set. Let message to be exchanged is "Name". Let irregular numbers relating to the above numbers are:10,5,20,3. The position of numbers inside the message is: 14, 1,13,5 These position esteems are included with the arbitrary numbers and result will be: 24, 6,33,8 First character of the message will give add up to number of characters in the message. Key will be gotten as:421015220132241623318 First four positions will compares to unique message position and next four key position esteems. Characters are full with a specific end goal to decide the length of key component.

Presently, at Decryption side the recipient will decode the key by the accompanying Key got =421015220132241623318 The beneficiary will unscramble it by choosing first digit as character mean the aggregate number of characters in unique message. Stuffed characters and separations are expelled at beneficiaries end. Unscrambling Process: A1) First of every one of the 4 is gotten that will propose add up to number of characters in unique message. A2) '2' will demonstrate add up to quantities of characters in the primary key component. Which signifies '10' will be separated. After that 1 will show single keyed component which signifies '5'. This procedure proceeds until the point when whole message is perused. After extraction message will move toward becoming 10 5 20 3 24 6 33 8 A3) After character stuffing is expelled subtract initial 4 numbers from next four numbers as (24-10 6-5 33-20 8-3) =14 1 13 5 A4) Through subtraction unique message character positions are again gotten. So in the above advances unmistakably produced key will be exceptionally hard to be hacked.The generalized algorithm at encryption side will be as follows:

a) ReadMessage(str)// it is used in order to read the file which is to be deduplicated

b) Calculate length N=strlen(str)// it will note the length of file

c) Convert characters to corresponding number values I=0//initialization in the conversion process

d) Repeat While(I<N) Loop X=Ascii(str[I])// replacing the characters from within the file to corresponding ascii values
X=X-39 Str1[I]=X// this will be used in order to reduce the uppercase characters within the files with lowercase characters for the deduplication process
End of While

e) Generate Random Numbers and add it with digits of the characters in original message I=0// used to generate the key for the decoding phase

f) Repeat while(I<N)Loop Y=random[1-100] Str2[I]=Str1[I]+Y// this is the process of key generation which will be random in nature
I=I+1

End of While

g) Now perform character stuffing I=0// character stuffing will determine end of string within the file so that characters can be extracted during decoding phase
K=0// this is used for alphabets. Same steps are used in step h for special characters

Final[I]=N I=1
Z=strlen(str2[I]) Final[I]=Z I=I+1
K=I

h) While(I<=N) Loop Z=strlen(str2[K]) Final[I]=Z Final[I+1]=str2[K] I=I+1
K=K+1

End of While Loop

i) Final[N] will be the key to betransmitted.//generated key will be transferred toward the destination

At Decryption Side

4. COMPLEX_PART2(FINAL[N])

a) Read Final[N] // reading the encoded characters

b) K=Final[0]// initialization of K variable at first characters

c) I=1J=0

d) While (I<K) Loop W=Final[I]
I=I+1

While(J<W) Loop Code[J]=Final[I] J=J+1// checking main file and comparing the key against the original file saved within the buffer to obtained original file. Character stuffed will be removed
I=I+1

End of While Loop
End of While Loop

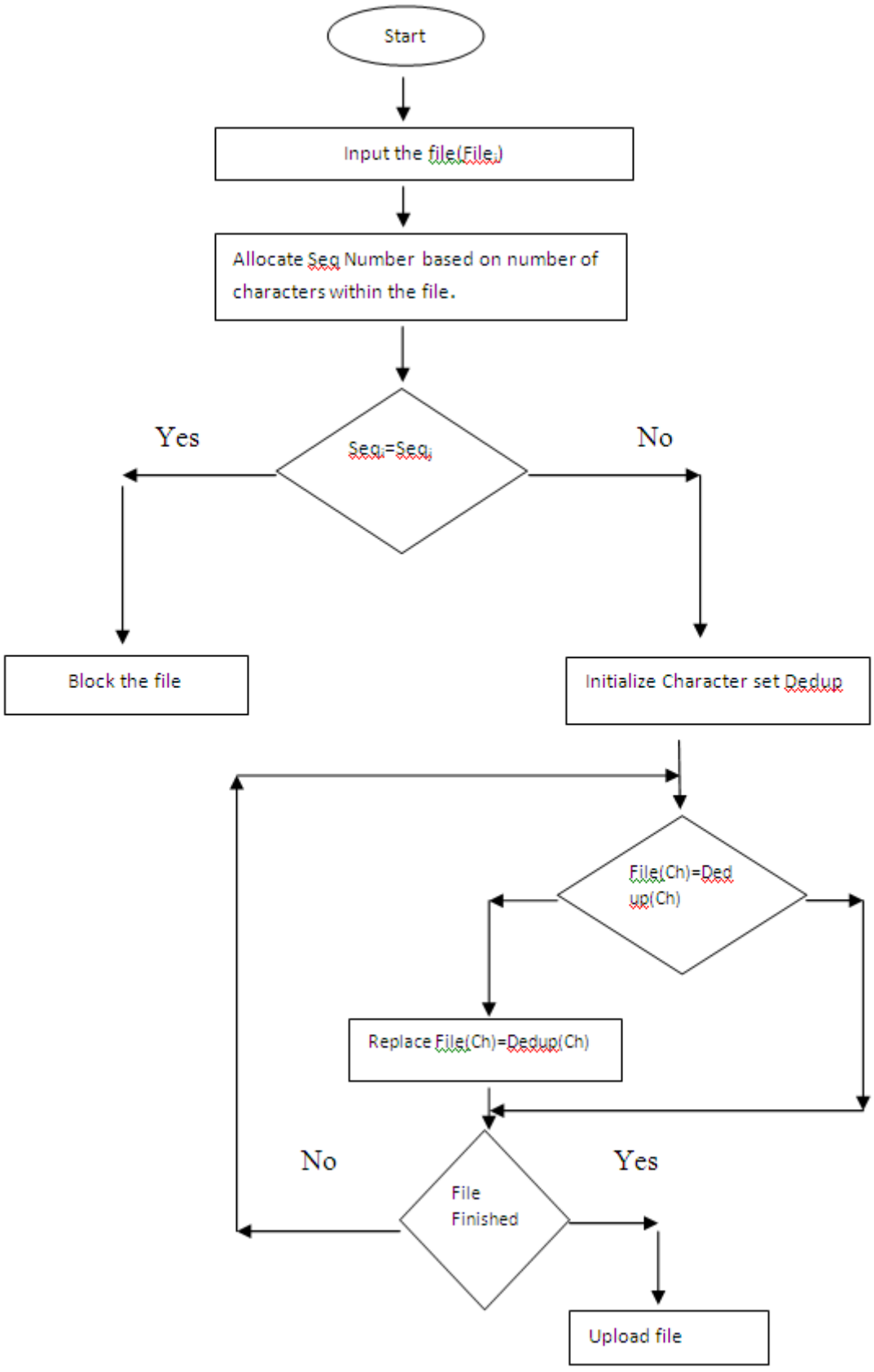
After the encryption process we will upload the data over the cloud
Deduplication instrument is utilized as a part of request to stop a similar document to be transferred again onto the cloud. The procedure will distinguish the copied record with the assistance of arrangement number allotted to the document which will be founded on substance of the records. On the off chance that next record shows up for transferring at that point grouping number will be thought about against the new document succession number. The deduplication which is performed is with the assistance of character set which is kept up on the machine. The characters will be contrasted against the record with be transferred. On the off chance that the match happens then character in the primary document be supplanted with the deduplicated character set. In the event that match happens, document as of now exist and record is disposed of. The

The flowchart for the deduplication process as follows

pseudo code for deduplication process is described as follows

Dedup($File_i$)

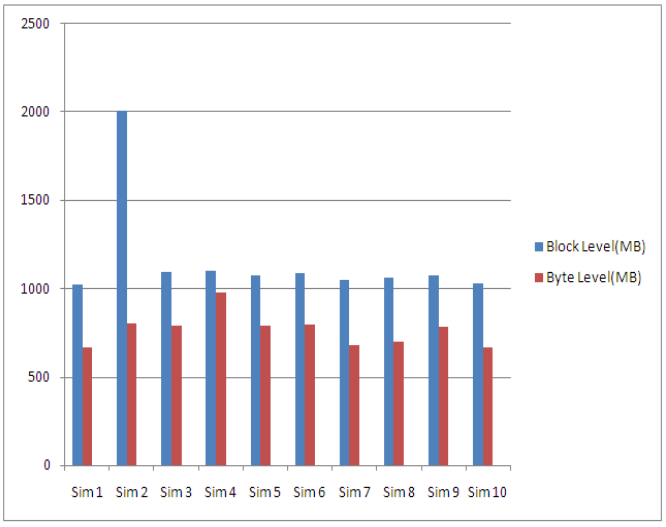
1) Load $File_i$
2) Check Contents of $File_i$ and allocate Seq_i depending upon the contents of the file// the contents of the file will be given sequence number which will be allocated depending upon number of uppercase and number of lowercase and special characters within the file.
   2.1 Repeat while j<n// where n is the total seq numbers allocated
   If(Seq_i==Seq_j)// the sequence number of already present file if matches with new file then
   Block the file
   Else
   Goto step 3
   End of if
   J=j+1
   End of loop
3) Compare the $File_i$ with the character set used for conversion// if file is not already uploaded then
   3.1 if $File_i$(ch)==Dedup(ch) then// original file will be replaced with deduplicated file
   Replace $File_i$(ch) with Dedup(ch)
   Move to the next Character
   End of if
   3.2 Repeat step 3.1 untill entire file is converted
4) Store the Dedup file on the cloud



5. RESULTS AND DISCUSSION

The result indicates that the proposed system show better result in terms of size which is occupied over the cloud. The existing system utilizes more files during the encryption and uloading over the cloud. The proposed system utilizes less space as compared to existing system. The time consumption is also minimized by the use of proposed methodology. The result in terms of bit level deduplication compared with the byte level deduplication is as shown through tabular structure.

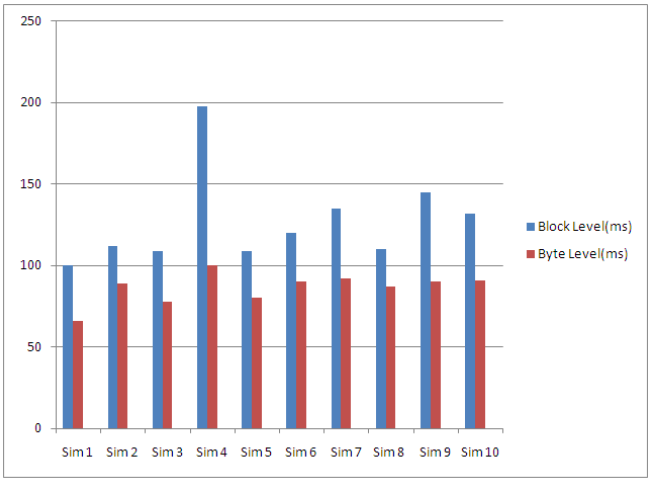
Simulation	Block Level(MB)	Byte Level(MB)
Sim 1	1024	667
Sim 2	2006	800
Sim 3	1097	789
Sim 4	1098	980
Sim 5	1078	789
Sim 6	1088	799
Sim 7	1048	680
Sim 8	1064	698
Sim 9	1072	784
Sim 10	1032	670



The results in terms of time when different data is presented is as follows

Simulation	Block Level(ms)	Byte Level(ms)
Sim 1	100	66
Sim 2	112	89
Sim 3	109	78
Sim 4	198	100
Sim 5	109	80
Sim 6	120	90
Sim 7	135	92
Sim 8	110	87
Sim 9	145	90
Sim 10	132	91

The result of time consumed is listed through the following chart



6. CONCLUSION AND FUTURE WORK

From the result it is clear that the proposed system is producing result which is optimal as compared to the existing system. The separate index file is not required to be transferred in this case. The size hence is reduced when file is uploaded on the cloud. Due to size constraint speed is also enhanced. The time consumption is minimized when byte level de duplication is followed.

In the future strong encryption along with size reduction policy can be worked out to be used along with de duplication to enhance the process further.

REFERENCES

[1] T. Jiang, X. Chen, Q. Wu, J. Ma, W. Susilo, and W. Lou, "Secure and Efficient Cloud Data Deduplication with Randomized Tag," IEEE Trans. Inf. Forensics Secur., vol. XX, no. XX, pp. 1–1, 2016.

[2] W. Leesakul, P. Townend, and J. Xu, "Dynamic data deduplication in cloud storage," Proc. - IEEE 8th Int. Symp. Serv. Oriented Syst. Eng. SOSE 2014, pp. 320–325, 2014.

[3] S. S. Patange and P. G. Scholar, "A Survey : Deduplication Ontologies," vol. 109, no. 1, pp. 30–33, 2015.

[4] C. Science and M. Studies, "New Challenges for Security against Deduplication in Cloud Computing," vol. 2, no. 1, pp. 374–378, 2014.

[5] R. Chen, Y. Mu, G. Yang, and F. Guo, "BL-MLE: Block-Level Message-Locked Encryption for Secure Large File Deduplication," IEEE Trans. Inf. Forensics Secur., vol. 10, no. 12, pp. 2643–2652, Dec. 2015.

[6] R. Miguel, "HEDup: Secure Deduplication with Homomorphic Encryption," in 2015 IEEE International Conference on Networking, Architecture and Storage (NAS), 2015, pp. 215–223.

[7] J. Li, Y. K. Li, X. Chen, P. P. C. Lee, and W. Lou, "A Hybrid Cloud Approach for Secure Authorized Deduplication," IEEE Trans. Parallel Distrib. Syst., vol. 26, no. 5, pp. 1206–1216, May 2015.

[8] T. S. Pawar, R. G. Sawant, P. S. Bothe, and Sh. A. Chopade, "A Survey on Login Authentication System using Captcha as Graphical Password Technquies," Int. J. Innov. Res. Comput. Commun. Eng., vol. 3, no. 10, pp. 10131–10138, 2015.

[9] X. Yang, "Bat Algorithm Literature Review and Applications," pp. 1–10, 2013.

[10] T. Veni and S. M. S. Bhanu, "Dynamic Energy Management in Cloud Datacenters : A Survey," Int. J. Cloud Comput., vol. 3, no. 4, pp. 13–26, 2013.

[11] L. Best-Rowden, H. Han, C. Otto, B. F. Klare, and A. K. Jain, "Unconstrained face recognition: Identifying a person of interest from a media collection," IEEE Trans. Inf. Forensics Secur., vol. 9, no. 12, pp. 2144–2157, 2014.

[12] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing," IEEE Trans. Parallel Distrib. Syst., vol. 22, no. 5, pp. 847–859, May 2011.

[13] P. Date, "Encryption in the Cloud," no. April, pp. 1547–1551, 2014.

[14] C. Science and M. Studies, "Securing user data on cloud using Fog computing and Decoy technique," vol. 7782, pp. 104–110, 2014.

[15] G. L. Prakash, M. Prateek, and I. Singh, "Data encryption and decryption algorithms using key rotations for data security in cloud system," Int. Conf. Signal Propag. Comput. Technol. (ICSPCT 2014), vol. 3, no. 4, pp. 624–629, 2014.

[16] K. Yang, X. Jia, and K. Ren, "Secure and Verifiable Policy Update Outsourcing for Big Data Access Control in the Cloud," IEEE Trans. Parallel Distrib. Syst., vol. 26, no. 12, pp. 3461–3470, Dec. 2015.

[17] K. Hwang, X. Bai, Y. Shi, M. Li, W.-G. Chen, and Y. Wu, "Cloud Performance Modeling with Benchmark Evaluation of Elastic Scaling Strategies," IEEE Trans. Parallel Distrib. Syst., vol. 27, no. 1, pp. 130–143, Jan. 2016.

[18] V. Inukollu, S. Arsi, and S. Ravuri, "Security Issues Associated With Big Data in Cloud Computing," Airccse.Org, vol. 6, no. 3, pp. 45–56, 2014.

[19] C. Wang, N. Cao, K. Ren, and W. Lou, "Enabling secure and efficient ranked keyword search over outsourced cloud data," IEEE Trans. Parallel Distrib. Syst., vol. 23, no. 8, pp. 1467–1479, 2012.

[20] S. Zhang, Z. Qian, Z. Luo, J. Wu, and S. Lu, "Burstiness-Aware Resource Reservation for Server Consolidation in Computing Clouds," IEEE Trans. Parallel Distrib. Syst., vol. 27, no. 4, pp. 964–977, Apr. 2016.

[21] W. Zhang, K. T. Lam, and C. L. Wang, "Adaptive Live VM Migration over a WAN: Modeling and Implementation," in 2014 IEEE 7th International Conference on Cloud Computing, 2014, pp. 368–375.

[22] 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, Jun. 2014.

[23] D. Darsena, G. Gelli, A. Manzalini, F. Melito, and F. Verde, "Live migration of virtual machines among edge networks viaWAN links," pp. 1–10.

[24] A. Strunk and W. Dargie, "Does Live Migration of Virtual Machines Cost Energy?," in 2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA), 2013, pp. 514–521.

[25] N. R. Katsipoulakis, K. Tsakalozos, and A. Delis, "Adaptive Live VM Migration in Share-Nothing IaaS-Clouds with LiveFS," in 2013 IEEE 5th International Conference on Cloud Computing Technology and Science, 2013, vol. 2, pp. 293–298.

[26] K. Hashizume, D. G. Rosado, E. Fernández-Medina, and E. B. Fernandez, "An analysis of security issues for cloud computing," J. Internet Serv. Appl., vol. 4, no. 1, p. 5, 2013.

[27] S. Sankar Bhunia et al., "Advances in Artificial Intelligence," in Signal, Image and Video Processing, 2015, vol. 1, no. 4, pp. 1–6.

[28] M. Okuhara, T. Shiozaki, and T. Suzuki, "Security architectures for Cloud computing," Fujitsu Sci. Tech. J., vol. 46, no. 4, pp. 397–402, 2010.

[29] K. Yang and X. Jia, "Expressive, Efficient, and Revocable Data Access Control for Multi-Authority Cloud Storage," IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 7, pp. 1735–1744, Jul. 2014.

[30] L. Ramaswamy et al., "Cache Clouds: Cooperative Caching of Dynamic Documents in Edge Networks," 2014 6th Int. Work. Sci. Gateways, vol. 86, no. Iccse, pp. 335–338, 2012.

[31] K. Yang and X. Jia, "An Efficient and Secure Dynamic Auditing Protocol for Data Storage in Cloud Computing," IEEE Trans. Parallel Distrib. Syst., vol. 24, no. 9, pp. 1717–1726, Sep. 2013.