A Method for Text Data Fragmentation to Provide Security in Cloud Computing

Archana M, Mallikarjun Shastry P M

Abstract: ***Security is one of the most crucial aspects in cloud computing, in order to provide security in cloud computing there are many cryptographic and non-cryptographic techniques [2][3] are used. Most of the non cryptographic approaches suffer from security breaches and the main drawback of cryptographic algorithms is computation time incurred in encryption and decryption of data. A methodology is proposed to implement unique approach to provide a security in cloud computing. Where text file will be fragmented based on random number generation.From the random number generation algorithm lower range and higher range values will be calculated so that the new random number which needs to compute should be within the lower and higher range of random number.Fragments can be obtained based on the new random number. With reference to the random number, block size or fragment size will be calculated. Hence text file will be divided into fragments.***

Keywords: Cryptography, Fragments, Random Number

I. INTRODUCTION

Cloud storage propositions an abstraction of unlimited storage space for clients to host data, in a compensation as-you-go way[2]. For instance a data sharing websites, video sharing websites chose to host terabytes of data. In this manner, rather than self-keeping up data centers, endeavors would now be able to redistribute the storage of a mass measure of digitized content to that outsider distributed storage suppliers to spare the budgetary overhead in management of data. To ensure redistribute data clear approach is to apply cryptographic encryption onto touchy data with a lot of encryption keys, yet keeping up and securing such encryption keys will make one more security problem. One explicit issue is files deletion requests.

Cloud storage providers may not totally expel all duplicates files (e.g., Cloud storage providers may make different record reinforcement duplicates and disperse them over the cloud for dependability and customers don't have a clue about the number or even the presence of these reinforcement duplicates) and inevitably have the data uncovered if the encryption keys are obtained unexpectedly, also by mishaps or by harmful assaults. In this manner, we look to accomplish a significant security objective by dividing the data into skimpy pieces and these parts will be put away on various hubs in the cloud.

Fragmentation of data takes place based on the pseudo random number [4], Random numbers are numbers which occur by satisfying two conditions (1) the numbers distributed uniformly within the given interval or set (2) it is unable to predict the successor number by knowing the previous numbers in other words the random numbers has no relation with among them.

From the last many years there are numerous methods are evolved to generate random numbers. Many mathematicians have given few acceptable random generation algorithms.

Let us describe the meaning of random number with the well known descriptive examples such as rolling a dice, tossing a coin etc. The actual definition of random can be stated as: A set of random number comprises numbersdistributed uniformly over all of the probable values and each one is independent of the numbers which are generatedalready. Random numbers are numbers which occur by satisfying two conditions (1) the numbers distributed uniformly within the given interval or set (2) it is unable to predict the successor number by knowing the previous numbers in other words the random numbers has no relation with among them.

Various sorts of random number generators exist [4] among them are broadly categorized into (i) True and (ii) Pseudo random number generators. In this paper we are using pseudo random number.

In cloud different types of files, with different size will be stored, for dividing such data files there are different techniques are used[8][9]. For instance, the data file fragmentation threshold determined to be created through the file proprietor. The file owner of the file can indicate fragmentation threshold concerning either the number and size or percentage of various fragments [8]. The fragmentation thresholdpercentage, for example, can direct that individual fragment will be of 5% size of the total size of the file.

On the other hand, the proprietor may produce distinct file encompassing information about size and number of the fragment, for example, the size of fragments one and two is 5,000 and 8,749 Bytes respectively. We contend that the proprietor of the file isn't the one appropriate to produce fragmentation threshold since as a proprietor probably might not be having enough technical acquaintance to predict the number of fragmentshence in this paper, we are using unique randomnumber generation algorithm. In the proposed approach care is taken that no fragment contains any useful information.

Revised Manuscript Received on December 30, 2019.
* Correspondence Author

Archana M, Research Scholar, School of Computing and Information Technology ,Reva University Bangalore, archanashivan85@gmail.com

Dr. Mallikarjun Shastry P M, Professor, School of Computing and Information Technology, Reva University Bangalore, mallikarjunashastry@reva.edu.in

II. BACKGROUND OF RESEARCH

Cloud computing has changed the manner in which the organizations move towards IT, enabling them to turn out to be progressively dexterous, present new plans of business, moderate IT costs and give more amenities. The implementation of cloud technologies can be carried out in a wide assortment of designs, under deployment models and various amenities and can exist together with different innovations and programming configuration methods [2].

The cloud computing scene keeps on acknowledging unstable development. The overall public cloud amenities market was anticipated to develop almost 20%t in 2012, to a sum of $109 billion, with 45.6%development for Infrastructure as a Service (IaaS), which is the quickest developing business sector fragment.

However, for security experts, the cloud displays an enormous problem: How would you grasp the advantages of the cloud while keeping up security powers over your associations' benefits? It turns into an issue of equalization to decide if the expanded risks are really worth the readiness and financial advantages [11].

Keeping up command over the data is vital to cloud accomplishment. 10 years back, big business data regularly lived in the association's physical infrastructure, all alone servers within the venture's server center, where one could isolate delicate data in the individual physical servers. Today, with the virtualization and cloud, the data might be under the association's intelligent control, yet physically dwell in infrastructure claimed and oversaw through alternative entity [12].

This move in charge is the main explanation new strategies and approaches are required to guarantee associations can keep up data security. At the point when an outside party possesses, controls, and oversees infrastructure and computational assets, how might you be guaranteed that business or administrative data stays secure and private and that your association is shielded from harming data breaksand feel you can at present totally fulfill the full scope of consistence, reporting and administrative necessities[14].

So to achieve the security in cloud a new approach is introduced where the file or data which needs to be stored on cloud will be fragmented and distributed across the nodes in a cloud. The file will be fragmented based on the random number which is generated using a random number generation algorithm .Hash value to be calculated and file will be transmitted to node. At node, file fragment will be encrypted and keys will be generated. The file fragments will be retrieved and file will be reconstructed and the file will be available to the user.A true random number generator uses entropy sources which already exist in order to generate set of random numbers[4]. Here the entropy means the amount of uncertainty with respect to the outcome, for instance tossing a coin has high amount of entropy because it is very difficult to predict accurate result of the event. It is because the source of entropy which incurred makes the true random number generator unpredictable. We cannot use the true number generators in simulation and modeling, cryptography [4].

Pseudo random number generators use mathematical formulae in order to generate a set of random numbers, these kind of random number are efficient and deterministic that is a big list of good random numbers are generated within a short interval of time and also given sequence of random numbers can be reproduced later date if the initial conditions are known. These pseudo random are also periodic in nature means the numbers in the list may be repeated eventually.

III.. PROPERTIES OF GOOD PSEUDO RANDOM NUMBER GENERATORS (PRNGS)

Great PRNGs pass various hypothetical and factual tests. Specifically, a great generator ought to have a significant stretch, implying that there are numerous numbers in the succession before it rehashes. Different factual tests have been proposed for PRNGs, a large number of these depend on goodness-of-fit of the points to the expecteddistribution in the event that you were examining from a uniform distribution. For instance, the equidistributional test watches that there is a similar number of points in interims of a similar length. This can be tried on the run-test as well as for tuples of numbers produced by means of PRNG [5]. One more test is the run-test which checks groupings of created numbers which are decreasing or increasing or stay over a specific value. Thesequences length and number of must display certain conduct.

IV.METHODOLOGY OF THE ALGORITHM

This algorithm is very much useful in the file fragmentation process, based on the random number generated the file can be divided into fragments equal to the random number.

A basic pseudo random number is generated using linear congruential generator (LCG) but here initial values for the algorithm are changed since the number of fragments for any file are more hence the initial values used in this algorithm are in the range of 100-500 and this value will be used to perform the mod operation in LCG.

A. Increased Pseudo Random number Generation Algorithm.

The algorithm of the random number generator is as follows

Step 1: Choose the initial value of M for LCG within the range min to max.

Step2: Generate random number using LCG

Step3: Add the LCG number and random number (selected between min to max) provides new random number.

Step4: Perform a XOR operation between new_num and a 4 digit secret number store it as a new random number.

Step5: Add above formed number to Increased Random number List goto step 2.

Step6: Select any number from the Increased Random number List.

B. Algorithm for Increased Pseudo Random Number

```
define initial values to A,C,M
define PIN
For (0 to max)
begin for
    $X[i+1] \leftarrow ( A*X_i +C)$  mod M
    Var ← X[i] + Rand_min_max ()
    Random[i] ←var XOR PIN
    R← Rand_min_max (Random[i])
end for
```

Algo1: Increased Pseudo Random number Generation Algorithm.
Fragment Generation Algorithm.

In the fragment Generation Algorithm, the text file will be fragmented. Each fragmentis referred as block in the algorithm. Block size will be calculated based on the random number generated. Input to the algorithm is Lower range and higher range which needs to be calculated since the new random number always within the range of lower and higher range, it is not within the specified range the random number will be discarded.

Once the random number is accepted then the total number of characters in the input file will be calculated on the basis of total file size and random number block size will be determined.

```
Input: Lr- Lower Range, Hr- Higher Range, F-File.

N ← Compute_Random();
If  Lr<=  N>= Hr
begin
    Size ← sizeofFile(F);
    Count ←  char_count(F);
endif
While (count not equal 0)
Begin while
    Block=count/n;
    For( i is lessthan or equal to block)
    Begin for
        B[j] ← readCharAt(i);
        J++;
        Create_Fragment(b[j]);
    endfor
Count ← count - block;
endwhile

output: fragments of the file
```

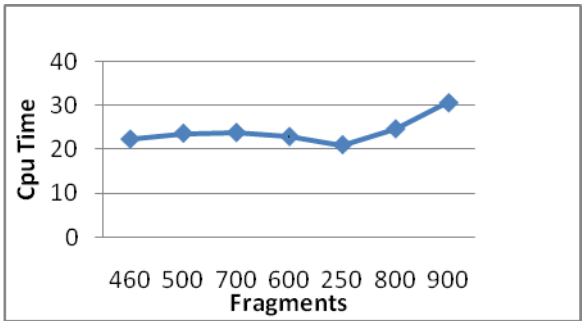
Algo2: Fragment Generation Algorithm

To create a fragment of size equal to block, the same block of characters are read from the input file and new file will be created. The same operation takes till the entire contents of the file is fragmented.

While fragmenting any text file we have taken caution that fragment does not contain any useful data which proves that even if a fragment is vulnerable to any attacker no useful information is revealed. Fragmented file blocks further will be encrypted any placed in the cloud. The above mentioned algorithm explains the proposed methodology of dividing text file into fragments.

IV. RESULTS AND DISCUSSION

Using java we have simulated the proposed approach in localhost, where the owner feeds the source file and then in the execution range of random number is calculated. After the calculation of random number range new pseudo random number is generated using above specified algorithm, intern based on the source file block/fragment size will be calculated with reference to the random number, number of fragments will be generated. We have computed the total time taken for generation of random numbers and fragments. In the proposed algorithm 4 digit secret pin is XORed with obtained random which greatlyenhances the randomness of a number. We compared the randomness of a number generated from our algorithm with the traditional pseudo random number. With reference to the graphs plotted below shows that randomness of a number is greatly enhanced and also our proposed algorithm consumes very less number off CPU resources for both random number generation and data fragmentation.



Performance graph

V. CONCLUSION

In cloud computing, security is a most importantburden. There are various techniques to fragment the data into chunks but in most of the techniques threshold value will be used to fragment the data; some techniques divide the file based on the percentage of the file size. In all these algorithms crucial information such as fragment size is vulnerable. In the proposed algorithm we attempted to generate the fragments basedon the increased pseudo random number. In this methodology each fragment will be of different size. In our proposed methodology information pertained to fragment size is encapsulated. We have calculated CPU cycles consumed to generate pseudo random number using our technique on standard three core processor 2.4GHz. Using our technique textdata stored into cloud file can be fragmented using random number and fragments can be encrypted to further enhance the security in cloud.

REFERENCES

1. Cloud Adoption Practices & Priorities Survey Report January 2015, https:// cloud security alliance.org/research/surveys/
2. W. A. Jansen, “Cloud hooks: Security and privacy issues in cloud computing,” In 44th Hawaii IEEE International Conference OnSystem Sciences (HICSS), 2011, pp.1-10.
3. Kalyani Kadam, Rahul Paikrao, Ambika Pawar, "Survey on Cloud Computing Security"International Journal of Emerging Technology and Advanced Engineering Volume 3, Issue 12, December 2013.
4. Randomness and integrity services Ltd https://www.random.org/randomness/
5. www.users.math.umn.edu/~garrett/ students / reu/ pRNGs.pdf
6. www.cs.princeton.edu/ courses/ archive/ spr03/cs126/assignments/cycle.html
7. https://en.wikipedia.org/wiki/inear_congruential_generator#cite_note-1
8. Mazhar Ali, Student Member, IEEE, Kashif Bilal, Student Member, IEEE, Samee U. DROPS: Division and Replication of Data in Cloud for Optimal Performance and Security,IEEE Transactions on Cloud Computing (Volume: PP, Issue: 99)
9. Dr.M.IndraDevi(Professor) &R.Swathiya(Assistant Professor), Dept of Computer Science and Engineering Kamaraj College of Engineering & Technology Virudhunagar, India.”, Division of data in cloud environment for secure data storage”, 2016 IEEE
10. Bo Li, Peng Liu, Li Lin Guangxi Key Lab of Multi-source Information Mining & Security Guangxi, China. “A Cluster-based Intrusion Detection Framework for Monitoring the Traffic of Cloud Environments”, 2016 IEEE 3rd International Conference on Cyber Security and Cloud Computing.
11. A. Juels and A. Opera, “New approaches to security and availability for cloud data,” Communications of the ACM, Vol. 56, No. 2, 2013, pp. 64-73.
12. G.Kappes, A. Hatzieleftheriou, and S. V. Anastasiadis, “Dike:Virtualization-aware Access Control for Multitenant Filesystems,” University of Ioannina, Greece,Technical Report No.DCS2013-1, 2013.
13. L. M. Kaufman, “Data security in the world of cloud computing,” IEEE Security and Privacy, Vol. 7, No. 4, 2009, pp. 61-64.
14. A. N. Khan, M. L. M. Kiah, S. U. Khan, and S. A. Madani, “Towards Secure Mobile Cloud Computing: A Survey,” Future Generation Computer Systems, Vol. 29, No. 5, 2013, pp. 1278-1299.
15. S. U. Khan, and I. Ahmad, “Comparison and analysis of ten static heuristics-based Internet data replication
16. techniques,” Journal of Parallel and Distributed Computing, Vol. 68, No. 2, 2015,pp. 113-136.