ISSN (ONLINE): 2395-695X
ISSN (PRINT): 2395-695X

International Journal of Advanced Research in Basic Engineering Sciences and Technology (IJARBEST)
Volume 2, Special Issue 19, October 2016

MULTI CLOUD COMPUTING DATA INTEGRITY USING ARTIFICIAL NEURAL NETWORK

N.M.MALLIKA
Assistant Professor
Department of Computer Applications(UG)
Sri Vasavi College(SFW),
Erode, India
nm.mallika@gmail.com

Dr. B.Srinivasan
Associate Professor(SG)
Gobi Arts & Science College
Gobichettipalayam, India

***Abstract-To E*nhance the security level the proposed system introduced an artificial neural network based data integrity for multi cloud computing. Here multilevel firefly Threshold (MFAT) scheme encrypts the file using the secret key before distributing the key shares among participant resource providers whom assume to be honest. Finally Artificial Neural Network (ANN) based classifier is used for checking the Cloud user data integrity.**

Keywords— Multilevel firefly Threshold , Artificial Neural Network ,Data integrity,Cloud user Data integrity.

I. INTRODUCTION

Cloud computing is an emerging technology which provides many services over internet. Many organizations are migrated towards cloud. As cloud is providing multiple characteristics such as on-demand service, location independency, resource pooling and so on. Instead of investing money in new hardware and software and also for the maintaince of resources, users can use servers, storage, applications that are available in cloud [1]. Cloud computing provides you luxury of using all the computer hardware and software from anywhere and at anytime. These softwares and hardwares are not actually installed on your local machine. Few companies provide you services which allow accessing such hardwares/ softwares over internet. Users are unaware about where these resources are actually located and how get managed. The information is transparent to end user [2] [3].

In order to solve the problem of data integrity checking, many schemes are proposed under different systems and security models. Syam et al. [4] proposed a distributed verification protocol using Sobol sequence to ensure availability and integrity of data, but it is also not addressed the data confidentiality issue. How to achieve a secure and efficient design to seamlessly integrate these two important components for data storage service remains an open challenging task in Cloud Computing.

In their subsequent work, Hao et al. [6] proposed a RSA-based privacy-preserving data integrity checking protocol with data dynamics in cloud computing. Their scheme extended the sebe's protocol to support public verifiability. It does not leak any information to third party auditors. However, like [7] it is also not protecting data leakage from the malicious servers

II. RELATED WORK

Al-Saffar et.al [10] presented data integrity approaches for multi cloud environment. Data integrity plays a vital role in cloud computing. Many schemes came into existence in order to secure cloud data. The schemes include public auditing, provable data possession and a host of other techniques. In this research, the proposed a model based on provable data possession in multi-cloud storage. The proposed framework has a combiner that takes request from client and distributed block-tag pairs to various cloud servers. When the combiner gets retrieval request, it gets a challenge and that is distributed among the servers and the server responses are aggregated prior to sending response back to client. The Private Key Generator used in the framework can produce private key based on the identity given. The client and cloud servers do their respective job while the proposed model is capable of ensuring data integrity in distributed environment.

Bachhav et.al [11] suggested multi cloud data sharing by using cryptosystem in cloud computing. In this research, three authentication techniques are discussed such as Attribute based encryption (ABE), Identity Based Encryption (IBE) and Key Aggregate Cryptosystem (KAC). The major concern in ABE is collusion resistance but not compression of secret keys. In IBE, random set of individualities are not match with the design of key aggregation. Key Aggregate Cryptosystem defends user's data privacy by compressing the secret key in public key cryptosystem which supports delegation of secret key for dissimilar cipher text classes. To avoid confusion with the delegated key, the KAC is used for data sharing in cloud storage.

Sravan Kumar and Ashutosh Saxena [12] describes that data integrity of a file stored in the cloud can be identified using a simple method. In their work, a file is divided into different blocks and some arbitrary bits are chosen from each block to find Meta data. This Meta data is then encrypted using some secret function only known to the verifier. The Meta data is appended to the file before uploading it to the cloud. At the time of verification the CSP is asked to send some specified bits from the Meta data, hence it provides a proof to the verifier whether the file is integrated or not. The concept is easy to implement, it also does not encrypt entire file which has less overhead on the thin clients. The major problem with this scheme being its capability to only process and check integrity of static data stored in the cloud.

Shacham et.al [13] proposed new technique to obtain PoR. Two schemes are implemented here. First scheme implements PoR with pubic verifiability. Shortest query response of any proof of retrievability is obtained which is secure in the random oracle model. Second scheme has shortest response of any PoR scheme with private retrivability and secure in the standard model. Two homomorphic authenticators are uses that are based on PRF's and second based on BLS signature. Both schemes allow only one authentication value. In this technique, user breaks an erasure encoded file into n blocks. Each file block is accompanied by authenticators of equal length. Use of BLS signature instead of RSA reduces proof size. It also tolerate higher error rate. , This scheme still works on static data only, without support of dynamic data update.

Bowers et.al [14] proposed a PDP model and supports provable updates on stored data. New version of authenticated dictionaries is used which is based on rank information. Rank information is used to organize dictionary entries. Authentication skip list is used to check the integrity of file blocks. It allows insertion and deletion of blocks within the data structure. File F and its skip list are stored on untrusted server. To prevent replay attacks root metadata is stored at client side. File f is divided into blocks. When client wants to verify integrity of block I he issues query atRank(i) to the server. Server then computes T(i) as its proof and send to client. Clients check integrity by comparing proof of server with stored metadata. Also to update the data client issue atRank(i)(for insertion)and atRank(i-1)(for deletion).Tags used here are more efficient than PDP for static data. It's computational and communication complexity can be up to O (logt). A limitation of DPDP is that it does not allow for public variability of the stored data. In addition it does not consider data freshness or fairness.

III. Proposed work

3. PROPOSED DATA INTEGRITY AND SECURITY BASED SCHEMAS

In the distributed cloud model, users do computation and store data in resources provided by other users. Security of data storage currently depends on how strong encryption keys a user has used or how effective the key management schemes used. One central concern in cloud computing is the privacy and integrity of data processed at the cloud. Data integrity checking basically means protection of data from unauthorized users or hackers and providing high security to prevent data intrusion.

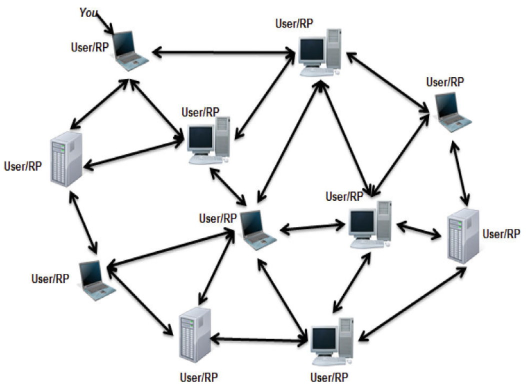

Fig. 1. Distributed cloud model

In this system an Artificial Neural Network (ANN) based classifier is proposed to secure the most important data of cloud user. It uses a classification function which takes a cloud user data and check the integrity of data. The MFAT – SS schema sends a unique random key for cloud user to achieve integrity on data from any modification or deletion. Instead of storing the whole data at server specific portions of data is stored; a deterministic verification approach is used. Discussed the problem of ensuring the availability and integrity of data storage in distributed cloud computing. Based on these results, designed our multi-provider cloud architecture that satisfies many of the requirements by providing increased availability, confidentiality and integrity of the data in the cloud. This architecture features secret sharing as an important measure to distribute data as fragments to different cloud services, which can provide higher redundancy and additional security and privacy protection in the case of key compromise, broken encryption algorithms or their insecure implementation.

3.1 Secret key generation using Firefly algorithm

Firefly algorithm is proposed to generate the secret key values for each user in the distributed cloud. The user splits the secret key into n number of shares S_i,where i ∈(1, n) and distribute them among all resource providers. In this research, multilevel cloud users are considered and threshold value is found for providing security as well as integrity which cloud users belongs to the corresponding threshold value.

Firefly algorithm is one of the efficient optimization algorithms**.** Firefly algorithm is based upon idealizing the flashing characteristic of fireflies. The idealized three rules are:-

1. All fireflies are considered as unisex and irrespective of the sex one firefly is attracted to other fireflies

2. The Attractiveness is proportional to their brightness, which means for any two flashing fireflies, the movement of firefly is from less bright towards the brighter one and if no one is brighter than other it will move randomly. Furthermore they both decrease as their distance increases.

3. The landscape of the objective function directly affects the brightness of the firefly. For a maximization problem, the brightness is proportional to the objective function’s value.

Algorithm 1:

Input: Multilevel Cloud Users
$MCU = (mc\imath$
Random prime positive integer
Output: Secret Key $S\imath$
Objective function f(x), x = $(x_1, \ldots\ldots x_4)^T$

```
1.  Generate initial population (multi cloud users) of fireflies $x_i$ (i=1,2,...n)
2.  Light intensity Ii at xi is determined by $f(x_i)$
3.  Define light absorption coefficient $\gamma$
4.  While(t>MaxGeneration)
5.  for i=1:n all n fireflies
6.  for j=1:i all n fireflies
7.  if $(I_j>I_i)$, Move firefly i towards j in d-dimension;
8.  endif
9.  Attractiveness varies with distance r via $\exp[-\gamma r]$
10. Evaluate new solution and update light intensity
11. end for j
12. end for i
13. Rank the fireflies and find the current best
14. end while

15. Post process results and visualization
```

The above firefly optimization algorithm describes that secret keys are generated for multi cloud users using threshold value. Multi-cloud user security is addressed and provides possible solutions. It is found that the research into the use of multi-cloud providers to maintain security along with secret keys.

Each share of secret S_i is replicated into k numbers so that if one resource provider goes offline or compromised then that share can be accessed from other resource provider. At each resource provider further split the share S_i into m number of shares S_{ij}. In this stage store the data file of each user in distributed cloud. Here if any data loss occurs it have been selected from another source using the data replication schema. Multi-Stage Stochastic Integer Programming (MSSIP) is introduced to solve QoS-aware data replication problem for data-intensive applications in cloud computing systems.

The multi cloud users need to access data from the cloud storage and hence they required multiple shares in the distribution environment. Intend to have multiple share pools and place one or two of them in each cluster of the distributed cloud. The CRT solution generates a number m which decides the number of shares to split and reconstruct in the second level. At each resource provider at least j number of dummy shares SD_{ij} ,where $i \in (1, n)$ and $j \geq m$, are generated.

Whenever a resource provider RP is compromised, the user revokes the access of that particular resource provider. Intend to have a greater number of dummy shares than secret shares, i.e.,$j \geq m$, so that if any outside attacker tries to get the share, the probability that he ends up with the dummy share instead of a real share is greater than or equal to 0.5. This helps the user to take action (e.g., revoke the access of that resource provider) accordingly when some attacker selects a dummy key. To reconstruct the key sub shares, each resource provider need to have the P_i from the user to generate the threshold value m. The user reconstructs the secret S from t numbers of S_i shares.

Algorithm 1: Multilevel firefly threshold schemes

Input: Number of multi cloud users $MCU = (mcu$
Random prime positive integer , shares
available resources providers R, the key
generated from firefly threshold as SK,
the secret key values is splitted into key shares
$sk_1 =$. The data stored of the each user file
is denoted as .
Output : Key sharing

```
1. for $CMU = (mcu_1,$do
2. Generate prime positive integer from the firefly and generate key values
3. User MCU encrypts the file D with secret key
4. Split the secret key SK into n number of shares, $Sh_1$, $Sh_2$, $Sh_3$, $Sh_4$, .., $Sh_n$.
5. To reconstruct SK at least t number of shares is required.
```

```
6.  for each share i of S, where $i \in 1, n$ do
7.  Replicate the share $Sh_i$ into $k \geq 1$ number of replicas.
8.  Distribute the replicas among n number of resource providers.
9.    If any data loss occurs call data replication MSSIP
10. end for
11. for each resource provider, $RP_i$, $i \in 1, n$ do
12. Select a pair $(P_i,N_i)$ by the following steps
        • for i = 1 atleast n do
        • Generate a random series of pairwise relatively prime positive integers, $P_i = p_{i1}, p_{i2}, ., p_{im}$
        • Generate a random series of m arbitrary integers $Ni = n_{i1}, n_{i2},..., n_{im}$.
        • Place these two series Pi along with Ni, represented as $(P_i,N_i)$
        • end for
13. Cloud User CU saves $(i, P_i)$ and $RP_i$ saves $N_i$
14. Get a unique solution $m = x_i$ from $(P_i,N_i)$
15. Split the share of secret $Sk_i$ into m number of shares, $S_{i1}, S_{i2}, S_{i3},S_{i4}, .., S_{im}$
16. Generate j number of dummy shares $S^D_{ij,}$ where $j \geq m$
17. Reconstruct the share of secret $Sh_i$ from m number of shares,
18. end for
19. for each resource provider , $Rp_i$, $i \in 1, t$ do
20. Collect the share $Sh_i$ from each resource provider
21. end for
22. Reconstruct the secret key SK from $Sh_i$ where $i = 1, ..., t$.
23. end for
```

3.2 Artificial neural network

Information security is the main concernment regarding with IDS, in which the objective is for protecting the confidentiality, integrity and availability of data in the system. In order to provide higher security for the confidential data with information integrity, the proposed system introduced an ANN classifier. The data integrity is compared on the basis of the classification accuracy and the minimum error obtained using the ANN. The ANN requires data to be presented in a certain format supported by the learning process of the network.

The common type of artificial neural network consists of three groups, or layers, of units: a layer of "input" units is connected to a layer of "hidden" units, which is connected to a layer of "output" units. The activity of the input units represents the raw information that is fed into the network. The activity of each hidden unit is determined by the activities of the input units and the weights on the connections between the input and the hidden units. The behavior of the output units depends on the activity of the hidden units and the weights between the hidden and output units.

The number of nodes in the input and the output was determined by the structure of user data. The number of neurons in the input layer and the output layer depends upon the number of input and the output variables

The samples (user data) were divided automatically into three subsets in ratio 2:1:1 i.e. training set (50%), selection set (25%) and testing set (25%). The training set is used to train the network, while the selection set is used to check the progress of the network and to define the epoch at which training should be stopped. Training is stopped when the error in the selection set reaches a minimum value. At this point the network has achieved the best generalization. If network training is not stopped, the network will over-train and the performance of the network will deteriorate, despite the error of the training set still decreasing. The testing set is used at the end to check the selection set is not artificial. If the ANN trained network shows a good performance, then it is tested using the testing set. If the ANN tested network shows a good result, then the network is identified as suitable for user data classification. If the ANN fails to produce better results during training or testing, then the ANN training process is repeated again.

Psudeocode 1 :

```
Input: Training samples (cloud Data)
Target: label values
Rate: learning rate
1. Initialize weights
2. While not termination condition satisfied do
3. For i      1 to number of samples do
4. For j 1      to number of features do
5. Weight [j]        Weight [j]  +rate * (target –sum [i] - weighted –sum[i]
6. Samples [i] [j]
7. If termination condition is satisfied then return weights
8. End
9.weight
```

4.PERFORMANCE EVALUATION

In the experimentation work performed simulations using a distributed cloud model that is based on the P2P overlay

Kademlia . It is implemented using proposed algorithm for secret sharing on the distributed cloud. It assumed that resource providers have a minimum of 2GB RAM up to 16 GB, 2 to 8 cores. First a node identifies available resource providers near him and then divides the key into multiple shares and distributes each share to an available resource provider. The Resource provider again creates more shares by using his share as the secret and stores it. User will maintain the list of providers who store the secret shares. When a user requires the key, he contacts other resource providers who have the shares. Once resource providers receive the request, they combine the shares they have and send the actual share to the user. finally they check integrity, security and replica.

Time complexity

The system is called better when it has lower time complexity

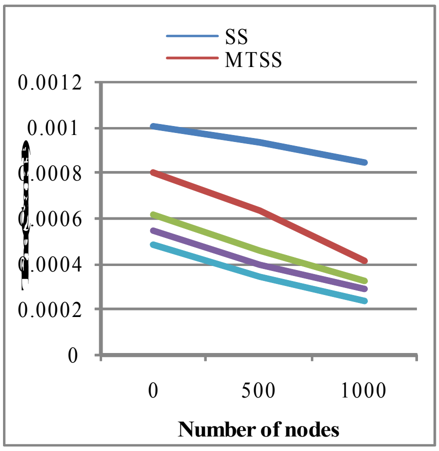

Figure.2. Time to split shares and distribute them to other nodes

Figure 2 shows the average total time of the Secret Sharing (SS), Multilevel Threshold Secret Sharing (MTSS) and the proposed Multilevel Differential Evolution Threshold (MDET) –SS schema, Multilevel Bat Threshold (MBT) and Multilevel Fierfly Algorithm Threshold based secret sharing schema (MFAT-SS) is proposed for security in cloud computing. The total time includes time taken to split the secret into shares at first level, find resources, distribute the shares to resource providers and split the shares at second level. It can see that as the number of nodes increases, the time to find nodes and distribute shares to nearby nodes decreases. Since the distributed cloud is formed by many users for secret sharing is feasible and efficient.

The QoS violation ratio is defined as follows:

=The total number of QoS - violated data block replicas/ The total number of data block replicas (1)

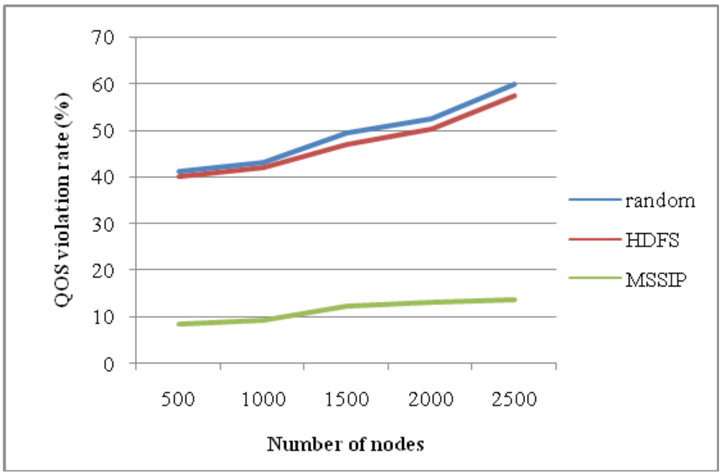

Figure.3. The numbers of QoS-violated data blocks under various device performance

Figure 3 shows the comparison of the QoS violation ratios in the above concerned algorithms. In Figure 3, the QoS violation ratios of these two algorithms are approximately 59 and 57 percent, respectively. The QoS requirement is considered in the proposed replication algorithms. The QoS violated data replicas are generated due to the limited replication space of a node. In addition to minimizing the replication cost, the MSSIP algorithm can also minimize the number of QoS-violated data replicas.

Total replication

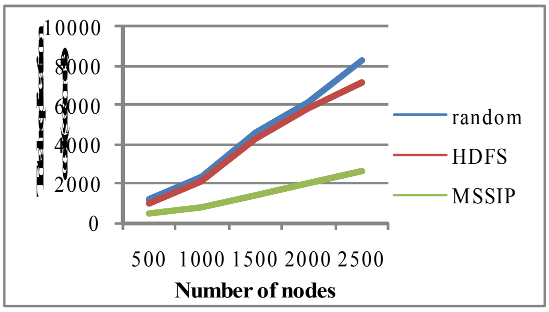

Figure. 4 Total replication cost under various device performance

Figure. 4 shows the total replication costs for different numbers of requested nodes from 500 to 2,500.In Figure. 4 the cloud computing system is configured device heterogeneity using the first three disk access time. The replication factor r_f is set to 2. Basically, the Hadoop replication algorithm adopts the random manner to place the replicas of a data block, but it additionally considers the possible rack failure. Therefore, the total replication cost of the Hadoop replication algorithm is similar to that of the random replication algorithm. The total replication cost of the proposed MSSIP is less when compared to other HDFS and random replication algorithm.

5.CONCLUSION

Cloud Computing is a relatively new concept that presents a good number of benefits for its users; however, it also raises some security problems which may slow down its use. The current work ensures the implementation of Multilevel Fierfly Algorithm Threshold (MFAT) based secret sharing schema is proposed for security in cloud computing. The main idea behind this secret sharing schemes, information is split into multiple shares and these shares are distributed among multiple users to ensure the security of cloud users. Proposed a Artificial Neural Network (ANN) based classifier is proposed in order to secure most important data of cloud user. Cloud user data integrity checking, uses a classification function which takes a cloud user data and verify the integrity. In addition the proposed investigated the QoS-aware data replication problem in distributed cloud computing. To solve the data replication problem the Multi-Stage Stochastic Integer Programming (MSSIP) is introduced. An experimentation result shows that the proposed MFAT based secret sharing schemas provide high security when compared to existing state of art methods.

REFERENCES

[1] Ahuja R. (June 2011) 'SLA Based Scheduler for Cloud storage and Computational Services', International Conference on Computatonal Science and Applications (ICCSA), 258-262.

[2] Albeshri A, Caelli W. (Sept 2010) 'Mutual Protection in a Cloud Computing Environment', 12th IEEE International Conference on High performance Computing and Communications (HPCC), 641-646.

[3] Almulla S, Chon Yeob Yeun. (March 2010) 'Cloud Computing Security management ', 2nd International Conference On Engineering Systems Management and Its Applications , 1-7.

[4] P. Syam Kumar, R. Subramanian, “Homomorpic Distributed Verification Ptorotocol for Ensuring Data Storage in Cloud Computing”. International Journal of Information, VOL. 14, NO.10, OCT-2011, pp.3465-3476.

[5] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson,and D. Song, “Remote Data Checking using Provable Data Possession,” ACM Trans. ACM Transactions on Information and System Security, Vol. 14, No. 1, Article 12, may 2011, pp. 12.1–12.34

[6] Z. Hao, S. Zhong, and N. Yu, “A Privacy-Preserving Remote Data Integrity Checking Protocol with Data Dynamics and Public Verifiability”, Accepted for publication in future issue of IEEE Trans. Knowledge and Data Engineering, DOI: 10.1109/TKDE.2011.62

[7] C. Wang, Q. Wang, K. Ren, and W. Lou, “Privacypreserving public auditing for data storage security in cloud computing,” In Proc. of IEEE INFOCOM’10, , March 2010. San Diego, CA, USA.

[8] J.Bohli, N.Gruschka, M.Jensen, L.Lo Iacono, N.Marnau, “Security and Privacy Enhancing Multi-Cloud Architectures”, IEEE Transaction on Dependable and secure computing . No.99, 2013.

[9] Z. Hao, S. Zhong and N. Yu,"A Privacy-Preserving Remote Data Integrity Checking Protocol with Data Dynamics and Public Verifiability", IEEE Transactions on Knowledge and Data Engineering, Vol. 23, No. 9, September 2011

[10]. Al-Saffar, Ali Mohammed Hameed. "Identity Based Approach for Cloud Data Integrity in Multi-Cloud Environment." *Identity* 4.8 (2015)

11. Bachhav et.al, Secure Multi-Cloud data sharing using Key Aggregate Cryptosystem for scalable data sharing, Suhas Bachhav et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (5) , 2015, 4479-4482

[12]. Sravan Kumar and Ashutosh Saxena “Data Integrity Proofs in cloud Storage” 978-1-4244- 8953-4/11/$26.00 c 2011 IEEE

[13]. H. Shacham and B. Waters, “Compact proofs of retrievability,”in ASIACRYPT ’08: Proceedings of the 14th International Conference on the Theory and Application of Cryptology and Information Security. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 90–107.

[14]. K. D. Bowers, A. Juels, and A. Oprea, “Proofs of retrievability: theory and implementation,” in Proceedings of CCSW 2009. ACM, 2009, pp. 43–54