Multilevel BAT Threshold Based Secret Sharing and Data Replication for Multi-Clouds Computing Security in Distributed Environment

Dr.B. Srinivasan and N.M. Mallika

Abstract--- Cloud computing is beneficial in terms of low cost and accessibility of data. In the preceding research, the issue of data replication is solved on distributed systems. To ensure the security, the existing system used a Multilevel Differential Evolution Threshold (MDET) based secret sharing scheme of security in cloud computing. However the existing system is not useful for multiple cloud users and the problem of resource wastage is not solved optimally. To avoid the above mentioned issues, in the proposed system, Multilevel Bat Threshold (MBT) is proposed to protect the secret keys. For improving the privacy as well as security, MBT algorithm is used to encrypt the file using the secret key prior to share out the keys between resource providers whom assume to be honest. In the proposed research, it split secret keys into multiple shares and stores them in diverse locations using MBT secret sharing algorithm. To handle the data corruption issue, Multi-Stage Stochastic Integer Programming (MSSIP) is adopts the intuitive idea of Quality of Service (QoS) to perform data replication. MBT secret sharing scheme create replicas of secret shares and distribute them among multiple resource providers to ensure availability. Thus the result proves that the proposed MBT based secret sharing scheme is potential to handle threats as well as to identify relate vulnerabilities with possible solutions when compared to existing state of art methods. It provides higher security and privacy for multi cloud users using MBT algorithm significantly.

Dr.B. Srinivasan, Associate Professor, PG & Research Department of Computer Science, Gobi Arts & Science College, Gobichettipalayam, Erode. E-mail:srinivasan_gasc@yahoo. com

N.M. Mallika, Assistant Professor, Department of Computer Applications, Sri Vasavi College, Erode. E-mail:nm.mallika@gmail. com

Keywords--- Cloud Computing, Quality of Services (QoS), Data Replication, Multilevel BAT Algorithm (MBT), Multi-Stage Stochastic Integer Programming (MSSIP***).***

I. INTRODUCTION

The distributed cloud [9] makes use of resources provided by users in a P2P manner. Cloud computing is a computing paradigm which provides to a large number of users and various information technology resources with a high level of scalability using internet technology. In a cloud computing environment users make access to a large scale computing environment through their computing devices connected to the Internet, use necessary information technology resources including applications, storage, operating systems, platforms, etc. as much as they want and at any time that they want, and pay the fee based upon the amount of resources that they have used. The degree of uncertainty is justified in cloud computing by using the cipher cloud. It makes the user to make the data in a secured manner on public frameworks. To accomplish this, the encryption process has been carried out (i.e.) user data will be sent to cloud server, and cloud server will send back the data again to the user. The secured data cannot be accessed in the public cloud computing, so to make the data to be private they are using encryption process [11]. The cloud enhances collaboration, agility, scalability, availability, ability to adapt to fluctuations according to demand, accelerate development work, and provides potential for cost reduction through optimized and efficient computing [8]. Some of the voluntary computing systems such as Berkeley Open Infrastructure for Network Computing

(BOINC) [2] and Planetlab [11] use resources provided by users, but these are managed by a central entity and are completely different from either a centralized cloud or the distributed cloud. Unlike users of the existing cloud, who don't have control over the resources, in the distributed cloud users can choose resources of their choice. Moreover, in the distributed cloud, resources are provided by users. Since users of the distributed cloud have more control over resource selection, they can perform strong encryption to secure their data.

II. RESEARCH METHOD

Parsikalpana et.al [8] presented cloud data security. Many companies are using the cloud architectures. Cloud data security depends more on the procedures and count measures. So this research discusses about the data security issues in cloud. Some of the issues like privacy and confidentiality, data integrity, data allocation and reallocation, data availability, storage and backup recovery etc. to overcome all these type of issues, RSA algorithm has been used. The data will be encrypted and sent to the user, when the user wants the data it will be sent in the decrypted format.

In [16] proposed multi-cloud computing framework using proxy VM instance for sharing resources and dynamic collaboration among cloud based services. This framework manages security, mutual trust a policy issues without need of pre-collaboration agreement, which is necessity in cloud mashups. Whenever cloud user wishes to use any services, he will send request to cloud where CSP has pre-installed proxy VM which will interact with multi-cloud services and provide results to user. It helps for collaboration among various cloud user .They have provided different proxy architecture as cloud hosted proxy, proxy as a service and on premise proxy of which first two architecture are cloud service provider and proxy service provider dependent which cannot resolves malicious system administrator problem. Peer-to-Peer (P2P) proxy architecture will be more secure where client have control over proxies.

Al-Saffar et.al [3] 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 [6] 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.

Kadam et.al [10] discussed security methods for providing strong protection in cloud computing. In many organizations, transformation of information and storage of sensitive data has highest priority. Client data should be kept secret as well as inaccessible from all other unauthorized hacks. To maintain the security of the user data, cloud computing environment has practiced. There are some combined benefits of multi-clouds and secret sharing scheme, such as, infrastructure deployment, data accessibility, user authentication etc. Multi-cloud is looking to be more secure, harder to compromise over single cloud data storage. Cloud computing is currently the latest trend when it comes to online computing, it may help the enterprise and the end user by providing their needs, but the provider has to make sure that they are valuable and customer data is safe. But still this research has issue with sharing confidential detail in large organizations.

Mohammed et.al [12] describes, it is a distributed cryptographic system (High Availability and Integrity Layer) that allows a set of servers to prove to a client that a stored file is intact and retrievable. HAIL relies on a single trusted verifier. It aggregates cryptographic protocols for proof of recoveries with erasure codes to provide a software layer to protect the integrity and availability of the stored data, even if the individual clouds are compromised by a malicious and mobile adversary. HAIL has at least three limitations: it only deals with static data, it requires that the servers run some code and does not provide guarantee of confidentiality of the stored data.

Bessani et.al [4] describes, the increasing popularity of cloud storage services has lead companies that handle critical data to think about using these services for their storage needs. However, the reliability and security of data stored in the cloud still remain major concerns. In this research, DepSky model is studied, a system that improves the availability, integrity, and confidentiality of information stored in the cloud through the encryption, encoding, and replication of the data on diverse clouds that form a cloud-of-clouds. Moreover, the monetary costs of using DepSky in this scenario is at most twice the cost of using a single cloud, which is optimal and seems to be a reasonable cost, given the benefits.

According to Tabaki et al. [18], the way the responsibility for privacy and security in a cloud computing environment is shared between cloud users and cloud service providers differs between delivery models. In SaaS, cloud service providers are more responsible for the security and privacy of application services than the cloud users. This responsibility is more relevant to the public than the private cloud environment because the clients need stricter security requirements in the public cloud. With PaaS, users are responsible for taking care of the applications that they build and run on the platform, while cloud service providers are responsible for protecting one user's applications from others. In IaaS, users are responsible for protecting operating systems and applications, whereas cloud service providers must provide protection for the users' data.

Alihodzic et.al [5] discussed swarm intelligence algorithm for optimization problem. BAT algorithm is one of the latest optimization metaheuristic and it provides improved results with differential evolution. In order to enhance the performance of this hybridized algorithm, a modified bat-inspired differential evolution algorithm is proposed. The modifications include operators for mutation and crossover and modified elitism during selection of the best solution. It also involves the introduction of a new loudness and pulse rate functions in order to establish better balance between exploration and exploitation. Experimental results show that in almost all cases, proposed method outperforms the hybrid BAT algorithm.

III. PROPOSED METHODOLOGY

A. *Proposed Multilevel BAT Threshold (MBT) Based Secret Sharing Schema with Multi User and Multi Cloud Computing Environment*

In the distributed cloud model, users do computation and store data in resources provided by other users as shown in Fig. 1. Since users use resources provided by other users, there are obvious security concerns. Dependable Storage in the Inter-cloud, provides reasons why single cloud is not secure, namely, need to have more than one cloud to make the system more secure. The Distributed cloud model by itself solves the problem of single domain cloud as it has more than one user who will be acting as a cloud provider. In distributed cloud, protecting data stored on other users' resources poses security issues, which is handled using standard encryption techniques, which in turn leads to key management issues and insider attacks. A resource provider can get access to the data stored on his resources with ease and can use brute force attacks on it. Security of data storage currently depends on how strong encryption keys a user has used or how effective the key management schemes used. So instead of having a single key that gives access to the encrypted files, propose using multiple shares.

The preceding research used the method named as Multilevel Differential Evolution Threshold (MDET) for secret sharing mechanism in the distributed cloud which uses secure mechanisms to enhance the security of secret key shares. In this model for a server with limited replication space, if there are many data object replicas to be placed in this server, the replicas of some data objects cannot be stored successfully. In such a case, the unsuccessful data object replicas will be put in other servers without QoS satisfaction. However it is not able to perform the security for multiple cloud users. It shares secret key for individual cloud providers. If there are n number of share keys for individual users then it requires n number of cloud providers so wastage of the resources to solve this problem n number of cloud providers is commonly used for multiple cloud users.

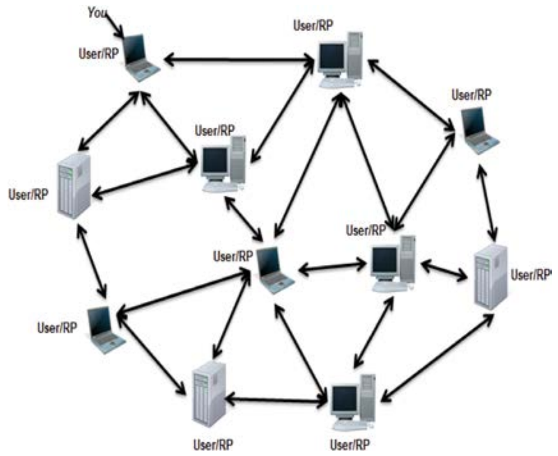

Figure 1: Distributed Cloud Model

To solve the above mentioned issue, secret sharing schemes to protect the secret key in the distributed cloud environment. The proposed method named as Multilevel Bat Threshold (MBT) scheme encrypts the file using the secret key before distributing the key shares among participant resource providers whom assume to be honest. The only way to retrieve the information back is to have all the shares or a qualified number of shares available. To solve data corruption or data leakage problem Multi-Stage Stochastic Integer Programming (MSSIP) is introduced to solve QoS-aware data replication problem for data-intensive applications in cloud computing systems. The data replication problem concerns how to efficiently consider the QoS requirements of applications in the data replication. The main goal of the data replication problem is to minimize the data replication cost and the number of QoS violated data replicas. By minimizing the data replication cost, the data replication can be completed quickly.

The distributed cloud makes use of virtualization to allocate resources to multiple users and shares available resources efficiently and it also avoids single point of failure. In this research, the algorithm is proposed MBT which is focused to protect the secret keys. The overall proposed architecture diagram is illustrated in the fig.2. In this work, at the first level the multiple cloud users split the key and distribute the shares among n resource providers. Instead of attaching key shares as metadata to the pieces of data split the key shares at each resource provider again into multiple shares in the second level [13]. The second level of this mechanism improves the security since to get the original secret the attacker has to have all the shares from the two levels. Generate the threshold value in the second level dynamically which enhances the security as the attacker cannot know about the threshold value beforehand. Here the each virtual bat flies dynamically generate random number and starts with a velocity $v_i$at position (solution) with a varying frequency. As it searches and finds its random number, it changes frequency, loudness and pulse emission rate r.

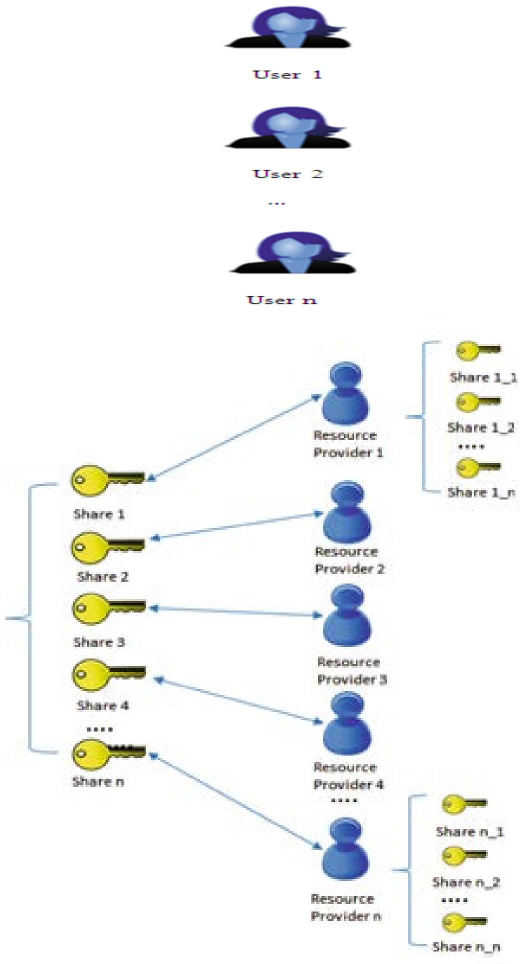

Figure 2: Proposed Multilevel Bat Threshold (MBT) Schemes

B. *Secret Key Generation using BAT Algorithm*

Input: Multilevel Cloud Users

$$MCU = (mcu_1, \ldots mcu_n),$$

Random prime positive integer $PN = (p_1, \ldots p_n)$

Output: Secret Key $SK = (sk_1, \ldots sk_n)$

Begin

Step 1: Compute the objective function $f(x), = (x_1, \ldots . x_d)t$, Set the generation counter t=1,

Initialize the population (multi cloud users) of n bats randomly and each bat corrsponding to a potential solution to the given problem; Define loudness Ai, pulse frequency Qi and the initial velocities vi $(i = 1, 2 \ldots N)$; Set pulse rate r_i

Step 2: Repeat. Generate new solutions by adjusting frequency, and updating velocities and locations/solutions using (1) (2) and (3)

$$f_i = f_{mn} + (f_{max} - f_{min})\beta \quad (1)$$

$$v_i^t = v_i^{t-1} + (x_i^t - x_*)f_i \quad (2)$$

$$x_i^t = x_i^{t-1} + v_i^t \quad (3)$$

if (rand > r_i) then

Select a solution among the best solutions;

Generate a local solution around the selected

Best solution;

End if

Generate a new solution by flying randomly;

$if\ (rand < A_i \&\& f(x_i) < f(x*))$ then

Accept the new solutions

Increase r_i and reduce A_i using below equations

$$A_i^{t+1} = \alpha A_i^t \quad (4)$$

$$r_i^{t+1} = r_i^0(1 - e^{-\gamma}) \quad (5)$$

where α and γ are constants.

End if

Rank the bats and find the current best one $x*$

t=t+1

Step 3: Until the termination criteria is not satisfied or t< MaxGeneration

Step 4: Post-processing the results and visualization

End

BAT algorithm is proposed to generate the secret key values for each user in the distributed cloud. The user splits the secret key sk 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. Bat Algorithm, proposed by Yang, is inspired by echolocation characteristic of bats. Echolocation is typical sonar which bats use to detect prey and to avoid obstacles. These bats emit very loud sound and listen for the echo that bounces back from the surrounding objects. Thus a bat can compute how far they are from an object . Furthermore bats can distinguish the difference between an obstacle and a prey even in complete darkness. In order to transform these behaviors of bats to algorithm, Yang idealized some rules:

1. All bats use echolocation to sense distance, and they also know the difference between food/prey and background barriers in some magical way.

Bats fly randomly with velocity vi at position xi with a frequency f_{min}, varying wavelength and loudness A0 to search for prey. They can automatically adjust the wavelength (or frequency) of their emitted pulses and adjust the rate of pulse emission r ϵ [0, 1], depending on the proximity of their target;

2. Although the loudness can vary in many ways, we assume that the loudness varies from a large (positive) A0 to a minimum constant value A_{min}.

The above BAT 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) [20] is introduced to solve QoS-aware data replication problem for data-intensive applications in cloud computing systems. At this level, i.e., second level the threshold value m is generated dynamically. To determine the number of shares m each resource provider Rp_i, i ∈ (1, n) selects a (P_i,N_i) pair from the share pool SP. The share pool is created beforehand. The user saves the pair (i, P_i) for each provider RP_i and the provider saves N_i.

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 ∈ (1, n) and j ≥ 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 ≥ 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.

Given a distributed cloud computing system with a set of storage nodes SN, these storage nodes can also run applications in addition to storing data. The storage node functionality is similar to the storage node in HDFS [21]. For a storage node $sn \in SN$, if its running application writes a data block db to the disk of r, a replication request will be issued from r to replicate a number copies of b to the disks of other nodes. In the cloud computing system, |SN| is usually large. It is very possible that there may have many concurrent replication requests issued from different nodes at a certain time instant. Due to space limitation, each node cannot store too many data replicas from other nodes. For a data block db, if it is replicated from node s to node q, one data replica dr of b will be stored at q. A desired access time T is specified for dr. In addition, dr is also associated with a replication cost RC and an access time AC. When the original copy of b cannot be read due to data corruption, r attempts to retrieve the data replica dbr from q. If AC is greater than T, dbr is one QoS-violated data replica. Table 1 lists the notations used in data replication algorithm.

Algorithm 2: Multilevel BAT Threshold Schemes

Input: Number of multi cloud users $MCU = (mcu_1, \dots mcu_n)$, Random prime positive integer $PN = (p_1, \dots p_n)$, shares available resources providers $RP = (rp_1, .. rp_n)$, the key generated from BAT threshold as $SK = (sk_1, \dots sk_n)$, the secret key values is splitted into key shares $sk_1 = (sh_{1_1}, .. sh_{1_n})$. The data stored of the each user file is denoted as $D = (d_1, \dots .. d_n)$.

Output : Key sharing

```
for $CMU = (mcu_1, \dots mcu_n)$do
    Generate prime positive integer PN from the BAT and generate key values SK
    User MCU encrypts the file D with secret key SK
    Split the secret key SK into n number of shares, $Sh_1, Sh_2, Sh_3, Sh_4, .., Sh_n$.
    To reconstruct SK at least t number of shares is required.
    for each share i of S, where $i \in 1, n$ do
        Replicate the share $Sh_i$ into $k \geq 1$ number of replicas.
        Distribute the replicas among n number of resource providers.
        If any data loss occurs call data replication MSSIP
    end for
    for each resource provider, $RP_i$, $i \in 1, n$ do
        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 $Ni = n_{i1}, n_{i2}, \dots, n_{im}$.
            • Place these two series Pi along with Ni, represented as $(P_i,N_i)$
            • end for
        Cloud User CU saves $(i, P_i)$ and $RP_i$ saves $N_i$
        Get a unique solution $m = x_i$ from $(P_i,N_i)$
        Split the share of secret $Sk_i$ into m number of shares, $S_{i1}, S_{i2}, S_{i3}, S_{i4}, .., S_{im}$
        Generate j number of dummy shares $S^D_{ij,}$ where $j \geq m$
        Reconstruct the share of secret $Sh_i$ from m number of shares,
    end for
    for each resource provider , $Rp_i$, $i \in 1, t$ do
        Collect the share $Sh_i$ from each resource provider
    end for
    Reconstruct the secret key SK from $Sh_i$ where i = 1, ..., t.
end for
```

Next we present replication algorithm for solving the data replication problem in the cloud computing system. Before elaborating the proposed algorithm, first give some definitions for clarifying the data replication problem. Given a distributed cloud computing system with a set of storage nodes SN, these storage nodes can also run applications in addition to storing data. The storage node functionality is similar to the storage node in HDFS [21]. For a storage node $sn \in SN$, if its running application writes a data block db to the disk of r, a replication request will be issued from r to replicate a number copies of b to the disks of other nodes. In the cloud computing system, |SN| is usually large. It is very possible that there may have many concurrent replication requests issued from different nodes at a certain time instant. Due to space limitation, each node cannot store too many data replicas from other nodes. For a data block db, if it is replicated from node s to node q, one data replica dr of b will be stored at q. A desired access time T is specified for dr. In addition, dr is also associated with a replication cost RC and an access time AC. When the original copy of b cannot be read due to data corruption, r attempts to retrieve the data replica dbr from q. If AC is greater than T, dbr is one QoS-violated data replica. Table 1 lists the notations used in data replication algorithm.

Table 1: Summary of Notations

Notation	*Description*
SN	A set of nodes in the distributed cloud computing
$\|SN\|$	The number of nodes in the distributed cloud computing system
SN_r	A Set of requests which is issued replication requests concurrently
$\|SN_r\|$	Total number of requested nodes in SN_r
r_i	A requested node in SN_r
$SN_n^{r_i}$	A set of nodes that store a data block replica sent from r_i
$SN_q^{r_i}$	A set of qualified nodes that store a data block replica sent from r_i
$SN_{uq}^{r_i}$	A set of un qualified nodes corresponding to the requested node r_i
$R(r_i)$	A function that determines that associated rack number of the node r_i
$S_n^{\overline{R(r_i)}}$	A set of nodes which rack numbers are different with that of r_i
q_j	A node in $S_n^{\overline{R(r_i)}}$
r_f	A given replication factor
$a(q_j)$	The available replication space for qualified node q_j in terms of the amount of block space.
$T_{Access}\left(r_i, q_j\right)$	The replica access time from q_j to r_i
$T_{storage}\left(r_i, q_j\right)$	The storage time to store one data replica from the r_i to q_j
$x(r_i, q_j)$	The {0,1} variable indicates whether a data replica is placed at node q_j from node r_i
$y(r_i, q_j)$	The {0,1} variable indicates whether a Qos- violated data replica is placed at node q_j from node r_i

When an application would like to write a data block, the node executing the application would issue a replication request for the data block. The information about the QoS requirement of the application (the desired access time of the data block) is also attached on the replication request to generate a QoS-aware replication request. Multiple QoS-aware replication requests may be issued concurrently from a number of nodes. These concurrent replication requests will be processed in a sequence based on the ascending order of their associated access time. If the replication request i has a higher QoS requirement than the replication request j, the replication request i is associated with a smaller access time than the replication request j. In such a case, the HQFR algorithm will first process the replication request i to store its corresponding data replicas. When processing a QoS-aware replication request from the requested node r_i, it is required to find the correspondingly qualified nodes that satisfy the QoS requirement of the running application in r_i. It has been known that the access time of a data block is used to represent the QoS requirement of a data-intensive application. Assumed that the QoS requirement of the running application in r_i is $T_{qos}(r_i)$ time units. If the node q_j would like to be one qualified node of r_i, it needs to meet the following two conditions.

The nodes q_j and r_i cannot be located within the same rack. This condition is for considering the possible rack failure.

$$R(r_i) \neq R(q_j) \tag{5}$$

where R is the function to determine in which rack a node is located.

The data replica access time from q_j to r_i(T_{access}(r_i; q_j) needs to meet the $T_{qos}(r_i)$ constraint

$$T_{access\,(r_i,q_j)} = T_{disk}\left(q_j\right) + T_{comm}\left(r_i, q_j\right) \leq T_{qos}\left(r_i\right) \text{ (6)}$$

where $T_{disk}(q_j)$ is the disk access latency for retrieving a data block replica from the disk of qj, and T_{comm}(r_i; q_j) is the network communication latency for transmitting a data block replica from q_j to r_i

According to the above two conditions, all the qualified nodes with respect to the requested node r_i can be found. Then, r_f qualified nodes are selected from all the qualified nodes. These r_f qualified nodes have smaller data replica access time than other qualified nodes. Next, the data block of r_i will be, respectively, made one replica to be stored in each of r_f qualified nodes. These r_f qualified nodes will also update their, respectively, available replication space. The replication cost is represented as the sum of the storage costs of all data block replicas, as follows:

$$\textstyle\sum_{\forall i \in SN_r} \sum_{\forall q_j \in SN_n^{r_i}} T_{storage}\left(r_i, q_j\right) \tag{7}$$

$T_{storage}$(r_i, q_j) is similar to T_{access}(r_i,q_j). In (6), have clearly defined T_{access}(r_i, q_j) to be the sum of the network communication latency and the disk access latency for retrieving a data block replica from node q_j to node r_i.

Therefore, $T_{storage}(r_i, q_j)$ includes the time to transmit a data block replica from r_i to node q_j and the time to write the data block replica to the disk of $q_{j.}$ The optimal solution of the data replication problem can be obtained using Multi-Stage Stochastic Integer Programming (MSSIP). The MSSIP is a well-known technique used to solve the optimal problems with the following characteristics: a linear objective function, a number of linear constraints, and an integer solution set. Given an instance P of the data replication problem, the corresponding MSSIP formulation can be expressed as (8) to (12). The used notations can be found in Table 1. In the given ILP formulation, the data replica placement can be obtained based on the binary variables x and y. If $x(r_i, q_j)$ is 1, the node q_j stores one data block replica of the requested node r_i. If $y(r_i, q_j)$ is also 1, the corresponding data block replica is one QoS-violated replica.

$$\text{Min}\left(\sum_{\forall i \in SN_r} \sum_{\forall q_j \in SN_n^{\overline{R(ri)}}} x(r_i, q_j) \times T_{storage\ (r_i,q_j)} \times \alpha(r_i, q_j) + y(r_i, q_j) \times T_{storage\ (r_i,q_j)} \times k(r_i, q_j)\right) \quad (8)$$

$$\text{Subject to } \forall q_j \in SN , \sum_{\forall i \in SN_r} x(r_i, q_j) \le a(q_j) \quad (9)$$

$$\forall r_j \in SN_r \wedge \ \forall q_j \ \in \overline{S_q^{r_i}}, y(r_i, q_j) = x(r_i, q_j) \quad (10)$$

$$\forall r_j \in SN_r \wedge \ \forall q_j \ \in S_q^{r_i}, y(r_i, q_j) = 0 \quad (11)$$

$$\forall r_j \in SN_r \wedge \ \forall q_j \ \in \overline{S_q^{r_i}}, x(r_i, q_j), y(r_i, q_j) \in \{0,1\} \quad (12)$$

$$\forall q_j \in SN_r, \sum_{\forall q_j \in SN_n \overline{R(r_i)}} x(r_i, q_j) = \mathrm{r_f} \quad (13)$$

where $\alpha(r_i, q_j)$ and $k(r_i, q_j)$ are the constraint variable of unqualified nodes and coefficient k is used to ensure that the number of QoS-violated data replicas will be first minimized parameters, respectively.By adding up all the values of y, the total number of QoS-violated data replicas can be obtained. This number is expected to be as small as possible by associating with a constant coefficient,

$$k = \text{ma x}_{\forall r_i \in SN_r \wedge \forall q_j \in S}\{(T_{storage}\ (r_i, q_j))\} + 1 \quad (14)$$

With the setting of k, each $y(r_i, q_j)$ has a larger coefficient than each $x(r_i, q_j)$. It is also known that the values of $x(r_i, q_j)$and $y(r_i, q_j)$ are either 0 or 1.

IV. Performance Evaluation

In the experimentation work performed simulations using a distributed cloud model that is based on the P2P overlay Kademlia [22]. 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.

Time Complexity

The system is called better when it has lower time complexity

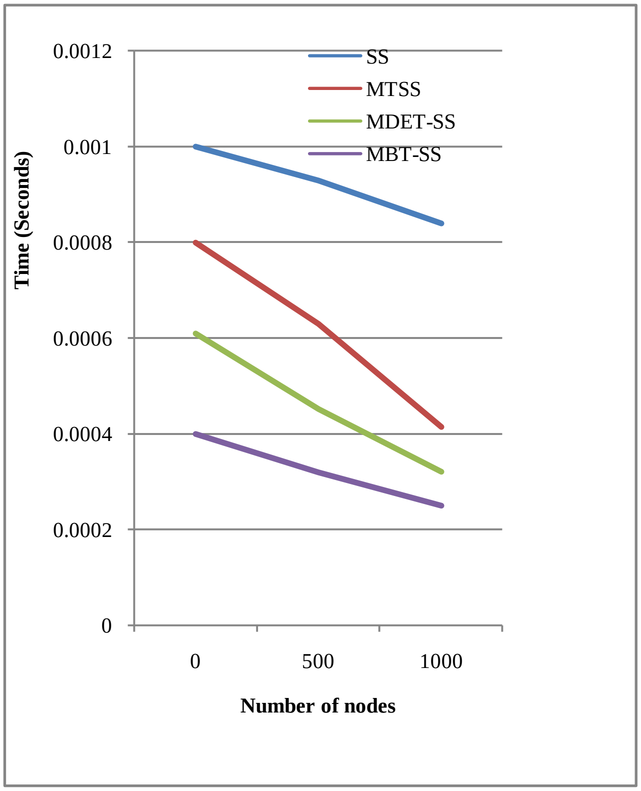

Figure 3: Time to Split Shares and Distribute them to Other Nodes

Figure 3 shows the average total time of the Secret Sharing (SS), Multilevel Threshold Secret Sharing (MTSS), Multilevel Differential Evolution Threshold (MDET) –SS schema and the proposed Multilevel BAT Threshold (MBT) –SS schema. 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 using proposed MBT algorithm.

Total Replication

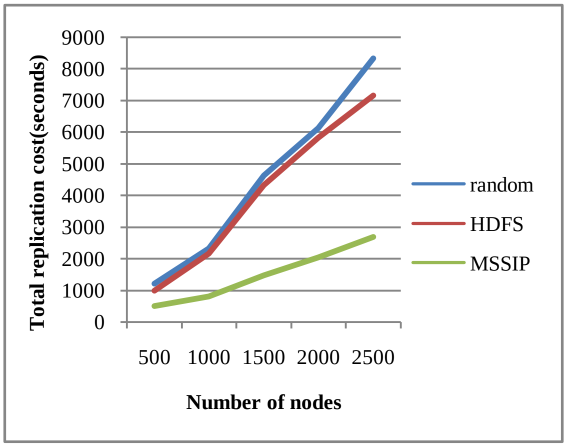

Figure 4: Total replication Cost Under Various Device Performance

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.

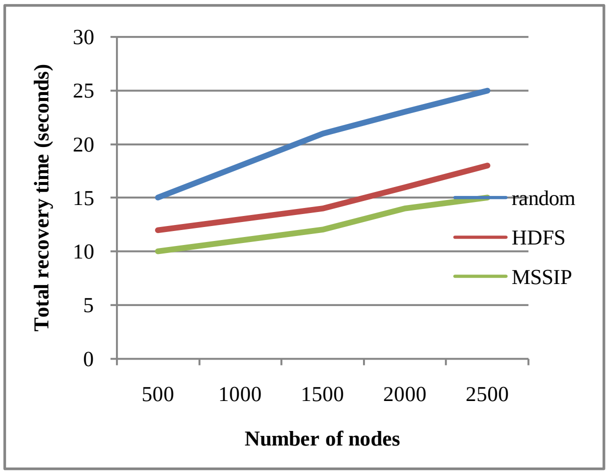

Figure 5: Total Replication Cost Under Various Device Performances

Figure. 5 show the comparison of the average recovery time for a corrupt data block. If the requested node r_i cannot read a data block from its disk due to data corruption, how much time is taken by $r_i$to retrieve one replica of the data bock from another node. The QoS violation ratio is defined as follows:

=The total number of QoS - violated data block replicas/ The total number of data block replicas (15)

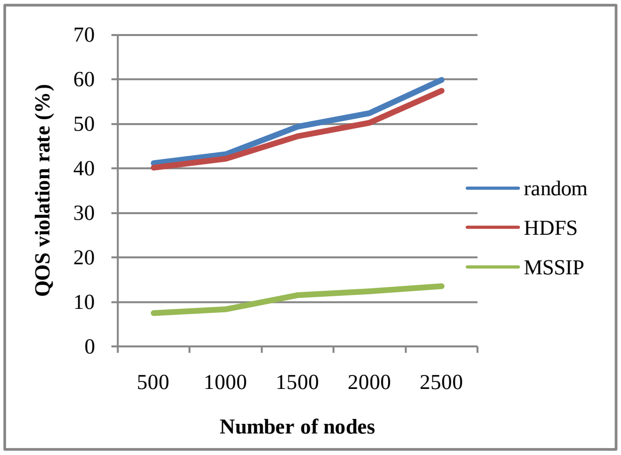

Figure 6: The Numbers of QoS-Violated Data Blocks Under Various Device Performance

Figure 6 shows the comparison of the QoS violation ratios in the above concerned algorithms. In Figure 6, 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.

V. CONCLUSIONS AND FUTURE WORK

In order to provide a secure environment and to protect sensitive static and dynamic data on cloud computing, firstly, different threats, vulnerabilities and risks are explained. The combination of multi-cloud users, providers and secret sharing algorithm is promising, but as of yet it deals with many uncertainties. The existing method of MDET is not dealt with multi cloud users and wastage of resources effectively. Hence the proposed research ensures implementation of multilevel secret sharing scheme for distributed cloud computing. Multilevel BAT Threshold (MBT) based secret sharing scheme of security in cloud computing, giving an overview of the current status of security in this emerging technology. In the proposed research, MBT is focused to split the keys into multiple shares and these shares are distributed among multiple users to ensure the security of secret key on the distributed cloud environment. The proposed MBT is used to produce the privacy as well as security for multi cloud users in the cloud setup. The QoS aware data replication issue is analyzed by using MSSIP approach efficiently. To solve the data replication problem, the device heterogeneity is also assumed in addition to the QoS requirements of applications. The cloud data are protected through MSSIP approach and secret keys are shared based on the threshold level. The proposed BAT generates optimal threshold value for providing privacy as well as security in the multi cloud user and provider setup.

It is used to identify the threats and vulnerabilities using MBT based secret sharing scheme. The experimental result shows that the proposed system provides higher performance rather than the existing system using MBT secret sharing scheme. In future, the integrity for multiple cloud users can be investigated using advanced cryptography algorithms.

REFERENCES

[1] S. Ahmed, A.J. King and Parija, “A multi-stage stochastic integer programming approach for capacity expansion under uncertainty”, Journal of Global Optimization, Vol. 26, No. 5, Pp. 3-24, 2013.

[2] D.P. Anderson, “Boinc: A system for public-resource computing and storage”, International Workshop on Grid Computing, Pp. 4-10, 2004.

[3] Saffar and A. Mohammed Hameed, “Identity Based Approach for Cloud Data Integrity in Multi-Cloud Environment”, Identity, 2015.

[4] A. Bessani, M. Correia, B. Quaresma, F. André and P. Sousa, “DepSky: dependable and secure storage in a cloud-of-clouds”, ACM Transactions on Storage (TOS), Vol. 9, No. 4, Pp. 12, 2013.

[5] Alihodzic and M. Tuba, “Improved hybridized bat algorithm for global numerical optimization”, International Conference on Computer Modelling and Simulation, Pp. 57-62, 2014.

[6] Bachha, “Secure Multi-Cloud data sharing using Key Aggregate Cryptosystem for scalable data sharing”, International Journal of Computer Science and Information Technologies, Vol. 6, No. 5, Pp. 4479-4482, 2015.

[7] Chun, “Planetlab: an overlay testbed for broad-coverage services”, ACM SIGCOMM Computing Communication. Rev., Vol. 33, No. 3, Pp. 3–12, 2003.

[8] Khalid, “Cloud Computing: applying issues in Small Business”, International Conference on Signal Acquisition and Processing, Pp. 278–281, 2010.

[9] P. Khethavath, J. Thomas, E. Chan-Tin and H. Liu, “Introducing a distributed cloud architecture with efficient resource discovery and optimal resource allocation”, IEEE Ninth World Congress on Services, Pp. 386-392, 2013.

[10] M. Kadam, S. Chaudhary and B. Carvalho, “Security Approach for Multi-Cloud Data Storage”, International Journal of Computer Applications, Vol. 126, No. 4, 2015.

[11] M. Kaur and R. Singh, “Implementing encryption algorithms to enhance data security of cloud in cloud computing”, International Journal of Computer Applications, Vol. 70, No. 18, 2013.

[12] M.A. AlZain, E. Pardede, B. Soh and J.A. Thom, “Cloud computing security: from single to multi-clouds”, International Conference on System Science, Pp. 5490-5499, 2012.

[13] T. Mather, S. Kumaraswamy and S. Latif, “Cloud security and privacy: an enterprise perspective on risks and compliance”, O'Reilly Media, Inc., 2009.

[14] P. Kalpana and S. Singaraju, “Data security in cloud computing using RSA algorithm”, IJRCCT, Vol. 1, No. 4, Pp. 143-146, 2012.

[15] K. Shvachko, H. Kuang, S. Radia and R. Chansler, “The hadoop distributed file system”, Mass storage systems and technologies (MSST), Pp. 1-10, 2010.

[16] M. Singhal, S. Chandrasekhar, T. Ge, R. Sandhu, R. Krishnan, G. Joon Ahn and E. Bertino, “Collaboration in Multicloud Computing Environments: Framework and Security Issues”, IEEE computer society journal, Vol. 46, No. 2, Pp. 76-84, 2013.

[17] H. Tabakiet, J.B.D. Joshi and G.J. Ahn, “Security and Privacy Challenges in Cloud Computing Environments”, IEEE Security & Privacy, Vol. 8, No. 6, Pp. 24-31, 2010.

[18] J. Xie, Y. Zhou and H. Chen, “A novel bat algorithm based on differential operator and Lévy flights trajectory”, Computational intelligence and neuroscience, 2013.

[19] S. Yilmaz and E.U. Kucuksille, “Improved bat algorithm (IBA) on continuous optimization problems”, Lecture Notes on Software Engineering, Vol. 1, No. 3, Pp. 279, 2013.