CHAPTER 7

CONCLUSION AND FUTURE SCOPE OF THE RESEARCH WORK

7.1 Summary of the Thesis

Four different solutions have been proposed for the identified problem of security in service ranking, selection, deployment of cloud services. Different security protocols have been designed and implemented for each of the identified problems and each implementation has produced the best results. To conclude each of the solutions is provided for the review purpose.

A MODIFIED TRUSTED CLOUD COMPUTING ARCHITECTURE BASED ON THIRD PARTY AUDITOR (TPA) PRIVATE KEY MECHANISM

The purpose of this scheme proposed is to provide data security in cloud environment: Data coloring and software watermarking techniques is used to protect shared data objects and massively distributed software modules. These techniques safeguard multi-way authentications, enable single sign-on in the cloud, and tighten access control for sensitive data in both public and private clouds. Watermarking is the process of adding the user text behind the image files, and it is used to shade the text into an image file. It is mainly used for digital patent administration and these techniques can be used to protect the data from unauthorized accesses. With the help of watermarking and data coloring process, this implementation can be used in real time process, to hide the data in a safe manner in cloud computing. In this, the system can compute and hide the data in safer side, while retrieving the data, allows only the authorized user to access the data in the cloud server.

In figure 7.1, the security level of our proposed system is analyzed with the comparison of the existing system.
The proposed system gives the best result for the security in the watermarking process. In contrast to traditional solutions, where the IT services are under proper physical, logical and personnel controls. Cloud Computing moves the application software and databases to the large data centers, where the management of the data and services may not be fully responsible. This unique attribute, however, poses many new security challenges which have not been well understood. This thesis, focuses on cloud data storage security, which has always been an important aspect of quality of service. To ensure the accuracy of users’ data in the cloud, an effective and flexible distributed scheme has been proposed with two salient features, opposing to its predecessors. By utilizing the homomorphism token with distributed verification of erasure-coded data, the scheme achieves the integration of storage correctness insurance and data error localization, i.e., the identification of the misbehaving server in the existing method of data coloring and the security level of the proposed method in the watermarking process. In cloud computing, security is used to process the data from the server, to upload a file.
MERKLE HASH TORRENT EXTENSION USING CLOUD COMPUTING MECHANISM

Scope: To provide a solution to the authentication problem exist with Third party auditing and Merkle Hash Function

Data coloring and watermarking techniques is used to transfer the data between the user and the TPA. It provides the secret key which connects to the cloud server for transferring the data between user and cloud server. The Merkle hash function is used to transfer the large size of data in the cloud server. The cloud server provides the private key and public key in order to transfer the data to the multi-cloud server. The security key provided to the users cannot change or modify by the TPA, only the authorized persons can be allowed to change the data in the cloud computing mechanism.

In the transaction data cloud computing, proposed system compares the security level and the existing system. In this, the private key is used to encrypt the data that was transferred from the TPA and it should be stored in the database. Those data to be watermarked and data colored, is used to transfer the data in the cloud server to decrypt the data and then the data will be viewed by the authorized persons. The TPA cannot change the data, only the authorized persons can modify and delete. In the proposed, security key will be provided to transfer the data to another cloud server.

The above figure 7.2, N implies comparable to the security execution level. This shows how much data to be transferred in the cloud server is based on data stored in the database that is transferred to the cloud server. Based on the comparison level, data will be transferred in the cloud.
Figure 7.2. Comparison for Data Security Performance in Merkel Hash Tree

The data is to be transferred in the cloud in view of the comparison level. The security component cloud computing utilizing the hiMerkle hash tree function transfers the data to the TPA and the general public cloud and then a copy of the data will be stored in the private cloud. The data ought to analyze the original final in the event that it is colored or modified. The Merkle hash function is utilized to transfer the large size of image file to the cloud computing mechanism. The existing system utilizes the small size data to transfer. This problem was overcome by utilizing a Merkle hash function as a part of a request to transfer the large size data in the particular cloud. The data is to be transferred in the cloud server by giving security to the clients, unauthorized client’s cannot roll out improvements in the particular cloud and it permits TPA to view the data however not to change the data. User requires to provide the security and then data to be transferred in the cloud server, the other persons cannot change in the particular data. The TPA data only viewing cannot change the particular data.
REAL-TIME SERVICE COMPOSITION AND DEPLOYMENT FOR SECURE COMPUTING IN CLOUD ENVIRONMENT

**Scope: To manage location security and Infrastructure Security**

The proposed strategies have delivered the best results. The keys created utilizing RICHS strategy have great security in nature; the attackers cannot recognize and process copy keys effectively in light of the fact that the randomization of characters utilized is changed intermittently. The runtime composition of services has the effect with different other protocols proposed in this scenario. The proposed strategy groups combine the necessary services at runtime and change it at regular interval. The attackers could not recognize or guess where the service is running in order to generate guesses attack or flooding attack.

The proposed algorithm delivers the best results in comparison with others. Due to the dynamic nature of our algorithm, the attackers could not predict where the service is running. The service composition and deployment is performed at each session, along these lines it builds the security level in the system.

The below Figure 7.3 demonstrates the time taken by our algorithm to deploy a number of services. It demonstrates that the time taken will increase slightly with the number of services. Still, hundreds of services can be deployed in a short time.
Figure 7.3: No of Services Vs Time Taken.

Figure 7.4 demonstrates the average time value required for a set of the service request to be processed.

Figure 7.4: Average time taken to process service request.
AN AGENT BASED SERVICE RANKING ALGORITHM FOR CSP SELECTION IN CLOUD ENVIRONMENT

Scope: Providing security in Service ranking and Selection

The agent-based service ranking algorithm produces efficient results. Service weight is computed to select the particular service provider. The service provider is selected based on the load on each service interface. The overall time to complete the service request is reduced because of the dynamic nature of the agent generation. For each request, an independent agent is generated and each agent is identified with unique agent id so that the overall processing time of service request is reduced hugely.

Figure 7.5 demonstrates the comparisons of overall time taken to process number of requests. It shows that the time taken to process the request is low, even when the number of requests increased, the time taken to process them is less.

![Time Taken to Process Request](image)

**Figure 7.5: The overall time taken.**
7.2 CONCLUSION OF RESEARCH

The solution for the hypothesis had been approached in a strategic approach to have a numerical model and extensive simulation study. The work employed made an effective service selection which guarantees reduced latency with increased completeness and reduced network overhead with increased security. To reduce the network overhead, a lightweight authentication algorithm is developed, which overcomes the overhead generated by different authentication algorithms. To overcome the incompleteness and dissatisfaction of the cloud users, few service selection algorithms are proposed. The selection of service is effectively done with the proposed method, which in turn increases the completion of the service and satisfies the user requirements.

7.3 FUTURE SCOPE OF THE RESEARCH WORK

The work of cloud service selection and authentication in cloud computing can be done extensively in the following ways.

- The time taken for the dynamic selection, composition and deployment of cloud service can be reduced to 10 to 20 %.
- The agent movements for the ranking of cloud services can be reduced by maintaining a minimum log so that the ranking process will be faster to reduce the overall time.
- The metrics used to compute the service quality can be increased by using the service bytes.
## APPENDICES

### APPENDIX I

### DEVELOPMENT SOFTWARE USED IN THIS STUDY

<table>
<thead>
<tr>
<th>Platform</th>
<th>Version</th>
<th>Details</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java</td>
<td>1.6.X</td>
<td>Java Development Environment</td>
<td>Used Java as virtual machine &amp; compiler for implementation.</td>
</tr>
<tr>
<td>Net beans</td>
<td>7.X</td>
<td>Net beans Integrated Development Environment</td>
<td>This makes our development as an easier task by providing integrated development environment.</td>
</tr>
<tr>
<td>Eclipse</td>
<td>4.3.1</td>
<td>An Integrated Platform for Software Development with various plug-ins to support Variety of applications</td>
<td>This makes the software development as easier task which reduces the production time as lesser one.</td>
</tr>
</tbody>
</table>
## DEVELOPMENT PLATFORMS USED IN THIS STUDY

<table>
<thead>
<tr>
<th>Platform</th>
<th>Version</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eucalyptus</td>
<td>4.1</td>
<td>Used to create cloud architecture.</td>
</tr>
<tr>
<td>Open Stack</td>
<td>4.2</td>
<td>Used to develop a cloud architecture which is provided by Apache foundations.</td>
</tr>
<tr>
<td>Cloud Stack</td>
<td>4.2</td>
<td>An open source cloud platform used to develop cloud applications.</td>
</tr>
<tr>
<td>Hadoop</td>
<td>4.3</td>
<td>A cloud platform which supports Linux based operating system to develop cloud applications.</td>
</tr>
</tbody>
</table>
### CLOUD SERVICES DEVELOPED AND TESTED IN THIS STUDY

<table>
<thead>
<tr>
<th>Service Name</th>
<th>Platform Used</th>
<th>Software Used</th>
<th>Operating System</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Sharing</td>
<td>Cloud Stack</td>
<td>Java/Net beans</td>
<td>Linux 6. x</td>
</tr>
<tr>
<td>Data Mapping</td>
<td>Hadoop</td>
<td>Java/Eclipse</td>
<td>Linux 6. x</td>
</tr>
<tr>
<td>CDocManage</td>
<td>Open Stack</td>
<td>Java/Eclipse/Apache Tomcat</td>
<td>Linux 6. x</td>
</tr>
<tr>
<td>CJobMapper</td>
<td>Eucalyptus</td>
<td>Java/Eclipse</td>
<td>Linux 6. x</td>
</tr>
</tbody>
</table>