Security As A Service

Priya R.L.[a] , Asawari Shekatkar[b], Rohan Jethwani[c], Sidhi Pandita[d]

[a]*Lecturer, Department of Computer Engineering, V.E.S. Institute of Technology, Chembur, Mumbai. (Email: priyas_24@rediffmail.com)*
[b]*Student, Department of Computer Engineering, V.E.S. Institute of Technology, Chembur, Mumbai. (Email: asawari1011@gmail.com)*
[c]*Student, Department of Computer Engineering, V.E.S. Institute of Technology, Chembur, Mumbai. (Email: rohanjethwani@yahoo.com)*
[d]*Student, Department of Computer Engineering, V.E.S. Institute of Technology, Chembur, Mumbai. (Email: sidhipandita.09@gmail.com)*

ABSTRACT

As more and more enterprises are deploying their software systems in the cloud environment, cloud security is becoming a mainstream area of information technology. The cloud applications are usually large scale and include a lot of distributed components. This makes it imperative to employ an efficient security strategy. Also, in the existing systems, the control of the data lies solely in the hands of the Cloud Service Provider (CSP). In case the CSP misbehaves, integrity and availability of data shall be compromised. Thus there exists a very low level of security assurance. Also, when an enterprise runs its application on a cloud, it should be given a promise of correct and smooth functioning. This means that a server failure or error should not have any impact on the operation. This spells out the need for fault tolerance.

Through this project we aim to develop a framework that can handle various security issues that can arise in a cloud-based architecture. The project focuses on developing an optimal security solution that can be provided as a service for cloud systems. Thus Security as a Service can be offered. The proposed scheme encompasses mainly two aspects of implementation- Maintaining the integrity of data in storage and Fault tolerance. In addition, it also offers support for dynamic data operations, fast error localization and correction.

Keywords: **Cloud, Fault Tolerance, Integrity of data storage, Security.**

1. INTRODUCTION

Cloud computing is a technology that uses the internet and central remote servers to maintain data and applications. Cloud computing allows consumers and businesses to use applications without installation and access their personal files at any computer with internet access. This technology allows for much more efficient computing by centralizing storage, memory, processing and bandwidth. [3]

In spite of the numerous benefits, cloud architecture poses certain threats to the users. The overall control of data and the applications lies with the Cloud Service Provider (CSP). Several cases have been observed where the CSP has behaved in an unfaithful manner exposing the system to both internal as well as external threats. Also, large cloud applications are very complex and are deployed over multiple cloud components. Failure of a component should not affect the functioning of the application in any way.[6]

Thus, the demand for highly reliable cloud applications is becoming unprecedentedly strong and providing this service is becoming a critical, challenging, and urgently-required research problem. This paper proposes a solution to this problem by suggesting the design of a security service which operates on the cloud as a separate entity, interacts with the application and takes care of security issues, mainly, data integrity and fault tolerance. Thus, the overall control no longer lies with only the CSP and an assurance of security can be given to the users.

The rest of the paper is organized as follows. Section 2 gives the problem statement after which the system is described in Section 3. It explains the proposed system and describes the system architecture along with the various components and their interaction. Then, the implementation details are given in Section 4. Finally, Sections 5 and 6 conclude the paper giving the future enhancements.

2. PROBLEM DEFINITION

2.1 Existing System

Although, cloud based operation provides huge amounts of storage space and customizable computing resources, it relinquishes the control of data from the local machines. The Cloud Service Provider (CSP) handles the availability and integrity of data. Since the users do not possess a local copy of outsourced data, the CSP may give wrong information to the users regarding the status of their outsourced data. For example, to increase the profit margin by reducing cost, it is possible for CSP to discard rarely accessed data without being detected in a timely fashion. Similarly, a CSP may even attempt to hide data loss incidents so as to maintain a reputation. Therefore, although outsourcing data into the cloud is economically attractive for the cost and complexity of long-term large-scale data storage, it lacks the assurance of data integrity and availability which may impede its adoption by enterprise and individual cloud users.

Also, the cloud applications are usually large scale, very complex and typically involve multiple cloud components communicating with each other. In order to transfer their critical systems to the cloud, enterprises need to be certain about the reliability of the cloud application. Thus, the demand for highly reliable cloud applications is becoming unprecedentedly strong and providing this service is becoming a critical, challenging, and urgently-required research problem.

2.2 Proposed System

In order to address these two major concerns, a security service needs to be developed which promises to overcome the limitations faced in the prior scenario. Efficient methods that enable on-demand data correctness verification on behalf of cloud users need to be designed. However, the fact that users no longer have physical possession of data in the cloud prohibits the direct adoption of traditional cryptographic primitives for the purpose of data integrity protection. Hence, the verification of cloud storage correctness must be conducted without explicit knowledge of the whole data files. The proposed security service handles all these issues and provides an overall system for handling cloud applications.[1]

2.3 Problem Statement

This project aims at providing security as a service to cloud based architectures. It proposes a two-fold strategy:[1]

- Security in data storage.
- Fault tolerance for applications running over the cloud.

Significant components in complex cloud applications are identified and a ranking based framework is used to build fault-tolerant cloud applications. It then implements Parallel Processing technique for fault tolerance.

Security in data storage is implemented by performing integrity checks over the data. It employs a challenge token computation for the same. Error localization, error recovery and support for dynamic data operations are also provided. Thus, efficient operation of the applications over the cloud is ensured even in case of faulty servers and data errors.

3. SYSTEM DESCRIPTION

3.1 System Architecture

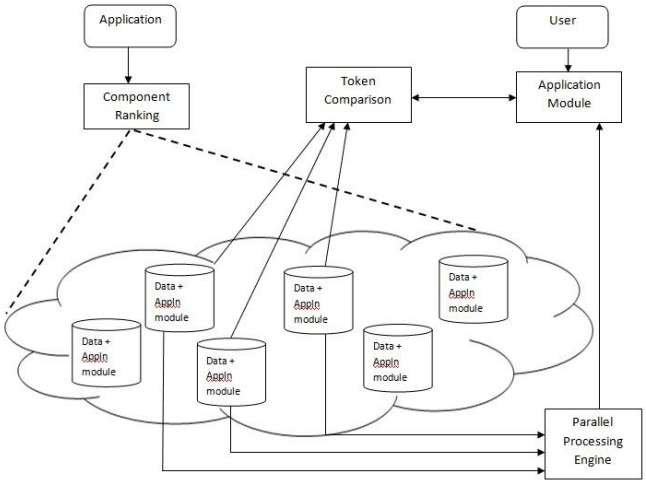

Figure-3a: SaaS System Architecture

The various components of the security system and their interaction with each other have been described in the shown diagram.

The cloud consists of multiple cloud servers which are responsible for storage of data as well as application execution. The data and various application modules are distributed over these servers. A component ranking algorithm identifies the critical and non-critical components which decides the number of cloud servers on which they are deployed. Based on the data requirements of the current application module, the respective cloud servers are invoked by Token Comparison Engine

(TCE). Based on a challenge response strategy, tokens are generated on the respective servers and sent to the TCE. TCE compares the responses and performs error localization and recovery. Correct data is now sent to the application module. To protect the system from a faulty server, multiple servers send their responses to the Parallel Processing Engine (PPE). The fastest of these responses is returned to the application module. Thus, the system is able to function efficiently and correctly even in case of a faulty server.[2]

Detailed description of the key components is as follows:

3.2.1 Component Ranking Engine

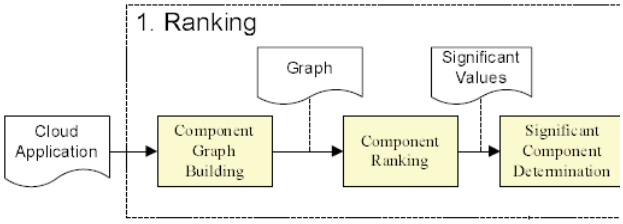

Figure-3b: Component Ranking Engine

The system designer provides the initial architecture design of a cloud application. A component graph is built for the cloud application based on the component invocation relationships.

- Significance values of cloud components are calculated by employing component ranking algorithms. Based on the significance values, the components can be ranked.
- The most significant components in the cloud application are identified based on the ranking results.[2]

3.2.2 Token Comparison Engine

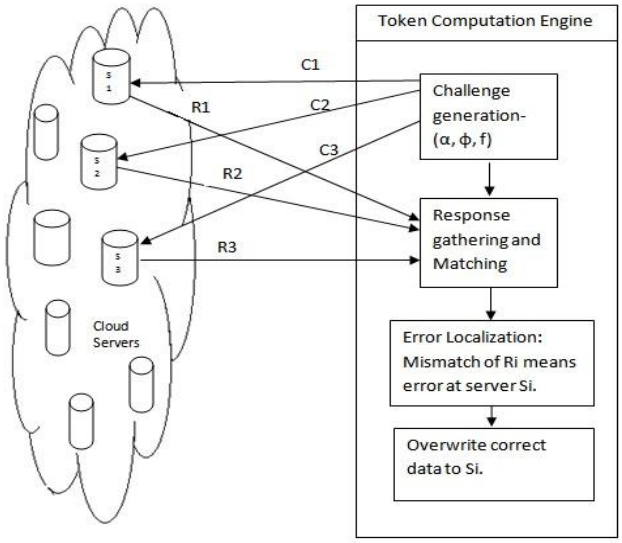

Figure-3c: Token Comparison Engine

All the servers storing the relevant data are challenged with randomly generated block indices. Each server applies a specific token generation function to the indices and produces a token. All servers send their respective tokens to the token comparison engine which compares the received values and locates inconsistencies. The inconsistent value determines the server with erroneous data. Thus error localization is achieved. To recover from this error, correct data from a verified server is overwritten on the faulty file.[4]

Thus, data integrity is ensured.

3.2.3 Parallel Processing Engine (PPE)

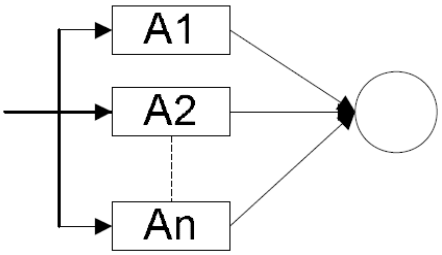

Figure-3d: Parallel Processing Engine

Parallel Processing Engine (PPE) is the strategy employed for fault tolerance. It collects the processed results from the end servers from which a particular application module is deployed. Since many servers return a result, even a faulty server does not hamper the performance of the application. PPE forwards the result from that server which gives the fastest response.[2]

4. IMPLEMENTATION

The data files required by the application are encrypted using a specialized cryptographic algorithm and are then distributed over multiple cloud servers. Component ranking algorithm is applied to the various modules of the application. Fault tolerance is implemented only for the critical components identified in this phase. Based on the obtained results these modules are deployed over multiple cloud servers.

Token computation takes place at the cloud servers. A challenge is sent to multiple data servers whose data needs to be checked. The responses are compared and any deviation or mismatch indicates error at that particular server. This can be corrected by overwriting from a correct source. Thus, error localization as well as correction is facilitated.

To implement fault tolerance, parallel processing technique is employed. Multiple servers on which the application module is deployed perform the computations and return their result to the parallel processing engine. The response which is received first is returned as the final result. In case a server becomes faulty the application is still able to run efficiently and correctly. Thus, the scheme gives a guarantee about correct functioning of the application even in a case of component failure.[5]

5. CONCLUSION

This paper proposes an efficient scheme to address the issue of centralized data storage on cloud. The exclusive control of data security is taken away from the CSP by including an independent security service. The scheme relies on token matching and parallel processing techniques to realize its functionality. It also facilitates simultaneous identification of faulty servers, error localization and error correction on stored data. The proposed scheme ensures efficient performance of cloud system by providing a holistic security system which has been missing in the current systems.

6. FUTURE ENHANCEMENTS

There are a few ways in which the functionality of the service can be enhanced. Certain features that can be added to the system in future are given below:

- Stronger encryption techniques can be used.
- It can be extended to Third Party Auditing.
- Multiple fault tolerant schemes can be used, each suited to the particular application or application module.
- Instead of depending on Application Designer to specify the critical and non-critical components of the application, more factors such as invocation latency, throughput etc. can be considered when computing the critical or non-critical components.[3]

REFERENCES

i. *Component Ranking for Fault-Tolerant Cloud Applications* Zibin Zheng, Tom Chao Zhou, Michael R. Lyu, Irwin King.[1]
ii. *Towards Secure and Dependable Storage Services in Cloud Computing* Cong Wang, Qian Wang, Kui Ren, Ning Cao, Wenjing Lou.[2]
iii. http://searchsecurity.techtarget.com/definition/Security-as-a-Service[3]
iv. http://www.wikinvest.com/concept/Cloud_Computing#_note-versionOneStudy[4]
v. http://msdn.microsoft.com/en-in/ff380142[5]
vi. http://searchcloudcomputing.techtarget.com/definition/cloud-computing[6]