# The Effective Method of Enhancing Data Security in a Cloud Storage System

S. Eswari
Research Scholar,
Head of the Department,
Department o Computer science,
Naina Mohamed College Of Arts & Science
Rajendrapuram

S. Manikandan, M.E., PhD
PROFESSOR & HEAD,
DEPT,OF CS & ENGINEERING
Sriram Engineering College
Perumalpattu
Chennai -602024

## ABSTRACT
Cloud computing presently hot topic in computing as a Service, where data owners can store their files in the cloud server to enable people to access applications and services from anywhere on the network. In these days mobile devices becoming more popular and it is difficult to extent storage in mobile devices to overcome it the cloud storage environment is provided, and it is important to provide users data and provide better service from the cloud. Here after users are not going to install applications for the services as there is less storage rather they can access Web Service. In order to make easier of rapid deployment of cloud Web service and regain security validation with outsourced services, efficient auditing that enable on-demand service validation on behalf of Service Providers have to be designed. In this article it is proposed that secure validation of cloud web service is able to help web service established globally. With service validation a trusted entity with expertise and capabilities of Service Providers do not possess can be delegated as an external audit party to assess the risk of outsourced services when needed. Such an auditing service not only helps save Clients computation resources and also to provide better Services for their Service Request. This concept describes approaches and procedures that should be brought into consideration, and outline challenges that need to be resolved for such a service validation to secure cloud service to become a proper computing.

## General Terms
Cloud Computing, Virtual Computing, Data Security Policies.

## Keywords
Third Party Auditing, Cloud Server, Cloud User. Private key generation.

## 1. INTRODUCTION
Cloud Service is generally improving mechanism in the field of Computer Technology. A cloud computing system is [1] collection of multiple cloud domains servers and every cloud domain has its own responsibilities and service lists, such as the Online Banking, Online Office Editor and Gaming etc. How to efficiently manage the cloud services over multiple cloud domains is difficult for providing proper cloud services. [2] This paper proposes a service validation system for cloud service transfer to secure the computation among multiple cloud domains. This system focuses on maximizing the rewards for both the cloud system and the users by minimizing the number of service rejections that degrade the user satisfaction level significantly.

The optimal service providing decisions are obtained by jointly considering the cloud's request and response process. [3] The validation simulation results show that the proposed service validation system can significantly improve the service quality and decrease service disruptions compared with the greedy approach. Clone Cloud focused on execution augmentation with less consideration on user preference or device status.

The problem of ensuring the integrity of data storage in cloud computing is studied [4]. Although resource management in wireless networks has been extensively studied, it is not well studied in mobile cloud computing. an economic cloud computing model is presented to decide how to manage the computing tasks with a given configuration of the cloud system, i.e., the computing tasks can be migrated between the mobile devices and the cloud servers.

One promising technology for improving Service Quality in Cloud network is Third Party Validation, which eliminates malicious data blocks by validating Services by its XML schema elements [5]. One major process of Service Validation is to analyze XML Content which representing the services, in which each data block is identified by its content type computed from a collision resistant hash of the content of the data block. Recent studies show that the VM images of different versions of the same Linux distribution generally have a high proportion in validating web services.

Other design concerns are generated by the large number of multimedia systems that need to provide services relying on the energy provided by a battery of limited weight and size, the limitation on computational capability of multimedia systems because of heat dissipation issues, and the dependability of multimedia systems operating at high temperatures because of excessive power dissipation [6]. Last but not least, the designing and manufacturing costs are increasingly important since many of the cloud devices have to be affordable in order to fulfill their prospective area of deployment.

Typical Cloud organizes Web Services into index list, which could be of fixed size or variable size. Most TPA systems analyze web services into variable XML representation blocks, so as to exploit different granularities of invalid contents and achieve a higher validation rate. The merits of using validation elements in cloud systems are studied. Nevertheless, it is chosen the standard block implementation in XML Schema validation from Service Providers. First, most commodity CPUs support fixed-size memory pages only.

The main challenge of Web Service validation is about the checking of XML Elements for maliciousness. It shows how DWSRV handles that issue [7]. Whenever a data block arrives at RDVS, its process is generated. The System uses the process of the incoming block to determine if the incoming block is unique. The first checkpoint is the fingerprint filter.

The fingerprint filter is a memory-based filter that aims to determine if the incoming block can be validated. If the incoming block is new to the file system, then it processed based on the algorithm for validation. The design of the fingerprint filter will be detailed in later discussion. Recalling the element filter does not store any complete elements, so the next step is to access the corresponding fingerprint on disk, in order to confirm if the incoming block can actually be validated [8]. If the target xml element does not contain valid information of the web service, then it implies that the DWSRV gives a false-positive result and that the incoming Service is valid.

To solve the existing security issues in the distributed cloud storage currently most of the clouds are using the TPA (Third Party Auditing) tools to validate and verify the services but it is not sufficient to find the true and false users those who intended to damage the cause the byzantine attacks in the cloud, so by done the literature review the proposed theory of research should be a better security auditing policy that can be added into one of the layers of cloud architecture to give better protection for the distributed and increasing storage of clouds on demand.

## 2. RELATED WORK

The type declaring of a Web Service into the documents is encouraged as data can be prototyped in various documents according to object requirements. XML Schema defines two elements and which are both valid data of the element and it is used when two schema elements have the same object and is used to join schema files from different objects. By doing so complex data types can be developed by combining existing files. The WSDL file says how and where to request a service by specifying network protocol and service completion location with the object elements. The WSDL file envelops the XML Schema using Service Elements.

Recent research on cloud computing has been focused on mobile devices of cloud computing, which enables running applications between resource-constrained devices and Internet-based clouds. Moreover, resource constrained mobile devices can outsource computation/communication/storage intensive operations to the mobile cloud.

Specialized hardware-based solutions for best service quality are expensive and may require changes on the applications. Software-based solutions for Service Quality are to provide virtualized execution environment (VM) for applications and fast recovery mechanisms when physical hosts become unavailable. A game theory-based resource allocation model to allocate cloud resources according to the users' requirements is not proposed well. The other mobile cloud computing solutions are limited and solely focused on the enhancement of the individual mobile device's capability. To the best of our knowledge, none of the previous works addressed how to construct a mobile cloud computing system reward model for resource allocation considering the whole rewards of both cloud systems and mobile users and how to select a cloud domain to allocate system resource through inter domain service transfers.

Elastic applications for mobile devices via cloud computing were studied in the earlier research and presented a framework that outsources the antivirus services from mobile devices to a cloud. Goyal and Carter proposed a secure cyber foraging mechanism for resource-constrained devices presented a mobile cloud computing model that allows the mobile device related operations residing either on mobile devices or dedicated VMs in the cloud.

## 3. PROPOSED DESIGN

### 3.1 Client Service Request Metric

The request response process is handled here the process begins when it receives the clients request and response it with the valid services. It should check related or matching Web Service present in the Cloud Server's service list. For example, the request for the file conversion is searched in the Web Service list and responded with proper service. When there is more than one similar service present in the list then the one with Best QOS is selected. As the new service is requested it is not searched over the cloud it can be imported from the neighbor Cloud. Monitoring the number of requests over a given time period can give you an idea of system read workload and usage patterns.

### 3.2 Dynamic Web Service Request Verification With Xml – Schema Elements And Meta Data Processing

Service verification is widely used in Cloud systems to prevent malicious services. In this paper, it is suggest a UDDI verification approach using XML-Schema verification and WSDL verification technique. The existing Service verification struggles with large amount of verification and shows poor performance when handling Web Services files. The key idea of this work is to verify Web Service XML elements. It can be easily verified several Web Service files point by comparing service key value and business entity offset within file similarity information. It is considered these XML elements as a hint for verify and ranking of Services on the Cloud. Using this approach, we can significantly improve the performance of Validation of spamming and virus spreading through the Cloud service. In experiment result, the proposed dynamic Web Service request verification with XML-Schema elements results in significant performance improvement for validated Web Service in the Cloud processing capability and shows fast processing time comparable to the previous method.

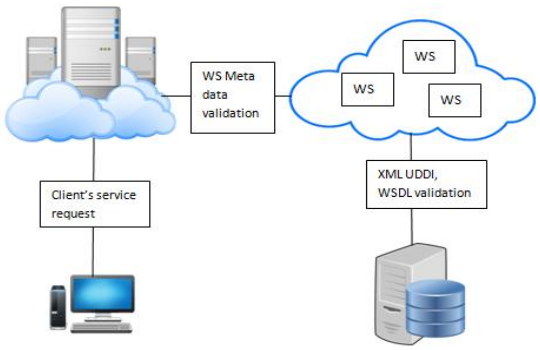



**Fig 1 : shows the Block Diagram of Proposed Security Layer**

### 3.3 Service Provider Verification Based On Service History

Web Service data has a direct impact on Cloud Security and performance. A proper choice of Web Service needs to list for those client of the network. Even though the Web Services are well validated for security it should perform well in the Cloud for clients. So, its service history also be verified before added to the service list of the Cloud. The Cloud will request similar

Service to the neighbor Cloud. It may collect data and XML-Schema from the neighbor Cloud. It will be repeated for every new Service request for every neighbor Cloud. The XML-Schema related to UDDI, WSDL and SOAP services. And the Service History of that Web Service gets extracted from the report gathered from the XML data from Clouds. Based on the elements the Similar Services are ranked in an order. Finally the best ranked services get updated to the Web Service List.

## 3.4 Rank Based Search For Service Provider

The Cloud server in Cloud Network is in charge of managing all Web Server Services. It exchanges a Web Services with every cloud server, in order to keep an up to date Service List. The Cloud server exchanges Service List with cloud peers in a round-robin fashion. The service list will be updated based on the rank gathered in from the neighbor servers. To speedup Service Ranking, whenever a server or client requested for the new service it will be forwarded to neighbor cloud. The Metadata is gathered as information. It is also verified that is it malicious free.

## 3.5 Algorithm: Dynamic Web Service Request Verification With Xml Schema Elements And Meta Data Processing

```
procedure SERVICE_VALIDATION (serviceList)
    boolean serviceAccepted ← false;
    while(serviceList[]!=NULL):
        service ← service_in_the_serviceList
        serviceUDDI ← getServiceUDDI()
        serviceWSDL ← getWSDLfromUDDI()
        WSDL_Elements ← extractWSDL_Elements()
        for elements in WSDL_Elements:
            serviceDefinition ← getDefinition()
            dataType ← getDataType()
            message ← getMessage()
            operation ← getOperation()
            portType ← getPortType()
            binding ← getBinding()
            port ← getPort()
            serviceType ← getServiceType()
            serviceLocation ← getServiceLocation()
        UDDI_Data ← getUDDI_Data()
        for data in UDDI_Data:
            bussinesskey ← getBussinessKey()
            servicekey ← getServiceKey()
            tModelKey ← gettModelKey()
        SOAP_Elements ← getSOAP_Elements()
        for s_element in SOAP_Elements
            envelope ← getEnvelope()
            header ← getHeader()
            body ← getData()
            fault ← getFault()
        if malicious(data) then
            serviceAccepted ← false
        endif
        if invalid(header) then
            serviceAccepted ← false
        endif
        if invalid(dataType) then
            serviceAccepted ← false
        endif
        if invalid(portType) then
            serviceAccepted ← false
        endif
        if unavailable(serviceLocation)then
            serviceAccepted ← false
        endif
        if invalid(bussinessKey) then
            serviceAccepted ← false
        endif
        if invalid(serviceKey) then
            serviceAccepted ← false
        endif
        if invalid(tModelKey) then
            serviceAccepted ← false
        endif
        if valid(envelop) then
            serviceAccepted ← true
        endif
        if valid(header)then
            serviceAccepted ← true
```

***endif***

***if*** *valid(body)* ***then***

*serviceAccepted ← true*

***endif***

***if*** *isPresent(fault)* ***then***

*serviceAccepted ← false*

***endif***

***return*** *serviceAccepted*

***end procedure***

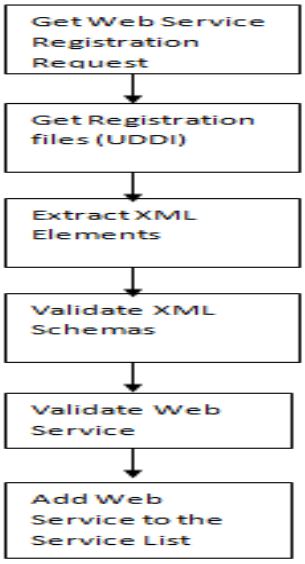



**Fig 2 : shows the process flow of Proposed Algorithm**

The above algorithm performs the process of web service validation. It reads the list of web services and gets its specific data elements likely UDDI, WSDL, SOAP and its XML-schemas. Then parameters like service id, service definition, message, port type, port, envelope etc. All the elements from the web services get validated and verified using the Neighbor Service Provider (NSP). Then the Verified service are in the listed get added to the Service Provider's service list.

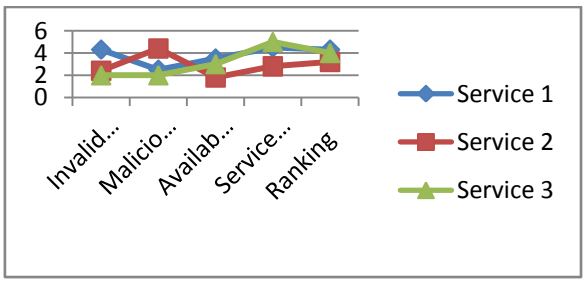
This process is get continued for all the web services which request to register their service in the cloud will be asked for registering their UDDI. UDDI is responsible for the describing, publishing and finding web services over the cloud network. It contains the business key, service key and tModel key to identify the service and its location on the cloud. Actually it is written in WSDL Web Service Description Language. WSDL is the standard format for describing the Web Services. It is used along with SOAP and XML Schema. Its main aim is to describe the process and availability of Web Service over the cloud.

These three element has the desired element which and XML schema used to represent the Web Service and used to register its component in the Cloud. The aim is to analyze and validate these elements to validate the Web Services. This is the thing the above algorithm does.

## 4. RESULT AND DISCUSSION

It is measured the Web Service Verification performance of different Environment using Dynamic Web Service Request Verification with XML – Schema Elements and Meta Data Processing. To test the verification process the example service is created and it is tested to upload and register to the sample Cloud. And the real Cloud Service data is taken in order to validate based on our algorithm. It is keenly giving better performance with the Cloud and the Service Providers and their Services gets ranked properly. Previous System feels struggle and missed its validation quality if the new services are requested by the Clients. As our proposed method of validating Web Service ranking the service based on getting information gathered from the neighbor Cloud it is easy to satisfy Clients new Service request by gathering service based on the rank and validity. Some access problems are detected when the Cloud communicating the Web Service of neighbor Cloud but it is solved using Service Switching process.

It is proposed DWSRV, a proficient Web Service validation file-system that is designed for Cloud Service in an open-source cloud with commodity configurations. DWSRV respects the Cloud design layout and allows general I/O operations such as read, write, modify, and delete, while enabling Service Validation. To support QOS, DWSRV exploits spatial locality to reduce the Service access overhead for looking up XML Elements that are stored on disk. It also supports journaling for Service Identification and Ranking.




**Graph 1 : shows the measuring parameter levels**

DWSRV is implemented as a Service Validation module that can be deployed without the need of modifying the Cloud source. It could be integrate DWSRV into a cloud platform and evaluate the deployment.

## 5. CONCLUSION

This proposed technique improves Service quality of the Cloud network Web Services, while its performance Validation for overall Service Provider (SR) services. The work demonstrates the feasibility of deploying Validation into Cloud Service in an open-source cloud. In this work, we mainly focus on Validating Service Element and Quality on multi cloud platform. Since a cloud platform is typically a distributed system, we plan to extend DWSRV in a distributed setting.

## 6. FUTURE ENHANCEMENT

In future, the challenging issue is to balance the trade-off between storage efficiency and fault tolerance. On one hand, Service Validation reduces the complexity by removing irrelevant and malicious Web Services, on the other hand, it sacrifices fault tolerance with the elimination of redundancy.

## 7. REFERENCES

[1] Collaborative Network Security in Multi-Tenant Data Center for Cloud Computing,Zhen Chen, Wenyu Dong, Hang Li, Peng Zhang, Xinming Chen, and Junwei Cao. 2014.

[2] Enabling Public Auditability and Data Dynamics for

Storage Security in Cloud Computing

[3] Qian Wang, Student Member, IEEE, Cong Wang, Student Member, IEEE, Kui Ren, Member, IEEE, Wenjing Lou, Senior Member, IEEE, and Jin Li.Robust Remote Data Checking Reza Curtmola Department of Computer Science New Jersey Institute of Technology Newark, Osama Khan Randal Burns Department of Computer Science Johns Hopkins University Baltimore.

[4] Provable Data Possession at Untrusted Stores, Giuseppe Ateniese, Randal Burns Reza, Curtmola, Joseph Herring, Lea Kissner, Zachary Peterso, Dawn Song.

[5] Privacy-Preserving Public Auditing for Secure Cloud Storage Cong Wang, Student Member, IEEE, Sherman S.-M. Chow, Qian Wang, Student Member, IEEE, Kui Ren, Member, IEEE, and Wenjing Lou, Member, IEEE.

[6] Toward Publicly Auditable Secure Cloud Data Storage Services, Cong Wang and Kui Ren, Illinois Institute of Technology Wenjing Lou, Worcester Polytechnic Institute Jin Li, Illinois Institute of Technology.

[7] Distributed, Concurrent, and Independent Access to Encrypted Cloud Databases Luca Ferretti, Michele Colajanni, and Mirco Marchetti 2014.

[8] Network Adaptability from Disaster Disruptions and Cascading Failures, Biswanath Mukherjee, M. Farhan Habib, and Ferhat Dikbiyik.

[9] Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage Cheng-Kang Chu, Sherman S.M. Chow, Wen-Guey Tzeng, Jianying Zhou, and Robert H. Deng, Senior Member, IEEE, 2014.

[10] Towards Secure and Dependable Storage Services in Cloud Computing Cong Wang, Student Member, IEEE, Qian Wang, Student Member, IEEE, Kui Ren, Member, IEEE, Ning Cao, Student Member, IEEE, and Wenjing Lou, Senior Member, IEEE.

[11] Efficient Remote Data Possession Checking in Critical Information Infrastructures

[12] Francesc Sebe´ , Josep Domingo-Ferrer, Senior Member, IEEE, Antoni Martı´nez-Balleste ,Yves Deswarte, Member, IEEE, and Jean-Jacques Quisquater, Member, IEEE.