International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-3 Issue-3, August 2013

Data Protection in Cloud Computing

Priyank Rajvanshi, Varun Singh Nagar, Priyanka Chawla

Abstract— We are in the middle of an insurgency in cloud computing. In short, cloud computing is "a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or services provider interaction." Current cloud computing systems pose serious limitation to protecting users' data confidentiality. Since users' sensitive data is presented in unencrypted forms to remote machines owned and operated by third party service providers, the risks of unauthorized disclosure of the users' sensitive data by service providers may be high. Many techniques for protecting users' data from outside attackers are available, but currently there exists no effective way for protecting users' sensitive data from service providers in cloud computing. Our approach is protecting the confidentiality of users' data from service providers, and ensures that service providers cannot access or disclose users' confidential data being processed and stored in cloud computing systems. Our approach has three major aspects:

1) Separating software service providers and infrastructure service providers in cloud computing,

2) Hiding information of the owners of data, and

3) Data obfuscation.

An example to show how our approach can protect the confidentiality of users' data from service providers in cloud computing is given and various types of attacks in cloud computing. Service providers neither can see user's confidential data, nor can modify it. That's approach is presented in our paper.

Keywords- Data confidentiality, Cloud computing system architecture, Data obfuscation, Data de-obfuscation.

I. INTRODUCTION

A. Cloud Computing Architecture

Cloud Computing can be divided into two sections, the user and the cloud. Generally, the user is connected to the cloud via the internet. It is possible for an organization to have a private cloud in which a user is connected via an intranet. However, both scenarios are identical other than the use of a private and public network or cloud. The user sends requests to the cloud and the cloud provides the service. Within the cloud, a central server is responsible for administering the system and in many ways functions as the operating system of the specific cloud network. Another name for this is called "middleware" which is the central server for a particular cloud.

Manuscript published on 30 August 2013.
*****Correspondence Author(s)
Priyank Rajvanshi, Computer Science And Engineering, Amity/ AMITY School Of Engineering And Technology, Noida, India.
Varun Singh Nagar, Computer Science And Engineering, Amity/ AMITY School Of Engineering And Technology, Noida, India.
Priyanka Chawla, Computer Science And Engineering, Amity/ AMITY School Of Engineering And Technology, Noida, India.

Examples include Google App Engine and Amazon EC2 Cloud Computing Architectural Framework, provides a conceptual framework and focus on a description. Computing that is specifically tailored to the unique perspective of IT network and security professionals.

The following three sections define this perspective in terms of the terminology used, to provide a consistent lexicon. The architectural requirements and challenges for securing cloud applications and services act as a reference model that describes taxonomy of cloud services and architectures.

Various components of a cloud:

- **Client computers:** Include various devices, such as desktops and laptops, which use the cloud computing services such as access to database servers, applications servers and storage devices. Application servers: Include various servers, which is either used for running the developed application by the user or is used to providing the application development software to develop the applications by the developer.
- **Network components:** Such as cables, hubs, routers, bridges etc.

Control Node: Controls and monitors the accessibility of various cloud computing services.[18]

B. Introduction to Data Security

Your Data confidentiality is defined as the assurance that sensitive information is not disclosed to unauthorized persons, processes, or devices. Hence, we must make sure that users' data is not disclosed to service providers in any aspect of the cloud computing systems, including applications, platforms, CPU and physical memories.

Cloud computing systems provide various Internet-based data storage and services. Due to its many major benefits, including cost effectiveness and high scalability and flexibility, cloud computing is gaining significant momentum recently as a new paradigm of distributed computing for various applications, especially for business applications. Along with the rapid growth of the Internet, service-oriented architecture (SOA) and virtualization technologies, cloud computing leads to the vision of "Internet as a supercomputer." This vision incorporates the concepts of "software as a service", "platform as a service", and "infrastructure as a service." However, cloud computing has a major limitation to be broadly adopted due to the serious barrier that current cloud computing systems cannot protect the confidentiality of users' data from service providers [1]. A recent survey [2] shows that most of cloud users fear the leakage of their sensitive data in the cloud because their data is processed and stored on remote machines owned and operated by various service providers, which the users do not have any control. Since users' data is processed and stored in cloud computing systems in unencrypted form in current cloud computing systems, there are serious risks of unauthorized uses of the users' data by service providers.

There exist many techniques for confidentiality protection in various computing systems for access control, identity management, end-to-end data confidentiality and integrity assurance, etc. However, these techniques cannot be applied to cloud computing systems for confidentiality protection because they were developed only for protection from malicious third parties outside the systems. Since cloud computing systems have service providers inside the systems as new threat on data confidentiality, the basic ideas about data confidentiality must be changed and an effective new technique for confidentiality protection from service providers of cloud computing systems is needed. This paper is organized as follows. In Section 3, we will articulate users' data confidentiality protection from service providers in cloud computing systems. This concept will be used to determine whether our approach can protect users' confidential data from service providers. In section 4, we will point out security threats in Cloud Computing and literature survey. In Section 7, the problems of current cloud computing architecture in terms of protection of users' data confidentiality will be discussed. In Section 8, we will present our approach, including a new architecture for cloud computing system and using data obfuscation to achieve the above goals of our approach. In Section 9, our data obfuscation and de-obfuscation developed for protecting user's data confidentiality in cloud computing will be discussed.

II. OBJECTIVES

All Our approach has the following specific goals:

- Service users are able to process and store their data in cloud computing systems of various service providers, but the service providers cannot collect or understand users' data.
- Service users can make sure that their sensitive data is not disclosed to the service providers even if there is no cooperation from service providers
- Our approach should not cause much overhead on service performance.

III. LITRATURE SURVEY

Confidentiality is defined as the assurance that sensitive information is not disclosed to unauthorized persons, processes, or devices [3]. Hence, we must make sure that users' data is not disclosed to service providers in any aspect of the cloud computing systems, including applications, platforms, CPU and physical memories. It is noted that users' confidential data is disclosed to a service provider, if all of the following three conditions are satisfied simultaneously:

Condition 1) the service provider knows where the users' confidential data is located in the cloud computing systems.

Condition 2) the service provider has privilege to access and collect the users' confidential data in cloud.

Condition 3) the service provider can understand the meaning of the users' data.

This is due to the following reasons: In order to collect users' data, the service provider must know the location of the data in cloud and have the privilege to access the data. Even if the service provider can collect users' data successfully, the service provider may not be able to understand the meaning of the data. Since every piece of data, including numbers, letters, images, sounds and videos, is in the form of binary numbers in computers, the service providers need to have at the least some of the following information to understand the meanings of the data:

- Types of data
- Functions and interfaces of the application using the data
- Format of the data

Hence, if we can prevent the service provider from satisfying all the three Conditions, we can protect the confidentiality of users' data in cloud computing from the service provider.

IV. THREATS TO CLOUD COMPUTING

Threat 1.) Abuse and Nefarious Use of Cloud Computing

Description: IaaS providers offer their customers the illusion of unlimited compute, network, and storage capacity — often coupled with a 'frictionless' registration process where anyone with a valid credit card can register and immediately begin using cloud services. Some providers even offer free limited trial periods. By abusing the relative anonymity behind these registration and usage models, spammers, malicious code authors, and other criminals have been able to conduct their activities with relative impunity. PaaS providers have traditionally suffered most from this kind of attacks; however, recent evidence shows that hackers have begun to target IaaS vendors as well. Future areas of concern include password and key cracking, DDoS, launching dynamic attack points, hosting malicious data, botnet command and control, building rainbow tables, and CAPTCHA solving farms.

Impact: Criminals continue to leverage new technologies to improve their reach, avoid detection, and improve the effectiveness of their activities. Cloud Computing providers are actively being targeted, partially because their relatively weak registration systems facilitate anonymity, and providers' fraud detection capabilities are limited.

Examples: IaaS offerings have hosted the Zeus botnet, Info Stealer Trojan horses, and download for Microsoft Office and Adobe PDF exploits. Additionally, botnet(s) have used IaaS servers for command and control functions. Spam continues to be a problem — as defensive measure, entire blocks of IaaS network addresses have been publicly blacklist.

Remediation: Stricter initial registration and validation processes, enhanced credit card fraud monitoring and coordination, comprehensive introspection of customer network traffic. Monitoring public blacklists for one's own network blocks.

Threat 2.) Insecure Interfaces and APIs

Description: Cloud computing providers expose a set of software interfaces or APIs that customers use to manage and interact with cloud services. Provisioning, management, orchestration, and monitoring are all performed using these interfaces. The security and availability of general cloud services is dependent upon the security of these basic APIs. From authentication and access control to encryption and activity monitoring, these interfaces must be designed to protect against both accidental and malicious attempts to circumvent policy. Furthermore, organizations and third parties often build upon these interfaces to offer value-added services to their customers. This introduces the complexity of the new layered API; it also increases risk, as organizations may be required to relinquish their credentials to third parties in order to enable their agency.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Impact: While most providers strive to ensure security is well integrated into their service models, it is critical for consumers of those services to understand the security implications associated with the usage, management, orchestration and monitoring of cloud services. Reliance on a weak set of interfaces and APIs exposes organizations to a variety of security issues related to confidentiality, integrity, availability and accountability.

Examples: Anonymous access and/or reusable tokens or passwords, clear-text authentication or transmission of content, inflexible access controls or improper authorizations, limited monitoring and logging capabilities, unknown service or API dependencies.

Remediation: Analyze the security model of cloud provider interfaces. Ensure strong authentication and access controls are implemented in concert with encrypted transmission. Understand the dependency chain associated with the API.

Threat 3.) Malicious Insiders

Description: The threat of a malicious insider is well-known to most organizations. This threat is amplified for consumers of cloud services by the convergence of IT services and customers under a single management domain, combined with a general lack of transparency into provider process and procedure. For example, a provider may not reveal how it grants employees access to physical and virtual assets, how it monitors these employees, or how it analyzes and reports on policy compliance. To complicate matters, there is often little or no visibility into the hiring standards and practices for cloud employees. This kind of situation clearly creates an attractive opportunity for an adversary — ranging from the hobbyist hacker, to organized crime, to corporate espionage, or even nation-state sponsored intrusion. The level of access granted could enable such an adversary to harvest confidential data or gain complete control over the cloud services with little or no risk of detection.

Impact: The impact that malicious insiders can have on an organization is considerable, given their level of access and ability to infiltrate organizations and assets. Brand damage, financial impact, and productivity losses are just some of the ways a malicious insider can affect an operation. As organizations adopt cloud services, the human element takes on an even more profound importance. It is critical therefore the consumers of cloud services understand what providers are doing to detect and defend against the malicious insider threat.

Examples: No public examples are available at this time.

Remediation: Enforce strict supply chain management and conduct a comprehensive supplier assessment. Specify human resource requirements as part of legal contracts. Require transparency into overall information security and management practices, as well as compliance reporting. Determine security breach notification processes.

Threat 4.) Shared Technology Issues

Description: IaaS vendors deliver their services in a scalable way by sharing infrastructure. Often, the underlying components that make up this infrastructure (e.g., CPU caches, GPUs, etc.) were not designed to offer strong isolation properties for a multi-tenant architecture. To address this gap, a virtualization hypervisor mediates access between guest operating systems and the physical compute resources. Still, even hypervisors have exhibited flaws that have enabled guest operating systems to gain inappropriate levels of control or influence on the underlying platform. A defense in depth strategy is recommended, and should include compute, storage, and network security enforcement and monitoring. Strong compartmentalization should be employed to ensure that individual customers do not impact the operations of other tenants running on the same cloud provider. Customers should not have access to any other tenant's actual or residual data, network traffic, etc.

Impact: Attacks have surfaced in recent years that target the shared technology inside Cloud Computing environments. Disk partitions, CPU caches, GPUs, and other shared elements were never designed for strong compartmentalization. As a result, attackers focus on how to impact the operations other cloud customers, and how to gain unauthorized access to data.

Examples: Joanna Rutkowska's Red and Blue Pill exploit Kortchinksy's Cloud Burst presentations.

Remediation: Implement security best practices for installation/configuration. Monitor environment for unauthorized changes/activity. Promote strong authentication and access control for administrative access and operations. Enforce service level agreements for patching and vulnerability remediation. Conduct vulnerability scanning and configuration audits.

Threat 5.) Data Loss or Leakage

Description: There are many ways to compromise data. Deletion or alteration of records without a backup of the original content is an obvious example. Unlinking a record from a larger context may render it unrecoverable, as can storage on unreliable media. Loss of an encoding key may result in effective destruction. Finally, unauthorized parties must be prevented from gaining access to sensitive data. The threat of data compromise increases in the cloud, due to the number of and interactions between risks and challenges which are either unique to cloud, or more dangerous because of the architectural or operational characteristics of the cloud environment.

Impact: Data loss or leakages can have a devastating impact on a business. Beyond the damage to one's brand and reputation, a loss could significantly impact employee, partner, and customer morale and trust. Loss of core intellectual property could have competitive and financial implications. Worse still, depending upon the data that is lost or leaked, there might be compliance violations and legal ramifications.

Examples: Insufficient authentication, authorization, and audit (AAA) controls; inconsistent use of encryption and software keys; operational failures; persistence and reminisce challenges; disposal challenges; risk of association; jurisdiction and political issues; data centre reliability; and disaster recovery.

Remediation: Implement strong API access control. Encrypt and protect integrity of data in transit. Analyzes data protection at both design and run time. Implement strong key generation, storage and management, and destruction practices. Contractually demand providers wipe persistent media before it is released into the pool. Contractually specify provider backup and retention Strategies.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Threat 6.) Account or Service Hijacking

Description: Account or service hijacking is not new. Attack methods such as phishing, fraud, and exploitation of software vulnerabilities still achieve results. Credentials and passwords are often reused, which amplifies the impact of such attacks. Cloud solutions add a new threat to the landscape. If an attacker gains access to your credentials, they can eavesdrop on your activities and transactions, manipulate data, return falsified information and redirect your clients to illegitimate sites. Your account or service instances may become a new base for the attacker. From here, they may leverage the power of your reputation to launch subsequent attacks.

Impact: Account and service hijacking, usually with stolen credentials, remains a top threat. With stolen credentials, attacker scan often access critical areas of deployed cloud computing services, allowing them to compromise the confidentiality, integrity and availability of those services. Organizations should be aware of these techniques as well as common defense in depth protection strategies to contain the damage (and possible litigation) resulting from a breach.

Examples: No public examples are available at this time.

Remediation: Prohibit the sharing of account credentials between users and services. Leverage strong two-factor authentication techniques where possible. Employ proactive monitoring to detect unauthorized activity. Understand cloud provider security policies and SLAs.

V. PROTECTION OF USER'S DATA CONFIDENTIALITY IN CLOUD

Design is the stage of when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

VI. PROPOSED SYSTEM

In this paper, we suggest an efficient and supple spreading system with open dynamic data support to make sure the accuracy of user's data in the cloud. We use erasure correcting code in the file circulation preparation to give redundancies and guarantee the data soundness. This construction considerably reduces the messaging and storage in the clouds as compared to the conventional replication-based file division techniques. By using the homomorphism token with distributed confirmation of erasure-coded data, our system achieves the storage rightness cover as well as data error localization: whenever data corruption has been detected during the storage correctness confirmation, our scheme can guarantee the concurrent localization of data errors, i.e., the identification of the misbehaving server(s).

1. Compared to many of its predecessors, that provide binary results for the storage state on distributed servers, the challenge-response protocol in our work further provides the localization of data error.
2. The prior works for ensuring remote data integrity, the new system supports secure and efficient dynamic operations on data blocks, including: update, delete and append.
3. Widespread security and performance analysis shows that the proposed scheme is highly efficient and resilient against Byzantine failure, malicious data modification attack, and even server colluding attacks.

VII. PROBLEMS WITH CURRENT CLOUD COMPUTING ARCHITECTURE

The current cloud computing system consists of three layers: software layer, platform layer and infrastructure layer, as shown in Fig.1. The software layer provides the interfaces for users to use service provider's applications running on a cloud infrastructure. The platform layer provides the operating environment for the software to run using system resources. The infrastructure layer provides the hardware resources for computing, storage, and networks. Platforms or infrastructures can be provided as virtual machines.

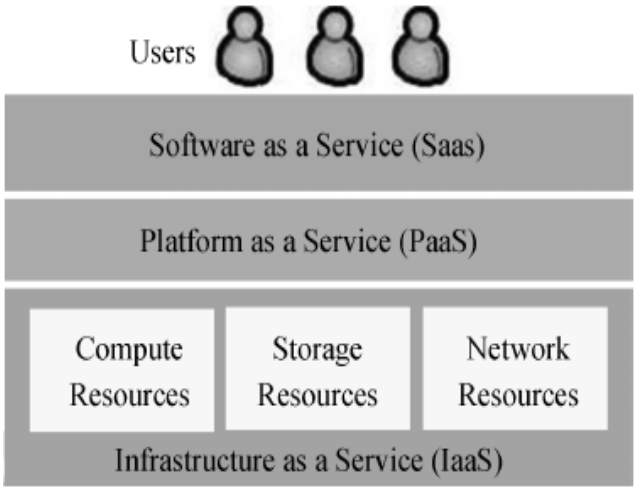

Fig. 1 Current cloud computing architecture

The following are the major problems of current cloud computing systems:

- Each service provider has its own software layer, platform layer and infrastructure layer. When a user uses a cloud application from a service provider, the user is forced to use the platform and infrastructure provided by the same service provider, and hence the service provider knows where the users' data is located and has full access privileges to the data.
- The user is forced to use the interfaces provided by the service provider, and users' data has to be in a fixed format specified by the service provider, and hence the service provider knows all the information required for understanding users' data.

VIII. OVERVIEW OF OUR APPROACH

Our approach can be depicted in Fig.2. In our approach, we have the following seven entities to protect the confidentiality of data processed and stored in cloud computing systems: Software Cloud, Infrastructure Cloud, Software Service Broker, Infrastructure Service Broker, Software Service Attestation Authority, Data Obfuscator and Data De-obfuscator [14].

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

The Software Cloud and Infrastructure Cloud have the same features of the software layer in the ordinary cloud computing architecture discussed in previous section. However, in our approach, we require that the software layer and infrastructure layer are not managed by the same service provider. The Software Service Brokers and Infrastructure Service Brokers have the same functionality of the service brokers in service-oriented architecture (SOA), but they have the additional function for identity-anonymization. The Software Service Attestation Authority, Data Obfuscator and Data De-obfuscator are additional entities introduced in our approach. Our approach makes sure that any of these entities in a cloud computing system does not satisfy the three conditions simultaneously in Section 6.

We will describe each of these entities below:

- **Software Cloud:** A Software Cloud provides software as a service upon users' requests. Each software cloud may contain multiple software services, and each software service can be discovered and accessed by users through Software Service Broker. An authenticated user with proper credentials [4] can request and acquire a service instance from the software cloud. A service instance is a piece of compiled executable code. The executable code can be deployed and run on any Infrastructure Cloud. In order to protect the intellectual property of the software, the code is compiled using various code obfuscation technologies [5, 6] so that reverse engineering on the service instance is this implies that Infrastructure service providers cannot understand the functionalities and algorithms of the service instance by looking at the code when the service instance is running on Infrastructure Cloud.
- **Infrastructure Cloud:** Infrastructure Cloud provides virtualized system resources, such as CPU, memory, and network resources. An authenticated user can request a virtual machine on which the user can deploy any platform or operating system to execute a software service instance.
- **Software Service Broker:** A Software Service Broker has two major functions. First, it helps users automatically discover and access available software services. Second, it helps users hide their identities from software cloud service providers. A Software Service Broker provides identity anonymization service, by which users can use pseudonyms instead of their true identities so that the users can acquire service instances without revealing their identities. The anonymization of user identities is very important for protecting the confidentiality of users' data because information about the data owners may reveal a lot of sensitive information regarding the data.
- **Infrastructure Service Broker:** An Infrastructure Service Broker also has two major functions similar to a Software Service Broker. It helps users automatically discover and use available infrastructure services. It also provides identity anonymization service to prevent the system from revealing users' true identities.
- **The Software Service Attestation Authority (SSAA):** The SSAA is a third party authority to verify that a service instance does not perform any malicious activity that may disclose users' confidential data. For example, a software service developer may have injected a hidden function on the software service which transmits user's confidential data to an unauthorized third party during its process without the user's consent. Since users do not know whether a service instance will act as described in the service description, the SSAA needs to help users test the service instance before users using it. When a service instance is deployed on the Service Testing Platform of SSAA, the SSAA tests whether the service instance performs exactly what the service provider claims, and whether the service instance may transmit users' data to any unauthorized entity. The testing can be done by 1) verifying whether the service instance satisfies the web service description language (WSDL) specification of the service, and 2) monitoring all the network traffics the software service produces during its process. An approach for automated web service testing based on syntactic and semantic analysis of WSDL specification was presented in ref. [15]. After completing the testing of the service instance, SSAA issues a digital certificate for the tested service instance. A certificate is attached to the service instance so that users will know whether the service instance has passed the SSAA's testing.
- **A Data Obfuscator:** A Data Obfuscator is a middle ware provided by a user that can be deployed on a virtual machine in an Infrastructure Cloud. The Data Obfuscator provides an operating system environment for software service instance to be run in an Infrastructure Cloud. The service instance in an Infrastructure Cloud can use system resources only through the interfaces of the Data Obfuscator. The Data Obfuscator has the following security tasks to ensure that users' confidential data is not disclosed to infrastructure service providers:

1) Sets up an encryption key with the user. The key is chosen by the user and not revealed to any process, platform or device of the cloud computing system.
2) Encrypts any data being stored in the physical storage of the cloud or being transmitted through the network.
3) Obfuscates sensitive data being processed in the service instances. The obfuscated data cannot be de-obfuscated in the platforms or physical devices of an Infrastructure Cloud so that its infrastructure provider cannot understand the meaning of the sensitive data.

- **A Data De-obfuscator:** It de-obfuscates obfuscated data so that a user can see the plain data. A Data De-obfuscator remains in the user's personal computer all the time.

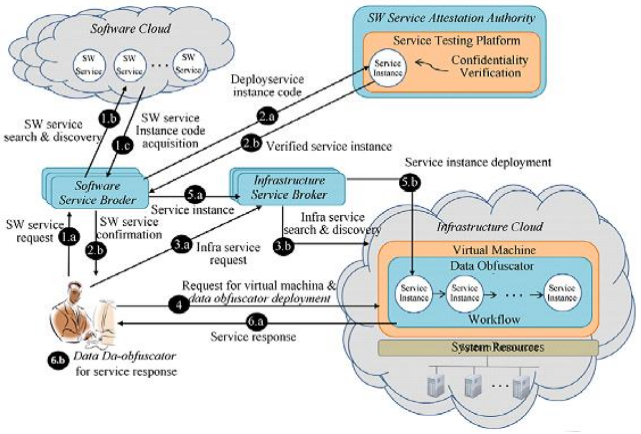

Fig.2 Structure of Approach

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Our approach is based on three features:

1) Separation of software service providers and infrastructure service providers,
2) Hiding information about the owner of data and
3) Data obfuscation.

With these three features, our approach can ensure that at least one of the three conditions in Section 5 is not satisfied in each of the seven entities in cloud computing system due to the following reasons:

• Although a software service provider knows the functionality of a service instance and the data format of users' input and can satisfy Condition 3), since software service instances are deployed to Infrastructure Cloud through its Software Service Broker and Infrastructure Service Broker, the software service provider does not know where users' data is located, and does not have the privilege to access the data. Thus, the software service provider cannot satisfy Conditions 1 and 2.

• Although an infrastructure service provider knows the locations of users' specific confidential data and has the privilege to access the data being processed and stored in the infrastructure cloud, the infrastructure service provider cannot understand the meaning of the data because

i) He / She do not know the functionality of the deployed software instance and the format of the users' specific confidential data.

ii) The deployed software instance cannot be reverse-engineered.

iii) The infrastructure service provider does not know the identity of the data owners

iv) The users' specific confidential data is obfuscated while processed and stored in the infrastructure. Hence, the infrastructure service provider cannot satisfy Condition 3.

• Because the Software Service Broker does not know where users' data is located, and does not have the privilege to access user's data, the Software Service Broker cannot satisfy Conditions 1 and 2.

• Because an Infrastructure Service Broker does not know the functionality of deployed software instance, the data format of user's data, and does not have the privilege to access the data, the Infrastructure Service Broker cannot satisfy Conditions 2 and 3.

Our approach can be summarized as follows, and depicted, where the numbers are corresponding to the steps:

S1)

a) A user requests a Software Service Broker to find a software service by providing the specification of the software service [16].
b) The Software Service Broker performs automatic service discovery [17] to find a service instance in the Software Cloud that satisfies the user's requested service requirement specification.
c) The Software Service Broker acquires the discovered software instance using an anonymous credential [19].

S2)

a) The Software Service Broker deploys the acquired service instance to the testing platform of a SSAA. The SSAA verifies whether the service instance performs according to the service description, and the service instance does not transmit users' data to any unauthorized entity.
b) After the verification procedure, the software service instance is sent back to the Software ServiceBroker.

S3)

a) The user asks the Infrastructure Service Broker to find an infrastructure service compatible to the service instance.
b) The Infrastructure Service Broker discovers an infrastructure service provider, who has the capability to execute the acquired software service instance.

S4)

The user requests the infrastructure service provider to set up a virtual machine and then deploys the Data Obfuscator on the virtual machine using the Agent Deployment Plans (ADPs), for automated middleware deployment and migration in service based systems [7].

S5)

a) The service instance acquired in S1) is sent to Infrastructure Service Broker. b) The service instance is deployed on the workflow of the Data Obfuscator set up in S4).

S6)

a) The user sends his/her data to the workflow to process. During the processing of users' input data, the user's data is obfuscated so that the infrastructure service provider cannot understand the meaning of user's data. After completing the processing, a service response of the workflow is sent to the user indicating that the processing of the user's input data has been completed.
b) The service response is de-obfuscated to plain data in the user's computer.

IX. DATA OBFUSCATION IN INFRASTRUCTURE CLOUDS

Data obfuscation is the process of transforming the format or structure of data to hide the meaning of data. The major difference between encryption and obfuscation is that encrypted data cannot be processed until it is decrypted, but obfuscated data can be processed without de obfuscation. In our approach, data obfuscation is used to process users' data in an infrastructure cloud without revealing any users' specific confidential data to the infrastructure service providers.

An approach to obfuscating data, which is transmitted from a user to software layer of a cloud computing system for protecting user's privacy, is available. How- ever, this approach is limited in the use of data obfuscation because the obfuscated data must fit into the user interfaces provided by service providers. In our approach, data obfuscation takes place between the software layer and the infrastructure layer as shown in Fig.2 so that the use of data obfuscation is not constrained by the user interfaces provided by the service providers.

The general algebraic description of data obfuscation is as follows. Suppose a user wishes to use an infrastructure cloud service to process a function F on the user's input x without revealing the meaning of x to the infrastructure service provider.

Obfuscation function O and de-obfuscation function D have following properties:

• D(F(O(x; k); k) = F(x), where k is an obfuscation key unknown to the infrastructure service provider

• The infrastructure service provider cannot understand the meaning of x by examining O(x; k)

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

- D or k cannot be derived from O
- O(x; k) and D(F(O(x; k); k) can be calculated in polynomial time

When a data obfuscator is deployed by a user on a virtual machine in the Infrastructure Cloud, the data obfuscator obfuscates the user's input data x using one of the following three methods:

Method A)

Take dummy input data from a user and generate arbitrary dummy outputs. In our approach, a user's input data is entered to a software service instance through the data obfuscator. Since the infrastructure service provider does not know the input data format of the service instance, the data obfuscator can take any number of dummy input data from the users. Only the user's inputs to be processed are given to the software service instance, and the data obfuscator generates arbitrary dummy outputs on the user's dummy inputs. The outputs generated by the service instance from the user's inputs, and the dummy outputs generated from the user's dummy inputs by the data obfuscator are mixed and encrypted together, and sent back to the user. The dummy outputs are marked so that data de-obfuscator in the user's computer can recognize the dummy outputs after the all the outputs of the service instance are decrypted.

Method B)

Use a file system not known to all the infrastructure providers. A file system is a method of storing and organizing files and data into computer memories and storage devices, such as hard disks or CD-ROMs. If an infrastructure service provider can understand the file system structure of a user's operating system running on the infrastructure cloud, the service provider may be able to locate and extract users' data from memories or storage devices. Data obfuscator uses a file system which is not known to any of the infrastructure providers so that the infrastructure service providers cannot extract meaningful data from the memory or storage devices of the cloud computing system.

Method C)

Transform users' input data. Data obfuscator transforms the format or value of users' input data so that the infrastructure service provider cannot understand the meaning of the data. All the operations associated with the original data must also be applicable to the transformed data so that the software service instance is able to process the obfuscated data.

X. CONCLUSION AND FUTURE SCOPE

In this paper, we have presented an approach to protecting users' confidential data in cloud computing from cloud service providers. Our approach is based on new cloud system architecture, which has three features:

1) Separation of software service providers and infrastructure service providers,

2) Hiding information about the owner of data and

3) Data obfuscation. Our cloud system architecture ensures that cloud service providers cannot know location of the users' data, access the user's data, or understand the meaning of the user's data simultaneously. Our experimental results on the performance of our data obfuscation and de-obfuscation show that the overhead for data obfuscation and de-obfuscation appears to increase linear with the size of input data. Hence, our approach is scalable with size of input data.

Our future research includes incorporation of the dynamic resource allocation in our cloud computing architecture to support multiple QoS, including security aspects. We'll also include load balancing concept in this approach so that when number of requests come then resources can be easily allocated.

REFERENCES

1. Horrigan J. Use of cloud computing applications and services. Pew Internet and American LifeProject Memo. 2008.
2. Heiser J, Nicolett M. Assessing the security risks of cloud computing. Gartner Report, 2009,http://www.gartner.com/DisplayDocument?id=685308.
3. DoD Trusted Computer System Evaluation Criteria, http://csrc.nist.gov/publications/history/dod85.pdf
4. Iwaihara M, Murakami K, Ahn GJ, et al. Risk evaluation for personal identity managementbased on privacy attribute ontology. Proc. 27th Int'l Conf. Conceptual Modeling (ER 2008). 2008. 183-198.
5. Mateas M, Michael N. A Box, Darkly: Obfuscation, Weird Languages, and Code Aesthetics.Proc. 6th Digital Arts and Culture Conference. 2005. 144-153.
6. Ertaul L, Venkatesh S. Novel obfuscation algorithms for software security. Proc. Int'l Conf. on Software Engineering Research and Practice. 2005. 209-215.
7. Yau SS, Zhu L, Huang D, Gong H. An approach to automated agent deployment in service-based systems. Proc. 10th IEEE Int'l Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC). 2007. 257-264.
8. Mateas M, Michael N. A Box, Darkly: Obfuscation, Weird Languages, and Code Aesthetics. Proc. 6th Digital Arts and Culture Conference. 2005. 144-153.
9. Ertaul L, Venkatesh S. Novel obfuscation algorithms for software security. Proc. Int'l Conf. on Software Engineering Research and Practice. 2005. 209-215.
10. Dong W, Yu H. Web service testing method based on fault-coverage. Proc. 10th IEEE Int'l Enterprise Distributed Object Computing Conference Workshops. 2006. 43-49.
11. Mowbray M, Pearson S, A client-based privacy manager for cloud computing. Proc. Conf. Communication System Software and Middleware. 2009. 138-145.
12. Yau SS, Yin Y. A privacy preserving repository for data integration across data sharing services. IEEE Trans. Services Computing, 2008, 1(3): 130-140.
13. Ritika Agarwal , Ishita Agarwal, Analysis of Cloud Computing Security.
14. Yau SS, An HG. Protection of users' data confidentiality in cloud computing. Proc. 2nd Asia-Pacific Symposium on Internetware. 2010. 32-37.
15. Dong W, Yu H. Web service testing method based on fault-coverage. Proc. 10th IEEE Int'l Enterprise Distributed Object Computing Conference Workshops. 2006. 43-49.
16. Gibson J. Developing A requirements specification for a web service application. Proc. 12thIEEE Int'l Conf. Requirements Engineering. 2004. 340-344.
17. Kona S, Bansal A, Gupta G, et al. Web service discovery and composition using USDL. Proc.3rd IEEE Int'l Conf. E-Commerce Technology. 2006. 65-69.
18. Cloud Security Alliance. Cloud Computing Architectural Framework. January 2011.
19. Damodaram A, Jayasri H. Authenticatio without identification using anonymous credential system. Int'l Jour. Computer Science and Information Security (IJCSIS), 2009, 3(1): 34-37.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)