© August 2015 | IJIRT | Volume 2 Issue 3 | ISSN: 2349-6002

SCALABLE DISTRIBUTED SERVICE INTEGRITY ATTESTATION

Poola Ravisekhar[1], V. Sujatha[2]
[1]*M.Tech student, CSE, Kottam college of Engineering,*
[2]*Assistant Professor, CSE , Kottam college of Engineering,*
Chinnatekuru(V),Kurnool,Andhra Pradesh, India

***Abstract*- Software-as-a service (SaaS) makes use of a cloud computing infrastructure to deliver their applications to many users regardless of their location. Because of this sharing nature SaaS clouds are vulnerable and provide more opportunities for attackers to exploit the system vulnerability and perform strategic attacks. In this paper, we present IntTest, an effective service integrity attestation framework for SaaS clouds. IntTest provides an integrated graph attestation analysis method that can pinpoint malicious service providers than existing methods. Also IntTest will automatically correct the corrupted result that are produced by the malicious service providers and replace it with good results produced by benign service providers. Our experimental results show that our scheme is effective and can achieve higher accuracy in pinpointing the attackers than the existing approaches.**

***Index Terms*- Distributed service integrity attestation, cloud computing, secure distributed data processing**

I. INTRODUCTION

In recent days the cloud computing technology is popular because it is an attracting technology in the field of computer science. Cloud computing is internet base computing that usually referred the shared configurable resources is provided with computers and other devices as services. Cloud computing delegate services with a customer's data, software and computation over a network. The customer of the cloud can get the services through the network. In other words, users are using or buying computing services from others. Cloud can provide Anything as a Service (AaaS). Many service model are provided by the cloud they are IaaS,SaaS and PaaS.Infrastructure as a service (IaaS) offer computers physical or virtual machines and other resources. Infrastructure as a service (IaaS) clouds often offer additional resources such as a virtual-machine disk image library, raw block storage, and file or object storage, firewalls, load balancers, IP addresses, virtual local area networks (VLANs), and software bundles. Infrastructure as a service (IaaS) cloud providers supply these resources on demand from their large pools installed in data centers.

In the Platform as a service (PaaS) models, cloud providers deliver a computing platform, typically including operating system, programming language execution environment, database, and web server. Application developers can develop,run their software solutions on a cloud platform without the cost complexity of buying and managing the underlying hardware,software layers. With some Platform as a service (PaaS) offers like Microsoft Azure and Google App Engine, the underlying computer and storage resources scale automatically to match application demand so that the cloud user does not have to allocate resources manually.

In large-scale multitenant cloud systems, large number of malicious attackers may launch colluding attacks on the targeted service functions to make them malicious. To address this challenge, IntTest takes a holistic approach by systematically examining both consistency and inconsistency relationships among different service providers within the entire cloud system. IntTest checks both per-function consistency and the global inconsistency graphs. An advantage of using this IntTest is it cannot only pinpointing the malicious attackers more efficiently but also it can suppress aggressive attackers and also limit the scope of damage that are caused by the attacks.

The experimental result shows that IntTest can achieve more accuracy in pinpointing malicious attackers than any other existing schemes. Also this IntTest is more scalable and it will reduce overhead produced by the attestation more than the other voting schemes.This paper implements

- Efficient and distributed service integrity attestation framework for large scale cloud computing infrastructures.
- An integrated service integrity attestation scheme that can achieve higher pinpointing accuracy than existing techniques.
- A result auto correction technique is used that will automatically correct the corrupted results produced by malicious attackers and replace it with good results.

Fig.1 shows the integrity attacks in software as a service clouds. Majority of software as a service cloud solutions are based on a multitenant architecture.

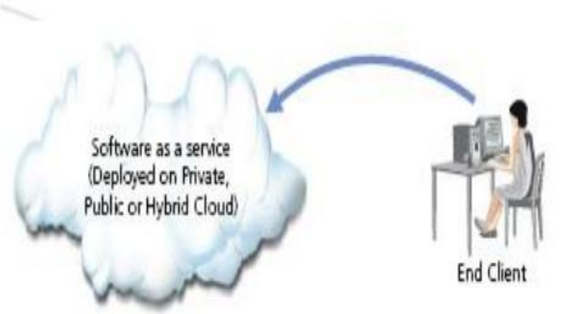


Fig. 1: Software-as-a Service

Our work focuses on service integrity attacks that cause the user to receive untruthful data processing results, illustrated by Fig. 2.

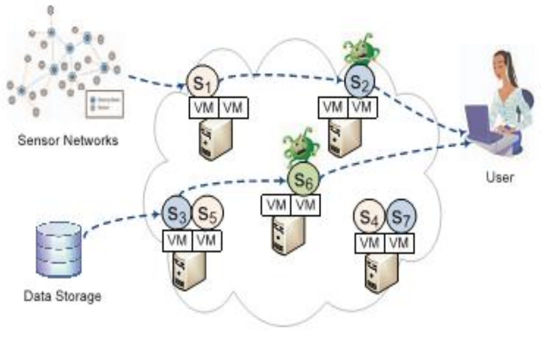


Fig. 2. Service integrity attack in cloud-based data processing. S_i denotes different service component and V M denotes virtual machines.

II. RELATED WORKS

In recent years many integrity attestation schemes have been developed for software as a service clouds. For example the BIND technique, AdapTest technique, RunTest technique etc. but all of these are having some problems some of them needs secure kernel support and special trusted hardware components. In BIND (Binding Information and Data) technique is a verification method of integrity services that are provided by the software as a service cloud system. It was a fine grained attestation framework and can provide the verification through a secure kernel or by a third party. This technique uses the following steps:

1) attestation annotation mechanism 2) sandbox mechanism 3) verification of authenticator through hash. BIND method uses the DiffeeHellman key exchange for the purpose of integrity attestation. Another existing technique is TEAS (Timed Executable Agent System) this is used for protecting the integrity of cloud computing platforms. An agent generation And the verification algorithm is used in this TEAS method.

Another one existing technique is the run test, it is a scalable runtime integrity attestation framework. It provides a light weight application level attestation method to assure the integrity of daa flow processing in cloud. This will will identify the untruthful data flow processing and will pinpoint mallicious data processing service provider and atlast it will detect the attackers behaviour. This RunTest will provide the benign service providers and will determine the malicious behaviour of the attackers. But the disadvantage is its low performance. The AdapTet is another one existing technique, it provides a novel adaptive data driven runtime service integrity attestation frmaework. This method will significantly reduce the overhead of attestation and will shoten the delay. It treats all components as black boxes and it does not need any special hardware or software requirements. In this AdapTest it will reduce the attestation overhead and the detection of malicious attackers or service providers will be high when compared to other techniques.

All the above methods that are used in the existing papers are having some disadvantages. And to overcome that disadvantages this IntTest is using. And by using this IntTest it will provides more integrity and it will provide more accuracy in pinpointing the malicious attackers and service providers. Also it will provide a result auto correction method and will correct the bad results and replace it with good results and also in this it does not require any special hardwares and secure kernel support.

III. PROPOSED PRIVACY PRESERVING AUTHENTICATED ACCESS CONTROL SCHEME

In this paper, we will provide a broad overview of the different techniques for verifying the service integrity. We will provide a broad view of the major algorithms available for each method, and the variations on the different techniques.

Fig.3 shows the over all aarchitecture of the proposed system. In this the user give request to cloud the servcie will be deployed in the cloud the cloud will forward the user request to the SaaS and the response will be send to the cloud by the SaaS. And then the IntTest process will be done. After that the result auto correction will be done. After that the result will be send to the user by the cloud. The architecture shows this IntTest module in detail.

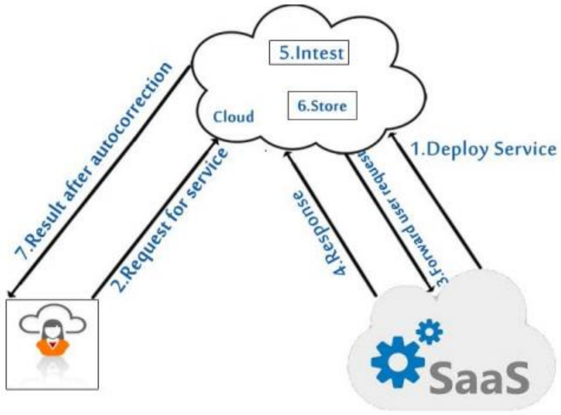


Fig.3: Over all architecture of the proposed method

A. Modules

In this section we present the main modules in the proposed system. Mainly it consists of four modules that are described below.

Baseline Attestation Scheme

IntTest is used to detect the service integrity attack and to pinpoint malicious service providers. For that first we are deriving the consistency and inconsistency relationship between service providers. Consider the fig. 4 it shows the consistency check method. In that p1,p2 and p3 are the service providers. All of them offers the same function f. The portal sends the original data d1 to the service providersp1 and gets the processing result f(d1). Then the portal sends the duplicate of d1 to p3 and gets the result f(d1'). And if both of them are same means it is consistent and if not means they are inconsistent.that is if two service providers disagree with each other, when processing the same input then any one of them will be malicious. Thus the malicious attackers cannot escape from detecting when they are providing bad results with good results.

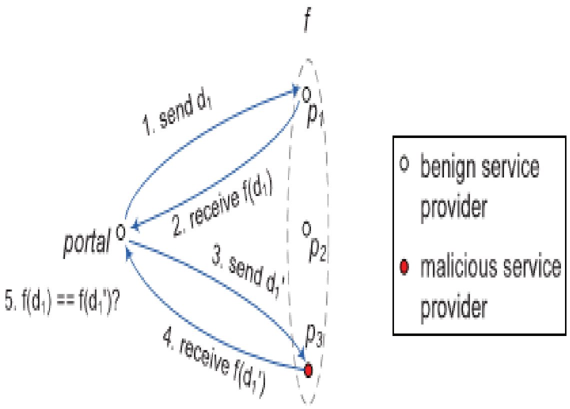


Fig. 4 Replay-based consistency check.

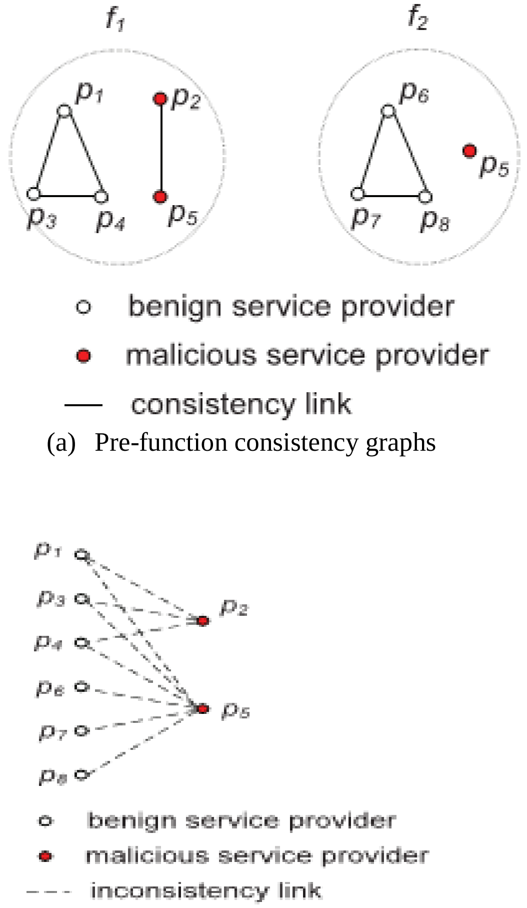
Integrated Attestation Scheme

Here we present an integrated attestation graph analysis algorithm.

Step 1: Consistency analysis: In the first step it will examine the per-function consistency graph and will pinpoint suspicious service providers. The consistency links in the consistency graph will provide a set of service providers. providers will form a clique in terms of consistency links. For example, in Fig. 5a, p1, p3 and p4 are benign service providers and they always form a consistency clique. It will keep consistent with each other on a specific service function. The benign service providers will always keep consistent with each other and will form a clique in terms of consistency links. The colluding attackers can try to escape from being detected. Then next we must examine the perfunction in consistency graph too.

Step 2: Inconsistency analysis: This inconsistency graph will contain only the inconsistency links, this may exist in different possible combinations of the benign node and the malicious node set. For example, in Fig. 5b, p2 and p5 form the minimum vertex cover.

First we assume that the total number of malicious service providers in the cloud system is not more than the benign service providers, then we can pinpoint a set of malicious service providers. If two service providers are connected by an inconsistency link, we can say that any one of them is malicious.



(a) Pre-function consistency graphs

(b) Global inconsistency graphs

Fig. 5. Attestation graphs.

In a shared cloud infrastructure ,malicious attacker can pretend to be legitimate service provider to provide fake service instance or compromised vulnerable benign service instance by exploiting their security roles. It consist of different form of malicious which are described below.

1) Malicious Intermediary

A malicious intermediary may arbitrarily alter and inject protocol data. To prevent such attacks, we can employ cryptographic construction such as message authentication codes or digital signatures.

2) The Data Misuse Attack

It uses authenticated protocol data in a malicious way. For instance , a malicious intermediary can perform a data suppression attack by effusing to forward any data. Then the attacker can perform the replay attack by replaying data that have been authenticated but are outdated.

3) Malicious process and the Data Falsification Attack

In a highly adversarial environment , an attacker may corrupt one or more process in the system. A malicious process is capable of injection bogus data into distributed system. We refer to this attack as the data falsification attack.

4) Non-collusion Always Misbehave(NCAM)

Malicious component always act independently and always give incorrect results. It correspond to bi =1 and ci =0.

5) Non-collusion Probabilistically

Misbehave(NCPM) Malicious components always act independently and give incorrect results probabilistically with probability less than 1. Different malicious components may have different misbehaving probability bi . It corresponds to 0< bi <1 and ci =0.

6) Full time Full Collusion(FTFC)

Malicious component always collude and always give the same incorrect results,corresponding to bi =1 and ci=1.

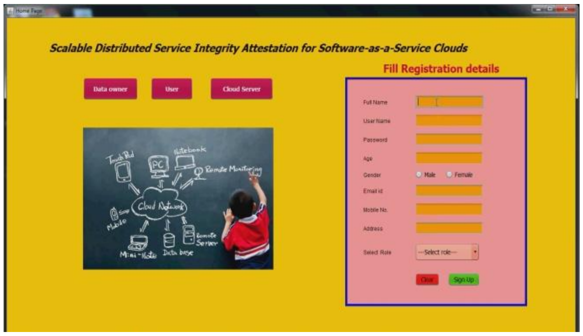
7) Partial Time Full Collusion(PTFC)

Malicious components always collude and give the same incorrect results on selected tuples,corresponding to 0 < bi < 1 and ci =1.

8) Partial Time Partial Collusion

Malicious component sometimes collude and sometimes act independently. It corresponds to 0< bi<1 and 0 < ci<1.

IV. SIMULATION RESULTS

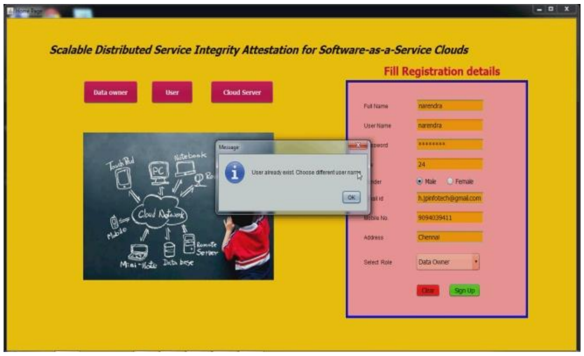


Scalable Distributed Service Integrity Attestation for Software-as-a-Service Clouds

Fill Registration details

Full Name
User Name
Password
Age
Gender ○ Male ○ Female
Email id
Mobile No.
Address
Select Role ---Select role---

Fig. 3



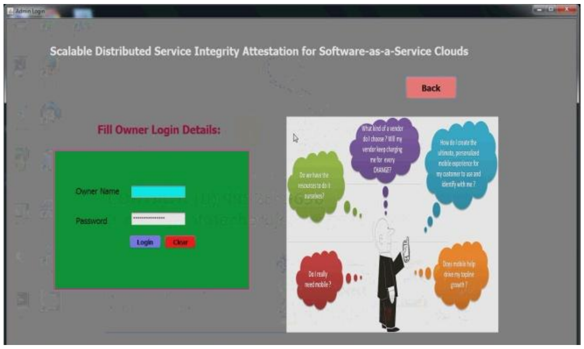
Scalable Distributed Service Integrity Attestation for Software-as-a-Service Clouds

Fill Registration details

Full Name: narendra
User Name: narendra
Password: ********
Age: 24
Gender: ● Male ○ Female
Mail id: h.jpinfotech@gmail.com
Mobile No: 9094039411
Address: Chennai
Select Role: Data Owner

Message: User already exist. Choose different user name

Fig. 4



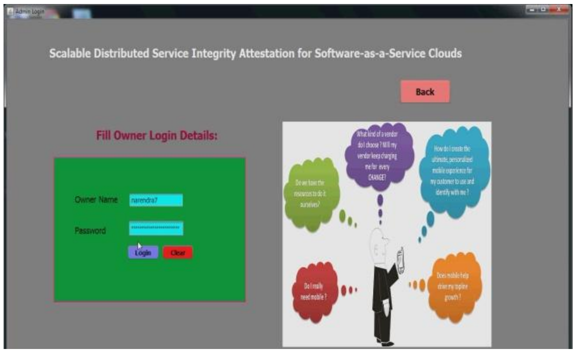
Scalable Distributed Service Integrity Attestation for Software-as-a-Service Clouds

Fill Owner Login Details:

Owner Name
Password

Back
Login
Clear
What kind of a vendor do I choose? Will my vendor keep charging me for every CHANGE?
Do we have the resources to do it ourselves?
How do I create the ultimate, personalized mobile experience for my customer to use and identify with me?
Do I really need mobile?
Does mobile help drive my topline growth?

Fig. 5



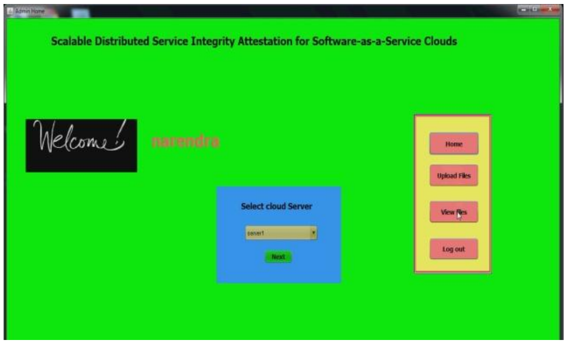
Scalable Distributed Service Integrity Attestation for Software-as-a-Service Clouds

Fill Owner Login Details:

Owner Name: narendra7
Password: ***********

Back
Login
Clear
What kind of a vendor do I choose? Will my vendor keep charging me for every CHANGE?
Do we have the resources to do it ourselves?
How do I create the ultimate, personalized mobile experience for my customer to use and identify with me?
Do I really need mobile?
Does mobile help drive my topline growth?

Fig.6

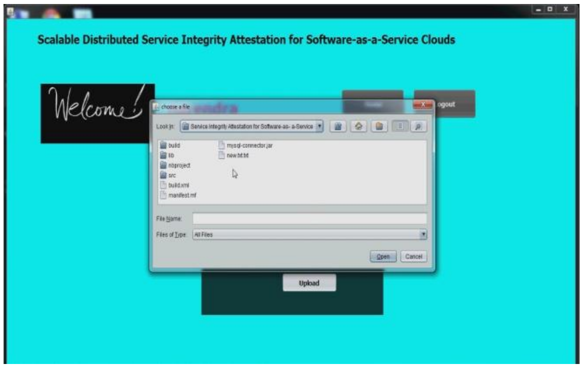


Scalable Distributed Service Integrity Attestation for Software-as-a-Service Clouds

Welcome! narendra

Select cloud Server: server1

Fig. 7



Scalable Distributed Service Integrity Attestation for Software-as-a-Service Clouds

Welcome!

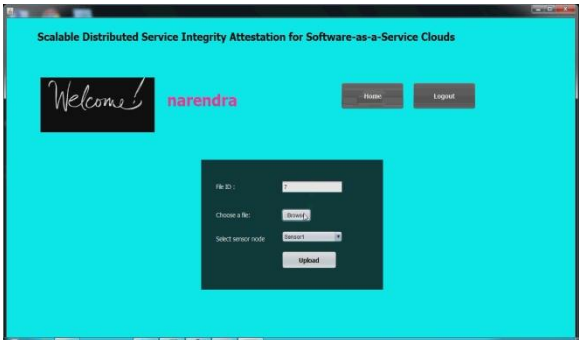
choose a file

Look In: Service Integrity Attestation for Software-as- a-Service

build
lib
nbproject
src
build.xml
manifest.mf
mysql-connector.jar
new.txt.txt

File Name:
Files of Type: All Files

Fig. 8



Scalable Distributed Service Integrity Attestation for Software-as-a-Service Clouds

Welcome! narendra

File ID : 7
Choose a file: Browse
Select sensor node: Sensor1

Fig. 9

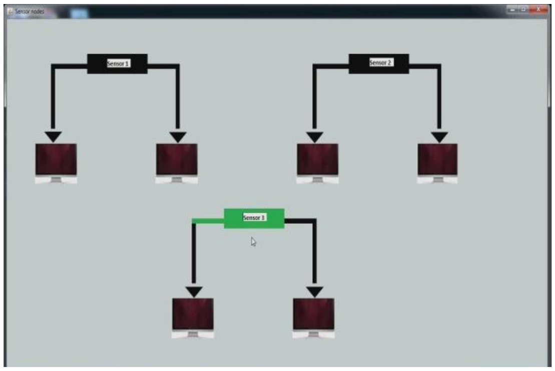
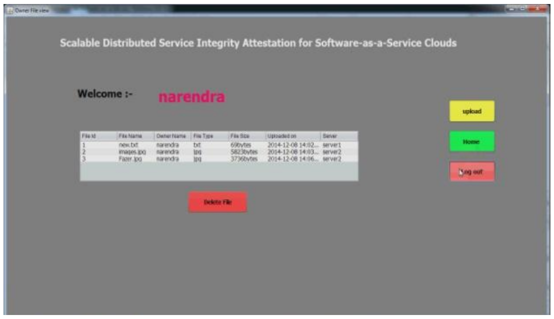


Fig. 10

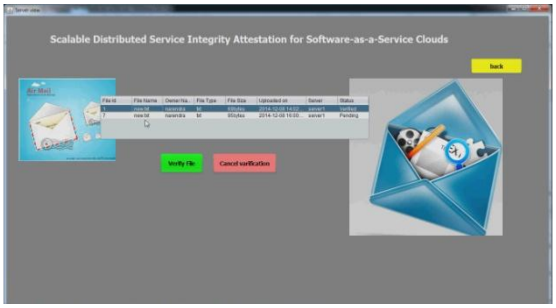


Scalable Distributed Service Integrity Attestation for Software-as-a-Service Clouds

Welcome :- narendra

File Id	File Name	Owner Name	File Type	File Size	Uploaded on	Server
1	new.txt	narendra	txt	69bytes	2014-12-08 14:02...	server1
2	images.jpg	narendra	jpg	5823bytes	2014-12-08 14:03...	server2
3	Fazer.jpg	narendra	jpg	3736bytes	2014-12-08 14:06...	server2

Fig.11


Server view
Scalable Distributed Service Integrity Attestation for Software-as-a-Service Clouds
back
Air Mail
Verify File
Cancel varification

File Id	File Name	Owner Na...	File Type	File Size	Uploaded on	Server	Status
1	new.txt	narendra	txt	69bytes	2014-12-08 14:02...	server1	Verified
7	new.txt	narendra	txt	95bytes	2014-12-08 16:00...	server1	Pending

Fig.12

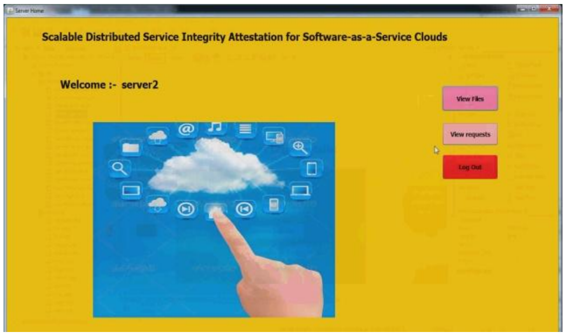
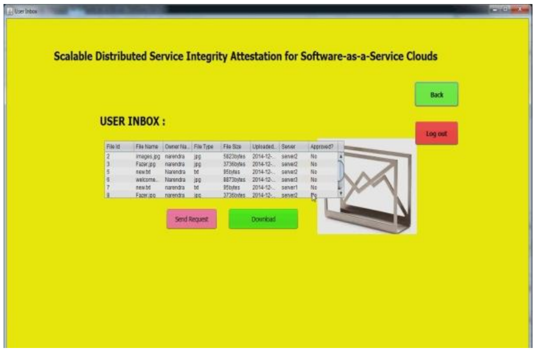
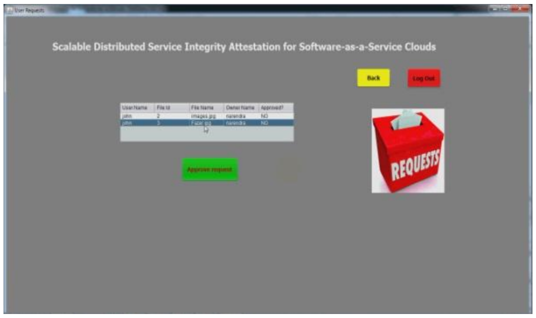

Server Home
Scalable Distributed Service Integrity Attestation for Software-as-a-Service Clouds
Welcome :- server2
View Files
View requests
Log Out

Fig.13


User Inbox
Scalable Distributed Service Integrity Attestation for Software-as-a-Service Clouds
Back
USER INBOX :
Log out
Send Request
Download

File Id	File Name	Owner Na...	File Type	File Size	Uploaded...	Server	Approved?
2	images.jpg	narendra	jpg	5823bytes	2014-12-...	server2	No
3	Fazer.jpg	narendra	jpg	3736bytes	2014-12-...	server2	No
5	new.txt	Narendra	txt	95bytes	2014-12-...	server2	No
6	welcome...	Narendra	jpg	8873bytes	2014-12-...	server3	No
7	new.txt	narendra	txt	95bytes	2014-12-...	server1	No
8	Fazer.jpg	narendra	jpg	3736bytes	2014-12-...	server2	No

Fig.14


User Requests
Scalable Distributed Service Integrity Attestation for Software-as-a-Service Clouds
Back
Log Out
Approve request
REQUESTS

User Name	File Id	File Name	Owner Name	Approved?
john	2	images.jpg	narendra	NO
john	3	Fazer.jpg	narendra	NO

Fig.15

V. CONCLUSION

In this paper we introduced a novel integrated service integrity attestation graph analysis scheme for multitenant software-as-a-service cloud system. IntTest uses a reply based consistency check to verify the service providers. IntTest will analyses both the consistency and inconsistency graphs to find the malicious attackers efficiently than anyother existing techniques. And also it will provide a result auto correction to improve the result quality.

REFERENCES

[1] Garay.J and Huelsbergen.L, "Software integrity protection using timed executable agents," in Proceedings of ACM Symposium on Information, Computer and Communications Security (ASIACCS), Taiwan, Mar. 2006.

[2] Juan Du Daniel J. Dean, Yongmin Tan, Xiaohui Gu, Senior and Ting Yu Scalable Distributed Service Integrity Attestation for Softwareas-a-Service Clouds .

[3] Du.J, Wei.W, Gu.X, and Yu.T, "Runtest: Assuring Integrity of Dataflow Processing in Cloud Computing Infrastructures," Proc.ACM Symp. Information, Computer and Comm. Security (ASIACCS),2010.

[4] Du.J, Shah.N, and Gu.X, "Adaptive Data-Driven Service Integrity Attestation for Multi-Tenant Cloud Systems," Proc. Int'l Workshop Quality of Service (IWQoS), 2011. Virtual Computing Lab, http://vcl.ncsu.edu/, 2013.

[5] Ho et al.T, "Byzantine Modification Detection in Multicast Networks Using Randomized Network Coding," Proc. IEEE Int'l Symp. Information Theory (ISIT), 2004.

[6] Hwang.I, "A Survey of Fault Detection, Isolation, and Reconfiguration Methods," IEEE Trans. Control System Technology, vol. 18,no. 3, pp. 636-653, May 2010.

[7] Lamport.L, Shostak.R, and Pease.M, "The Byzantine Generals Problem," ACM Trans. Programming Languages and Systems, vol. 4,no. 3, pp. 382-401, 1982

[8] Shi.E, Perrig.A, and Doorn.L.V, "Bind: A fine-grained attestation service for secure distributed systems," in Proceedings of the IEEE Symposium on Security and Privacy, 2005.

[9] Xu.W, Venkatakrishnan.V. N, Sekar.R, and Ramakrishnan .I. V, "A framework for building privacy-conscious composite web services,"in IEEE International Conference on Web Services, Chicago, IL, Sep. 2006, pp. 655–662.

[10] Zhang.H, Savoie.M,Campbell.S, Figuerola.S, von Bochmann.G, and Arnaud.B.S, "Service-oriented virtual private networks for grid applications," in

IEEE International Conference on Web Services, Salt Lake City, UT, Jul. 2007, pp. 944–951.

BIODATA

Author

Poola Ravisekhar presently pursuing his M.Tech in CSE, Kottam College of Engineering, Chinnatekuru (V),Kurnool,Andhra Pradesh, India.

Co-Author

V.Sujatha received M.Tech. Presently working as Assistant Professor in Kottam College of Engineering, Chinnatekuru (V),Kurnool,Andhra Pradesh, India.