Volume No:1, Issue No:6 (November-2015) ISSN No : 2454-423X (Online)

International Journal of Research in Advanced Computer Science Engineering

A Peer Reviewed Open Access International Journal

www.ijracse.com

Security and Sharing Resource Integrity Declaration in Cloud Computing Using SaaS

E.S.Mahesh Kumar
M. Tech Student,
Department of CSE,
Dr.Samuel George Institute of Engineering & Technology, Darimadugu, Markapuram, Andrapradesh, India.

D. Kumar
Associate Professor,
Department of CSE,
Dr.Samuel George Institute of Engineering & Technology, Darimadugu, Markapuram, Andrapradesh, India.

Abstract:

SaaS provides a flexible environment through which application service providers allowed to host their applications in a distributed environment, so that users can access the hosted services in a easier way Web services and service-oriented architecture (SOA) mature and new developmental approaches, such as Ajax, become popular. SaaS clouds are vulnerable to malicious attacks because of their sharing nature we present Function Combination Generator along with the IntTest in order to detect the attackers more in number. Also, Result Auto Correction is provided to correct the incorrect results provided by the attackers Int Test does not require any special hardware or secure kernel support and imposes little performance impact to the application,

which makes it practical for large-scale cloud systems the proposed system will limit the attack scope using integrity attestation on the service components by doing this it will be difficult to attack the popular service functions. By doing this the attackers can be easily pinpointed. Using integrity attestation on service components will improve privacy and the computation time will be reduced to greater extent. The proposed system will also provide result auto correction to automatically correct compromised results to improve the result quality.

Index terms:

Distributed Service Integrity Attestation, Cloud Computing, SaaS, PaaS, IaaS, VMs and ASP

I.INTRODUCTION:

Service-oriented computing is a popular paradigm for implementing and designing distributed systems. Companies governments and universities have developed grid, cloud and web services to provide access to data and for performing resource-intensive computation [1][3] There are many advantages over previous ad-hoc systems. Services can scale to match demand, charge on a per-use basis and provide backup and redundancy[2]Furthermore web service interfaces are described using open interoperable standards allowing them to be composed together so that complex systems can be built from many individual services This can even be automated allowing for rapid system development Cloud computing is the lease of the resources through which the users can use the resources depending upon the requirement and pay based on the usage. Trough cloud computing the user can decrease the cost and can use the resource at any time [5].

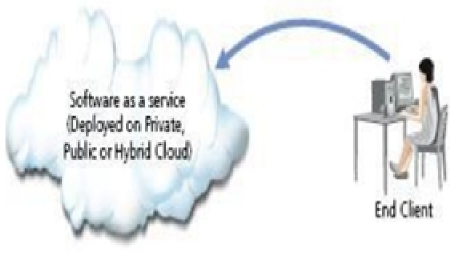

Software as a service (Deployed on Private, Public or Hybrid Cloud)
End Client

Figure 1: Software-as-a Service

Users could benefit from knowing to what extent a cloud provider delivers the promised service. More concretely, the contract between the cloud and its users should be verifiable [4] and the ability to detect failures, without relying solely on the cloud provider's report, can be useful to the users.

Volume No:1, Issue No:6 (November-2015) ISSN No : 2454-423X (Online)

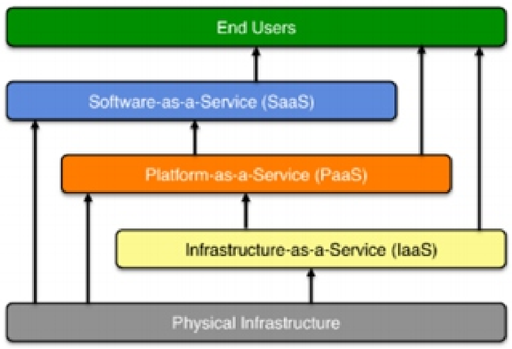
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

For example, it may be important to promptly find out that a service does not respect its functional specification; or that it generously shares personal data with the world; or that it is down, underperforms [6] its basic security controls seem to be failing. This information can be especially helpful for critical applications such as medicine or banking and facilitate their process of adopting cloud technologies. In addition verification tools to check these aspects can help consumers pick and choose a particular cloud provider Function Combination Generator are provided for the IntTest to overcome the limitation[7] Function Combination Generator along with IntTest can attain more attacker identifying accuracy than existing schemes like Run Test and AdapTest. In particular, AdapTest and RunTest with the other conventional voting schemes wants to believe that benevolent service providers take bulk in every service function.

II.RELATED WORK:

SaaS clouds are given with various integrity attestation schemes in recent years. The BIND scheme, TEAS, RunTest and AdapTest are some of the schemes, but these in turn have some issues that are to be dealt with. Some of them want trusted hardware and support of secure kernel. BIND [10] (Binding Information and Data) is one that requires secure kernel or a third party support. To verify the service integrity for SaaS clouds, BIND exhibits the fine grained attestation framework. This BIND scheme[8] Attestation annotation mechanism Sandbox mechanism Verification of authenticator through hash. In order to address the service integrity attestation, Diffee-Hellman key has been used by the BIND scheme It develops upon the concepts of Software as a Service (SaaS) and Service Oriented Architecture (SOA) which allows application service providers (ASPs) to deliver their applications via large-scale cloud computing infrastructures. Amazon Web Service (AWS) and Google App Engine are examples to provide a set of application services supporting enterprise applications and big data processing. A distributed application service can be dynamically composed from individual service components provided by different ASPs[9] a disaster assistance claim processing application consists of voice-over-IP (VoIP) analysis component, email analysis component, community discovery This paper [4] propose Robust Service Integrity Attestation (ROSIA) framework.

This can efficiently verify the integrity of stateful dataflow processing services and pinpoint malicious service within a large-scale cloud ystem. ROSIA support stateful dataflow Services and hence achieves robustness. ROSIA performs integrity attestation by examining both consistency and inconsistency relationships. This frame work attains higher attack detection accuracy and also limits the scope of the damage caused by colluding attackersPlatform-as-a-Service (PaaS) sits on top of the physical infrastructure or IaaS. Similarly to IaaS, PaaS incorporates services for computing and storing data[11]these services are offered at a higher level of abstraction (e.g., databases, runtime and web app hosting) and are supported by a richer set of auxiliary services (e.g., message handling). Examples of PaaS services include Google AppEngine [10] and Microsoft Azure [12] PaaS services are typically implemented by middleware components that operate on top of the operating system and include execution runtimes frameworks, and database servers.


End Users
Software-as-a-Service (SaaS)
Platform-as-a-Service (PaaS)
Infrastructure-as-a-Service (IaaS)
Physical Infrastructure

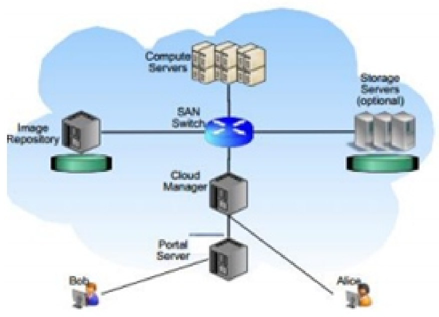
III.EXISTING SYSTEM:

Which enable application service providers (ASPs) to deliver their applications via the massive cloud computing infrastructure. In particular, our work focuses on data stream processing services that are considered to be one class of killer applications for clouds with many real-world applications in security surveillance, scientific computing, and business intelligence. However, cloud computing infrastructures are often shared by ASPs from different security domains, which make them vulnerable to malicious attacks [13] attackers can pretend to be legitimate service providers to provide fake service components, and the service components provided by benign service providers may include security holes that can be exploited by attackers ASAN is a specialized high speed

Volume No:1, Issue No:6 (November-2015) ISSN No : 2454-423X (Online)

International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

network of storage devices and switches connected to computer systems. The users will make the request and the cloud manager will authenticate the user and he keep track of the users and their request and due to the streaming techniques and AMoV will adjust the streaming flow with a video coding technique will adjust the flow and increase the quality. ESov monitors the social network interactions[14].



IV.PROPOSED SYSTEM:

Software as a service and service oriented architecture are the basic concepts of SaaS clouds and this will allow the application service provider to deliver their application via cloud computing infrastructure. In our proposed method we are introducing a new concept called IntTest. The main goal of IntTest is, it can pinpoint all the malicious service providers. IntTest will treat all the service providers as black boxes and this does not need any special hardware or secure kernel support[5] When we are considering the large scale cloud system multiple service providers may simultaneously compromised by a single malicious attacker. In this we assume that the malicious nodes are not having any knowledge about the other nodes except those which they are directly interacting [8].

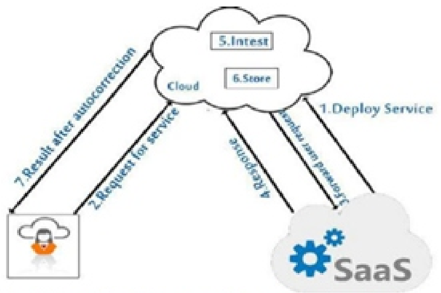


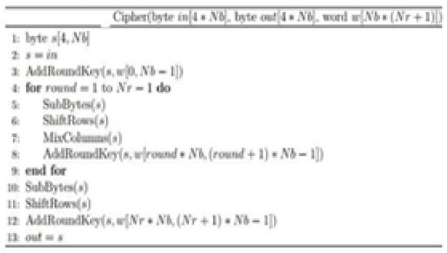
Fig.2. Over all architecture of the proposed method.

The overall architecture of the proposed system In this, first the services are deployed in the cloud then the user sends the request to the cloud . After that 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.Finally, the result will be send to the user by the cloud. The architecture shows this IntTest module in detail[3].

A.Encryption:

For encryption this paper uses AES (Advanced Encryption Standard), the cipher takes a plaintext and a key as input and outputs a ciphertext. The plaintext is represented as a byte matrix with 4 rows and 4 columns. The intermediate cipher result is called the state. After an initial round key addition, the state is transformed by implementing a round function 10, 12, or 14 times for 128- bit, 192-bit or 256-bit keys, respectively. Each round function, except the final round, contains four transformations which are SubBytes (SB), ShiftRows (SR), MixColumns (MC) and AddRoundKey[15] (ARK). The final round is slightly different from the first Nr − 1 rounds as it does not include the MixColumns operation. The encryption process is described in pseudo code in Procedure.

Figure 4: Encryption Procedure

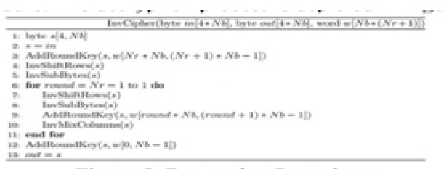


```
Cipher(byte $in[4*Nb]$, byte $out[4*Nb]$, word $w[Nb*(Nr+1)]$)
byte $s[4, Nb]$
$s = in$
AddRoundKey($s, w[0, Nb-1]$)
for $round = 1$ to $Nr-1$ do
    SubBytes($s$)
    ShiftRows($s$)
    MixColumns($s$)
    AddRoundKey($s, w[round*Nb, (round+1)*Nb-1]$)
end for
SubBytes($s$)
ShiftRows($s$)
AddRoundKey($s, w[Nr*Nb, (Nr+1)*Nb-1]$)
$out = s$
```

B.Decryption:

For decryption, the cipher takes a ciphertext and a key as two input parameters and outputs the corresponding plaintext. The four transformations SubBytes ShiftRows, MixColumns and AddRoundKey, can be inverted in reverse order to provide the decryption of the cipher.

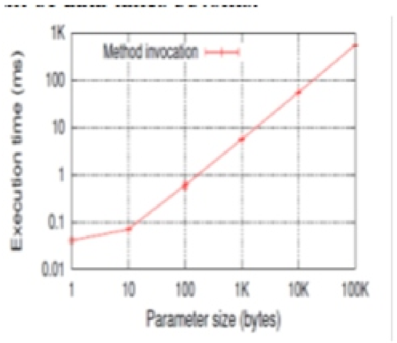
The decryption algorithm is expressed in pseudo code in Procedure. The decryption process is depicted [16] Decryption Procedure The inverse operations of SubBytes, ShiftRows and MixColumns are represented as InvShiftRows, InvSubBytes and InvMixColumns, respectively. Note that the inverse function of AddRoundKey is itself.



```
InvCipher(byte in[4 * Nb], byte out[4 * Nb], word w[Nb * (Nr + 1)])
byte s[4, Nb]
s = in
AddRoundKey(s, w[Nr * Nb, (Nr + 1) * Nb − 1])
InvShiftRows(s)
InvSubBytes(s)
for round = Nr − 1 to 1 do
    InvShiftRows(s)
    InvSubBytes(s)
    AddRoundKey(s, w[round * Nb, (round + 1) * Nb − 1])
    InvMixColumns(s)
end for
AddRoundKey(s, w[0, Nb − 1])
out = s
```

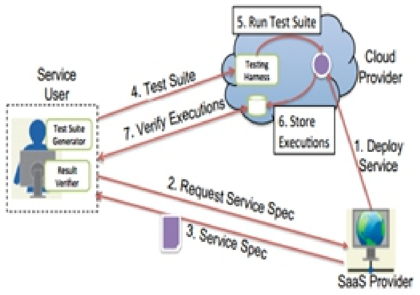
Figure 5: Decryption Procedure

Some light on the performance impact of cryptographic operations in the attested primitives, our evaluation results for seal and unseal as I vary the size of the data to be sealed and the size of the envelope to be unsealed. Because the Attestation makes use of the AES to implement cryptographic operations in native code, seal and unseal are efficient. Sealing 1KB takes 5.3ms and unsealing the same amount of data takes 33.6ms [17].



1.VERIFYING FUNCTIONAL PROPERTIES OF CLOUD SERVICES :

The techniques described in the previous section allow the PaaS services to be associated with a strong identity, which is being preserved throughout the entire software lifetime withstanding administration mistakes, and tampering attempts we focus on a complementary question, namely, given a uniquely identified service instance deployed and running on the trusted PaaS platform, how can we efficiently verify that its behavior complies with the functional properties advertised by its provider[18] Our approach to verifying functional properties of the PaaS services is based on the software testing paradigm. Conceptually, the software testing process can be viewed as consisting



2.Integrity Attestation Scheme:

IntTest is mainly used to detect the service integrity attack in SaaS clouds and also pinpoint malicious service providers. Integrity Attestation Scheme is the main basis of IntTest. Integrity Attestation Scheme in turn uses the replay based consistency check to identify the attackers. In Cloud Computing, several providers develop the same function as they are popular[19] Function Combination Generator after generating patterns sends the results to the IntTest. IntTest then obtains the consistency and inconsistency relationships among the different service providers for a particular set of service function generated.

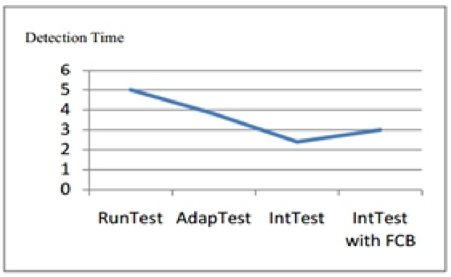
V.RESULT ANALYSIS:

Considering the parameters like integrity, server utilization, extendibility ,overhead and vulnerabilities we could find that Trust virtual data center and Placement and Extraction method has high server utilization [20],low overhead but they have denial of service, malicious service provides can still escape.Stateful data processing method seems to have low overhead and scalability but malicious providers can still escape while Privacy proxy can provide security to user data it cannot avoid colluding attacks. Privacy conscious composite web services has automated techniques for checking models at the consumer site for compliance of consumer privacy policies but still they cannot address malicious service providers that intentionally lie about their usage of consumer data, also have low server utilization [21].

Schemes	Detection Rate	Detection Time
1. RunTest	Low than IntTest	Higher than AdapTest
2. AdapTest	Low than IntTest	40% lesser than RunTest
3.IntTest with no Function Combination Generator	High than RunTest and AdapTest but not 100%	Less than RunTest and AdapTest
4.IntTest with Function Combination Generator	Higher than all the three and 100% effective	High than IntTest

Table: Comparison Results

The table that compares parameters like detection rate and the detection time among various approaches like AdapTest, RunTest, and IntTest with no Function Combination Generator and IntTest with Function Combination Generator



VI.CONCLUSIONS:

In this paper a wide survey of the different frameworks for providing security to SaaS has been carried out and pointed out their advantages and drawbacks. We need to further improve those frameworks or develop some efficient 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 we conclude that IntTest is light-weight, which assess an impact to the data processing processing services which runs inside the cloud computing infrastructure Function Combination Generator analyses both the consistency and inconsistency graphs to find the malicious attackers efficiently than any other existing techniques. And also it provides a result auto correction to improve result quality in the future. We hope that our paper will encourage future research in this area

VII.FURTHER WORK:

Furthermore IntTest provides result autocorrection to automatically correct compromised results to improve the result quality. Also implemented IntTest and tested it on a commercial data stream processing platform running inside a production virtualized cloud computing infrastructure consistency and inconsistency graphs to find the malicious attackers efficiently than any other existing techniques. And also it will provide a result auto correction to improve the result quality.

REFERENCES:

[1] Amazon Web Services, http://aws.amazon.com/, 2013.

[2] Amazon Web Services, http://aws.amazon.com/, 2013.

[3] Google App Engine, http://code.google.com/appengine/, 2013.

[4]Software as a Service http://en.wikipedia.org/wiki/Software as a Service, 2013.

[5] G. Alonso, F. Casati, H. Kuno, and V. Machiraju, Web Services Concepts, Architectures and Applications (Data-Centric Systems and Applications). Addison-Wesley Professional, 2002

[6] ¨UlleMadise and TarviMartens. E-voting in Estonia 2005 pages 15–26, Castle Hofen, Bregenz, Austria, August 2006.

[7] M.R. Clarkson, S. Chong, and A.C. Myers. Civitas: Toward a Secure Voting System. In S&P "08: Proceedings of the IEEE Symposium on Security and Privacy, pages 354–368. IEEE, May 2008.

[8] B. Aggarwal, N. Spring, and A. Schulman, "Stratus : EnergyEfficient Mobile Communication using Cloud Support," in ACM SIGCOMM DEMO, 2010.

[9] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A Survey of Mobile Cloud Computing : Architecture , Applications , and Approaches," in Wiley Journal of Wireless Communications and Mobile Computing, Oct. 2011.

[10] Gentry Craig. "A fully homomorphic encryption scheme".Ph.D. thesis, Stanford University; 2009. [retrieved 21.04.11]

[11] W. Xu, V.N. Venkatakrishnan, R. Sekar, and I.V. Ramakrishnan, "A Framework for Building Privacy-Conscious Composite Web Services," Proc. IEEE Int'l Conf. Web Services, pp. 655-662, Sept. 2006.

[12] P.C.K. Hung, E. Ferrari, and B. Carminati, "Towards Standardized Web Services Privacy Technologies," IEEE Int'l Conf. Web Services, pp. 174-183, June 2004.

[13] L. Alchaal, V. Roca, and M. Habert, "Managing and Securing Web Services with VPNs," Proc. IEEE Int'l Conf. Web Services, pp. 236- 243, June 2004

[14] S. A. Baset. Cloud SLAs: present and future. SIGOPS Oper. Syst. Rev., 46(2):57–66, July 2012.

[15] A. Bessani, M. Correia, B. Quaresma, F. Andr´e, and P. Sousa. DepSky: Dependable and secure storage in a cloud-of-clouds. In Proc. 6th European Conference on Computer Systems (EuroSys), pages 31–46, 2011.

[16] S. Bleikertz, A. Kurmus, Z. A. Nagy, and M. Schunter. Secure cloud maintenance – protecting workloads against insider attacks.

[17] T.S. Group, “STREAM: The Stanford Stream Data Manager,” IEEE Data Eng. Bull., vol. 26, no. 1, pp. 19-26, Mar. 2003.

[18] D.J. Abadi et al., “The Design of the Borealis Stream Processing Engine,” Proc. Second Biennial Conf. Innovative Data Systems Research (CIDR ’05), 2005.

[19] B. Gedik et al., “SPADE: The System S Declarative Stream Processing Engine,” Proc. ACM SIGMOD Int’l Conf. Management of Data (SIGMOD ’08), Apr. 2008.

[20] S. Berger et al., “TVDc: Managing Security in the Trusted Virtual Datacenter,” ACM SIGOPS Operating Systems Rev., vol. 42, no. 1, pp. 40-47, 2008.

[21] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, “Hey, YouGet Off My Cloud! Exploring Information Leakage in Third-Party Compute Clouds

Author’s Details:

E.S. Mahesh Kumar, he has received B. Tech (CSE) Degree in 2013 from JNTU Kakinada University, Kakinada. Currently, he is pursuing his M. Tech in Dr. Samuel George Institute of Engineering & Technology, Markapuram, and Prakasam Dist. Andhra Pradesh. His research interested areas are cloud computing, networking and data mining.

D. Kumar received B.Tech (CSIT) Degree from JNT University in 2006 and M.Tech (CSE) Degree from JNTUK Kakinada in 2011. He has 09 years of teaching experience. He joined as Assistant Professor in Dr. Samuel George Institute of Engineering & Technology, Markapuram, India in 2006. Presently he is working as Associate Professor in CSE Dept. His Interested research areas are Image Processing and Computer Networks. He attended Various National and International Workshops and Conferences.