International Journal of Science Engineering and Advance Technology

Integrated Attestation Scheme for Scalable Distributed Services in Software-as-a-Service Clouds

Pampana Satya Venkata Gowtham Raju[1], Md Imran[2]
[1]M.Tech (CSE), Nimra Institute of Science and Technology, A.P., India.
[2]Asst. Professor, Dept. of Computer Science & Engineering, Nimra Institute of Science and Technology, A.P., India.

Abstract *— Ensuring Security, Data Integrity and Seamless Availability for information very still, in movement, and being used, for Outsourcing delicate and vital information in the hands of a cloud supplier is essential assignment. Among the 3 sorts of administrations IaaS, PaaS and SaaS gave by the cloud, Software as a Service (SaaS) is a product dispersion model in which applications are facilitated by a seller or administration supplier and made accessible to clients over a system, ordinarily the Interne. In any case, because of their sharing nature, SaaS mists are helpless against vindictive assaults SaaS cloud frameworks empower application administration suppliers to convey their applications by means of huge distributed computing bases.. In this paper, we introduce IntTest, an adaptable and powerful administration uprightness authentication structure for SaaS mists. IntTest gives a novel incorporated validation chart examination plot that can give more grounded aggressor pinpointing power than past plans. In addition, IntTest can consequently upgrade result quality by supplanting terrible results delivered by vindictive assailants with great results created by benevolent administration suppliers. We have executed a model of the IntTest framework and tried it on a creation distributed computing base utilizing IBM System S stream handling applications. Our trial results demonstrate that IntTest can accomplish higher assailant pinpointing precision than existing methodologies. We show IntTest, another coordinated administration trustworthiness confirmation structure for multitenant cloud frameworks. IntTest gives a reasonable administration trustworthiness authentication plot that does not expect trusted substances on outsider administration provisioning destinations or require application modificationsIntTest does not require any uncommon equipment or secure part bolster and forces little execution effect to the application, which makes it down to earth for vast scale cloud frameworks.*

Keywords — Distributed service integrity attestation, cloud computing, secure distributed data processing

I. INTRODUCTION

Cloud offers special points of interest in versatility and cost-sparing. Thus, facilitating information serious question administrations in the cloud turn out to be progressively mainstream. Administration proprietors can advantageously scale up or down the administration and pay for the hours of utilizing the servers with the cloud bases. You can essentially run your organization's IT operations with only a program and an Internet association .Cloud processing has risen as a practical asset renting worldview, which deters the requirement for clients keep up complex physical figuring bases independent from anyone else. Programming as-an administration (SaaS) mists (e.g., Amazon Web Service (AWS) [1] and Google AppEngine [2]) expand upon the ideas of programming as an administration [3] and administration arranged structural engineering (SOA) [4], [5], which empower application administration suppliers (ASPs) to convey their applications by means of the gigantic distributed computing framework. Specifically, our work concentrates on information stream handling administrations [6], [7], [8] that are thought to be one class of executioner applications for mists with some true applications in security observation, investigative registering, and business insight. Be that as it may, distributed computing foundations are frequently shared by ASPs from diverse security areas, which make them helpless against malignant assaults [9], [10]. For instance, aggressors can put on a show to be real administration suppliers to give fake administration parts, and the administration segments gave by favorable administration suppliers might incorporate security gaps that can be misused by assailants. Our work concentrates on administration respectability assaults that cause the client to get untruthful information preparing results, showed by Fig. 1. Despite the fact that classification and security insurance issues have been broadly contemplated by past examination [11], [12], [13], [14], [15], [16], the administration trustworthiness confirmation issue has not been appropriately tended to. Besides, benefit honesty is the most predominant issue, which should tended to regardless of whether open or private information are handled by the cloud framework. Albeit past work has given different programming respectability confirmation arrangements [9], [10], [11],[12], those systems

frequently require extraordinary trusted equipment or secure portion support, which makes them hard to be sent on vast scale distributed computing frameworks. Conventional Byzantine adaptation to non-critical failure (BFT) procedures [14], [15] can distinguish discretionary mischievous activities utilizing full-time greater part voting (FTMV) over all imitations, which however bring about high overhead to the cloud framework.

In this paper, we show IntTest, another coordinated administration respectability confirmation structure for multitenant cloud frameworks. IntTest gives a handy administration honesty confirmation plot that does not expect trusted substances on outsider administration provisioning locales or require application adjustments. IntTest expands upon our past work RunTest [16] and AdapTest [7] however can give more grounded vindictive aggressor pinpointing power than RunTest and AdapTest. In particular, RunText and AdapTest and also conventional greater part voting plans need to accept that considerate administration suppliers take lion's share in each administration capacity. On the other hand, in huge scale multitenant cloud frameworks, numerous noxious aggressors might dispatch plotting assaults on certain focused on administration capacities to discredit the supposition. To address the test, IntTest takes aholistic approach by deliberately inspecting both consistency and irregularity connections among distinctive administration suppliers inside of the whole cloud
framework.

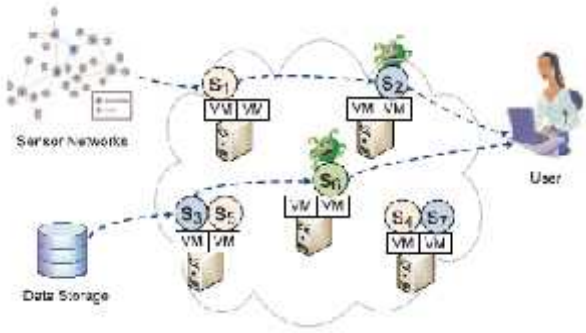
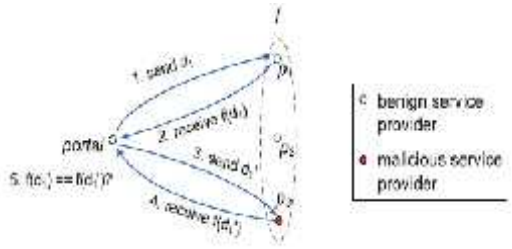


Fig. 1. **Service integrity attack in cloud-based data processing. Si**
denotes different service component and VM denotes virtual machines

The per-capacity consistency raph examination can restrict the extent of harm brought on by plotting assailants, while the worldwide irregularity diagram investigation can adequately uncover those aggressors that attempt to bargain numerous administration capacities. Consequently, IntTest can at present pinpoint malignant aggressors regardless of the fact that they get to be lion's share for some administration capacities. By taking a coordinated methodology, IntTest can pinpoint assailants all the more proficiently as well as can smother forceful aggressors and limit the extent of the harm brought about by plotting assaults. Besides, IntTest gives result auto adjustment that can consequently supplant undermined information handling results delivered by vindictive aggressors with great results created by considerate administration suppliers. In particular, this paper makes the accompanying commitments:

- We provide a scalable and efficient distributed service integrity attestation framework for large scale cloud computing infrastructures.

- We present a novel integrated service integrity attestation scheme that can achieve higher pinpointing accuracy than previous techniques.



- We describe a result auto correction technique that can automatically correct the corrupted results produced by malicious attackers.

- We conduct both analytical study and experimental evaluation to quantify the accuracy and overhead of the integrated service integrity attestation scheme.

We have executed a model of the IntTest framework and tried it on NCSU's virtual figuring lab (VCL) [8], a generation distributed computing foundation that works similarly as the Amazon flexible process cloud (EC2) [9]. The benchmark applications we use to assess IntTest are appropriated information stream handling administrations gave by the IBM System S stream preparing stage [8], [3], an industry quality information stream preparing framework. Exploratory results demonstrate that IntTest can accomplish more exact pinpointing than existing plans (e.g., RunTest, AdapTest, and full-time larger part voting) under deliberately conspiring assaults. IntTest is adaptable and can diminish the authentication overhead by more than one request of greatness contrasted with thetraditional full-time greater part voting plan.

II. PROBLEM STATEMENT

Given a SaaS cloud system, the goal of IntTest is to pinpoint any malicious service provider that offers an untruthful service function. IntTest treats all service components as black boxes, which does not require any special hardware or secure kernel support on the cloud platform. We now describe our attack model and our key assumptions as follows:

Attack model. A malicious attacker can pretend to be a legitimate service provider or take control of vulnerable service providers to provide untruthful service functions. Malicious attackers can be stealthy, which means they can misbehave on a selective subset of input data or service functions while pretending to be benign service providers on other input data or functions. The stealthy behavior makes detection more challenging due to the following reasons:

- The detection scheme needs to be hidden from the attackers to prevent attackers from gaining knowledge on the set of data processing results that will be verified and therefore easily escaping detection; and

- The detection scheme needs to be scalable while being able to capture misbehavior that may be both unpredictable and occasional.

In a vast scale cloud framework, we have to consider conspiring assault situations where numerous malevolent aggressors connive or different administration locales are at the same time traded off and controlled by a solitary noxious assailant. Aggressors could sporadically plot, which implies an assailant can intrigue with a subjective subset of its colluders whenever. We accept that malignant hubs have no information of different hubs aside from those they connect with straightforwardly. In any case, assailants can correspond with their colluders in a discretionary way. Assailants can likewise change their assaulting and plotting systems self-assertively. Suspicions we first accept that the aggregate number of noxious administration parts is not exactly the aggregate number of kindhearted ones in the whole cloud framework. Without this supposition, it would be hard, if not absolutely unthinkable, for any assault location plan to work when tantamount ground truth handling results are not accessible.

Fig.2. Replay-based consistency check.

Be that as it may, not quite the same as RunTest, AdapTest, or any past lion's share voting plans, IntTest does not expect considerate administration parts must be the dominant part for each administration capacity, which will significantly improve our pinpointing control and restrict the extent of administration capacities that can be traded off by noxious assailants. Second, we expect that the information handling administrations are data deterministic, that is, given the same information, a generous administration segment dependably creates the same or comparable yield (in light of a client characterized closeness capacity). Numerous information stream handling capacities fall into this classification [8]. We can likewise effortlessly extend our validation system to bolster stateful information handling administrations [8], which however is outside the extent of this paper. Third, we likewise expect that the outcome irregularity created by equipment or programming issues can be stamped by deficiency location plans [3] and are rejected from our malignant assault recognition.

III. RELATED WORK

To identify administration respectability assault and pinpoint noxious administration suppliers, our calculation depends on replay-based consistency check to infer the consistency/irregularity connections between administration suppliers. For instance, Fig. 2 demonstrates the consistency check plan for verifying three administration supplier's p1, p 2, and p 3 that offer the same administration capacity f. The entry sends the first information d1 to p1 and gets back the outcome f(d1). Next, the entry sends d0, a copy of d1 to p3 and gets back the outcome f(d0).

The gateway then thinks about f(d1) and f(d0) to see whether p1 and p3 are reliable. The instinct behind our methodology is that if two administration suppliers can't help contradicting one another on the handling aftereffect of the same data, no less than one of them ought to be pernicious. Note that we don't send an information thing and its copies (i.e., authentication information) simultaneously. Rather, we replay the authentication information on diverse administration suppliers in the wake of getting the handling consequence of the first information. In this manner, the vindictive aggressors can't maintain a strategic distance from the danger of being distinguished when they create false results on the first information. In spite of the fact that the replay plan might bring about postponement in a solitary tuple preparing, we can cover the confirmation and typical handling of successive tuples in the information stream to conceal the validation delay from the client. In the event that two administration suppliers dependably give reliable yield results on all info information, there exists consistency relationship between them. Something else, on the off chance that they give distinctive yields on no less than one info information, there is irregularity relationship between them. We don't restrict the consistency relationship to fairness capacity since two amiable administration suppliers might deliver comparable however not the very same results. For instance, the financial assessments for the same individual may change by a little contrast when acquired from distinctive credit authorities. We permit

the client to characterize a separation capacity to measure the greatest passable result contrast.

Definition 1. For two output results, r1 and r2, which come from two functionally equivalent service providers, respectively, result consistency is defined as either r1 = r2, or the distance between r1 and r2 according to user-defined distance function D(r1,r2) falls within a threshold ɗ.

For scalability, we propose randomized probabilistic attestation, an attestation technique that randomly replays a subset of input data for attestation. For composite data-flow processing services consisting of multiple service hops, each service hop is composed of a set of unction ally equivalent service providers. Specifically, for an upcoming tuple $\boldsymbol{d_i}$, the portal may decide to perform integrity attestation with probability $\boldsymbol{p_u}$. If the portal decides to perform attestation on $\boldsymbol{d_i}$, the portal first sends di to a pre-defined service path p_1 —›p_2 • • • • ——› p_l providing functions f_1 —› f_2• • • •—› f_l. After receiving the processing result for $\boldsymbol{d_i}$, the portal replays the duplicates of $\boldsymbol{d_i}$, on alternative service path(s) such as $\acute{p}_1$—› $\acute{p}_2$• • • • • —› $\acute{p}_j$ providing functions f_j as $\acute{p}_j$. The portal may perform data replay on multiple service providers to perform concurrent attestation.

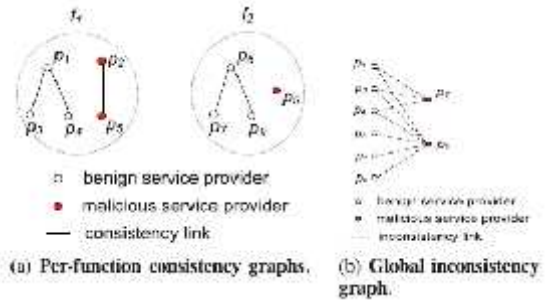


Fig. 3. Attestation graphs.

With replay-based consistency check, we can test practically identical administration suppliers and acquire their consistency and irregularity connections. Fig.3. Authentication charts both the we utilize consistency diagram and irregularity chart to total pairwise validation results for further examination. The charts mirror the consistency/irregularity connections over different administration suppliers over a timeframe. Before presenting the verification diagrams, we first characterize consistency joins and irregularity joins.

Definition 2. A consistency join exists between two administration suppliers who dependably give predictable yield for the same information amid verification. An irregularity join exists between two administration suppliers who give no less than one conflicting yield for the same information amid verification.

We then develop consistency diagrams for every capacity to catch consistency connections among the administration suppliers provisioning the same capacity. Fig 3 (a) demonstrates the consistency diagrams for two capacities. Note that two administration suppliers that are predictable for one capacity are not as a matter of course reliable for another capacity. This is the motivation behind why we keep consistency diagrams to individual capacities.

Definition 3. A for every capacity consistency diagram is an undirected chart, with all the authenticated administration suppliers that give the same administration capacity as the vertices and consistency joins as the edges.

We utilize a worldwide irregularity diagram to catch irregularity connections among all administration suppliers. Two administration suppliers are said to be conflicting the length of they differ in any capacity. Subsequently, we can determine more exhaustive irregularity connections by incorporating irregularity joins crosswise over capacities. Fig. 3(b) demonstrates an illustration of the worldwide irregularity diagram. Note that administration supplier p5 gives both capacities f1 and f2. In the irregularity diagram, there is a solitary hub p5 with its connections reflecting irregularity connections in both capacities f1 and f2.

IV. RESULTS AND ANALYSIS

We first investigate the accuracy of our scheme in pinpointing malicious service providers. Fig. 4(a) compares our scheme with the other alternative schemes (i.e., FTMV, PTMV, and RunTest) when malicious service providers aggressively attack different number of service functions. Inthis set of experiments, we have 10 service functions and 30 service providers. The number of service providers in each service function randomly ranges in [1, 8].

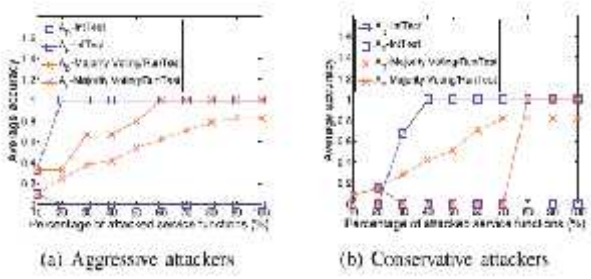


Fig 4.Malicious attackers pinpointing accuracy comparison with
20 percent service providers being malicious.

Each generous administration supplier gives two haphazardly chose administration capacities. The information rate of the data stream is 300 tuples every second. We set 20 percent of administration suppliers as noxious. After the entry gets the preparing aftereffect of another information tuple, it arbitrarily chooses whether to perform information authentication. Each tuple has 0.2 likelihood of getting authenticated (i.e., verification likelihood Pu ¼ 0:2), and two confirmation information reproductions are utilized (i.e., number of aggregate information duplicates including the first information r ¼ 3). Every examination is rehashed three times. We report the normal discovery rate and false alert rate accomplished by diverse plans. Note that RunTest can accomplish the same identification exactness results as the dominant part voting based plans after the randomized probabilistic confirmation covers all bore witness to administration suppliers and finds the larger part inner circle [6]. Interestingly, IntTest extensively looks at both flawlessness consistency charts and the worldwide irregularity diagram to settle on the last pinpointing choice. We watch that IntTest can accomplish much higher recognition rate and bring down false caution rate than different choices. In addition, IntTest can accomplish better identification exactness when vindictive administration suppliers assault more capacities. We likewise watch that when vindictive administration suppliers assault forcefully, our plan can identify them despite the fact that they assault a low rate of administration capacities Fig. 4(b) demonstrates the noxious administration supplier recognition precision results under the moderate assault situations. The various test parameters are kept the same as the past analyses. The outcomes demonstrate that IntTest can reliably accomplish higher recognition rate and bring down false caution rate than alternate options. In the traditionalist assault situation, as appeared by fig. 4(b), the false caution rate of IntTest first increments when a little rate of administration capacities are assaulted and afterward drops to zero rapidly with more administration capacities are assaulted. This is on the grounds that when assailants just assault a couple administration capacities where they can take lion's share; they can conceal themselves from our recognition plan while deceiving our calculation into marking amiable administration suppliers as pernicious. On the other hand, on the off chance that they assault more administration capacities, they can be identified since they bring about more irregularity connections with kind administration suppliers in the worldwide irregularity diagram. Note that dominant part voting-based plans can likewise distinguish noxious aggressors if assailants neglect to take lion's share in the assaulted administration capacity. Be that as it may, lion's share voting-based plans have high false alerts since assaults can simply trap the plans to name favorable administration suppliers as pernicious the length of assailants can take lion's share in every individual service function

V. CONCLUSION

In this paper, we have introduced the outline and execution of IntTest, a novel coordinated administration respectability verification structure for multitenant programming as-an administration cloud frameworks. IntTest utilizes randomized replay-based consistency check to confirm the uprightness of disseminated administration parts without forcing high overhead to the cloud base. IntTest performs incorporated investigation over both consistency and irregularity authentication diagrams to pinpoint conniving aggressors more proficiently than existing systems. Besides, IntTest gives result autocorrect particle to naturally amend traded off results to enhance the outcome quality. We have actualized IntTest and tried it on a business information stream preparing stage running inside a generation virtualized distributed computing foundation. Our trial results demonstrate that IntTest can accomplish higher pinpointing exactness than existing option plans. IntTest is lightweight, which forces low-execution effect to the information preparing administrations running inside the distributed computing base.

REFERENCES

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

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

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

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

[5] T. Erl, Service-Oriented Architecture (SOA): Concepts, Technology, and Design. Prentice Hall, 2005.

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

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

[8] 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.

[9] 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.

[10] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, "Hey, You Get Off My Cloud! Exploring Information Leakage in Third-Party Compute Clouds," Proc. 16th ACM Conf. Computer and Communications Security (CCS), 2009.

[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] H. Zhang, M. Savoie, S. Campbell, S. Figuerola, G. von Bochmann, and B.S. Arnaud, "Service-Oriented Virtual Private Networks for Grid Applications," Proc. IEEE Int'l Conf. Web Services, pp. 944-951, July 2007.

[15] M. Burnside and A.D. Keromytis, "F3ildCrypt: End-to-End Protection of Sensitive Information in Web Services," Proc. 12th Int'l Conf. Information Security (ISC), pp. 491-506, 2009.

[16] I. Roy et al., "Airavat: Security and Privacy for MapReduce," Proc. Seventh USENIX Conf. Networked Systems Design and Implementation (NSDI), Apr. 2010.