ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 7, July 2015

Integrity Attestation on Service Components to Pinpointing the Attackers in Cloud Infrastructure

S. Zubair[1], B.Sowmya[2]

P.G Student, Department of CSE, Intell Engineering College, Affiliated to JNTUA University[1]

Asst.Professor, Department of CSE, Intell Engineering College, Affiliated to JNTUA University[2]

Abstract: Software-as-a-service (SaaS) offers the consumer to utilize the provider's applications working on a cloud infrastructure. All the applications are easily reached from different client devices from a client interface like a web browser. With the Software-as-a-service model, the consumer has slight or no authority how input information is computed, but must be capable to contain confidence in the cloud provider's liability and fulfilment or can organize which input he/she provides to a Software-as-a-service. Firstly user can avoid providing sensible data to the SaaS. Secondly user might be capable to safeguard the sensible data before providing it as input to Software-as-a-service. In the existing system the attackers can escape the detection if they attack only a few service functions. So, in order to overcome this limitation 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.

Keywords: SaaS, PaaS, IaaS, VMs, 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. 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. 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.

However, the move to remote services presents new security challenges [2, 3]. Many potential users, such as pharmaceutical companies, financial services, and government departments have stringent security requirements [4, 5]. One example is scientific provenance. When processing gigabytes of data for climate models or drug trials, a key requirement is that researchers should be able to trust the result of remote computation . If the computer that ran the experiment was insecure, it could be tampered with to produce incorrect results. This could reduce accuracy and cost time, money and the researcher's reputation. Unfortunately, the motivation for attacking and compromising these systems exist, as the recent 'Climategate' scandal has shown, and mechanisms are required for protecting these systems. Users need the ability to establish the trustworthiness of remote services despite the presence of motivated attackers.

For the purposes of this dissertation trustworthiness is defined in terms of behaviour. When users seek assurance of a service, they aim to make sure that it will behave in the manner they expect. This means that security requirements are met, and that more general integrity guarantees hold, including the behaviour of an algorithm, or the reliability of storage. The aim is to go from services which are trusted relied upon without any supporting evidence to assured relied upon because of unforgeable evidence of their behaviour.

This thesis explores the problem of attestation for establishing trust, and answers the following questions:

- To what extent is remote attestation a practical solution for web service assurance?
- What are the key problems, and how significant are they?
- Can it be made more feasible through new tools and software engineering techniques?

In the following section, the motivation for answering these questions is discussed, and several example situations are described.

1.1 Importance of Trustworthy Services

Before diving into the main thesis question – to what extent attestation is a feasible mechanism for gaining assurance in services – it is worth considering whether there is any real need for trustworthy, high-assurance services. Are there situations where an unreliable or insecure service would have a significant impact? The focus of this dissertation is on assurance in terms of security behaviour but algorithmic behaviour, or even formal correctness is just as important.

The Existing System presents IntTest, a scalable and effective service integrity attestation framework for SaaS clouds. IntTest provides a novel integrated attestation graph analysis scheme that can provide stronger attacker pinpointing power than previous schemes.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 7, July 2015

A new integrated service integrity attestation framework for multitenant cloud systems. IntTest provides a practical service integrity attestation scheme that does not assume trusted entities on third-party service provisioning sites or require application modifications. IntTest builds upon our previous work RunTest [1] and AdapTest [1] but can provide stronger malicious attacker pinpointing power than RunTest and AdapTest. Specifically, RunTest and AdapTest as well as traditional majority voting schemes need to assume that benign service providers take majority in every service function. However, in large-scale multitenant cloud systems, multiple malicious attackers may launch colluding attacks on certain targeted service functions to invalidate the assumption. To address the challenge, IntTest takes a holistic approach by systematically examining both consistency and inconsistency relationships among different service providers within the entire cloud system.

The problem is that the attackers can still escape the detection if they attack only a few service functions and take majority in all the compromised service functions.

II. RELATED WORK

A framework/system for trusted storage of a client's data within the cloud is developed. The proposed system is called 'A Trusted Storage System for the Cloud'. As an enormous quantity of electronic data is being generated, there is a requirement of vast storage systems which can hold that data. The requirement is not just storing the data but storing it securely, i.e., the confidentiality and integrity of the data should be maintained. The question of confidentiality and integrity of data comes into the picture when the owner's data is being stored in third party storage systems like the cloud.

National Institute of Standards and Technology (NIST) defines cloud computing as follows: "Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" . Though cloud computing provides cost-effective storage services, it is a third party service and therefore, a user/client cannot trust the cloud service provider to store its data securely within the cloud. Hence, many organizations and users may not be willing to use the cloud services to store their data in the cloud until certain security guarantees are made. Limitations of the Current Cloud Computing Stack To motivate the need for cloud attestation, we must first understand the risks that cloud customers incur in the current cloud computing model. A simplified model of existing cloud services can be represented by the diagram in Figure 1.

Despite the diversity and complexity of services and players that populate the cloud ecosystem, existing cloud services can be grouped according to the abstraction layer at which services are delivered to their respective clients:

- Infrastructure-as-a-Service (IaaS) includes the basic infrastructure services for virtual machine hosting (e.g., Amazon EC2) and data storage (e.g., Amazon S3). Operated by cloud providers like Amazon and Google these services run directly on a hardware infrastructure consisting of geographically dispersed datacenters, each of them hosting thousands of cloud nodes and other hardware elements. The software infrastructure that implements IaaS executes on the cloud nodes and consists of low-level software components, including a hypervisor or an operating system for virtual machine hosting or data storage services.
- Platform-as-a-Service (PaaS) sits on top of the physical infrastructure or IaaS. Similarly to IaaS, PaaS incorporates services for computing and storing data. However, 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 [11]. PaaS services are typically implemented by middleware components that operate on top of the operating system and include execution runtimes (e.g., Java), frameworks, and database servers.
- Service-as-a-Service (SaaS) implement applications such as CRM, games, mail, portals, etc. SaaS services can be implemented on "bare metal", on PaaS, or on IaaS (hosted in a virtual machine).

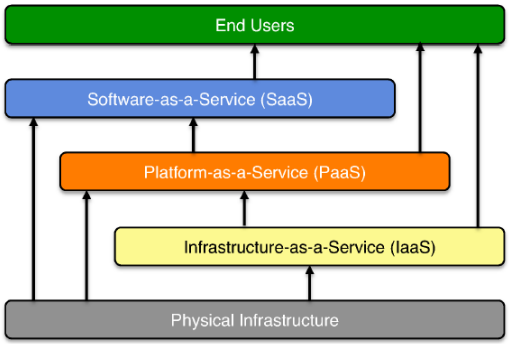

Figure 1: Cloud Computing Layers

III. INTEGRITY ATTESTATION

Cloud devices are contributed research infrastructures composed of a collection of actual owners interconnected by using sites. Every web host can certainly work multiple Virtual Machines (VMs) which could fit in with various proprietors. The appliance supplier (ASP) can certainly rental an accumulation VMs to web host their software products and services. Every service example, denoted through candor, provides a unique info research perform, denoted through fi, for example sorting, filtering, relationship, or maybe info exploration utilities. Numerous service instances is usually functionally-equivalent, giving identical service perform regarding fill balancing or maybe mistake tolerance requirements. Moreover, favorite products and services by natural means entice various service providers regarding benefit. A multi-party service provisioning national

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 7, July 2015

infrastructure usually engages some website nodes to combination various service components into composite products and services while using user's requirements. Anyone accesses impair products and services through submitting insight info towards website node that could onward the person info to various service instances regarding digesting and then provide benefits returning to the person. Web site nodes can certainly authenticate end users allowing simply certified end users to gain access to the actual impair products and services.

3.1 Attack Model

In a common cloud foundation, attackers can claim to be real administration suppliers to give fake administration occurrences or trade off defenseless kindhearted administration cases by abusing their security parts. Our work concentrates on distinguishing the administration honesty assault where a pernicious (or traded off) administration case gives untruthful information handling results. To escape location, vindictive aggressors may need to perform specific duping. Accordingly, the assault recognition plan must have the capacity to catch bad conduct that is both eccentric and incidental without losing versatility. In spite of the fact that we can perform respectability validation on constantly, the overhead of uprightness confirmation would be high, particularly for high throughput information handling administrations in extensive scale cloud frameworks. Therefore, a compelling assault location plan must perform subtle verification, which can keep assailants from picking up information about our authentication plan (i.e., when and which set of information will be validated.). Generally the assailant can trade off the trustworthiness of specific information handling results without being identified by any stretch of the imagination. Moreover, distributed computing frameworks frequently involve countless running numerous more VMs and application administration examples.

The proposed integrity attestation plan has two noteworthy outline objectives: 1) backing runtime persistent confirmation with low overhead; and 2) pinpoint vindictive (or bargained administration examples among countless administration occurrences without expecting any former learning about which administration occasions are trusted. AdapTest embraces an information driven way to deal with accomplish the above outline objectives without forcing any unique equipment or programming prerequisites over remote authenticated administrations, outlined by Figure 2. AdapTest influences the entryway hub to perform administration trustworthiness confirmation. To accomplish non-revocation, every administration occasion is obliged to deliver a receipt for every information it gets and sign the information it has prepared. AdapTest performs assault identification utilizing replay-based consistency check . The essential thought is to copy some unique inputs and re-send them as confirmation information to distinctive practically comparable administration examples for consistency - check. Note that confirmation information and unique information are made undefined to administration occurrences. In addition, our verification plan does not influence the first information handling. At the end of the day, unique information can be directed as before to distinctive administration occurrences for handling taking into account certain heap adjusting and nature of-administration (QoS) administration destinations.

The authentication information are replayed after the entry gets the first information handling results instead of being sent simultaneously with the first information. Therefore, we can keep two conspiring assailants from identifying confirmation by looking at their got information and accordingly getting away recognition. Despite the fact that the replay plan may bring about postponement in a solitary information thing handling, we can cover -the verification and ordinary preparing of sequential information things to conceal the authentication delay from the client. AdapTest influences our beforehand created fraction based calculation to pinpoint pernicious hubs, showed by Figure 3.2. The gateway hub develops a validation chart where hubs are practically equal administration occurrences. Something else, on the off chance that they give conflicting results on no less than one info information, we connect them utilizing a conflicting connection. Since every kind hub will dependably give reliable right results, they will frame a consistency fraction in the authentication diagram. Interestingly, the vindictive hubs will be uncovered with conflicting connections when their mischief is gotten by our validation plan. Note that plotting malignant hubs may attempt to shape a consistency inner circle by continually giving the same wrong results. In any case, on the off chance that we expect kindhearted hubs are the larger part, we can say a hub is doubtlessly malignant if the hub is outside of every last one of clubs whose sizes are bigger than 50% of the aggregate hubs.

Case in point, in Figure 2, we can see the authentication chart incorporates two fractions {s1, s4, s5} and {s2, s3}. Since the extent of the first fractions is bigger than a large portion of the aggregate hubs, s2 and s3 are effectively recognized as noxious hubs despite the fact that they likewise attempt to shape a club through conspiring. AdapTest performs versatile confirmation rapidly uncover vindictive hubs.

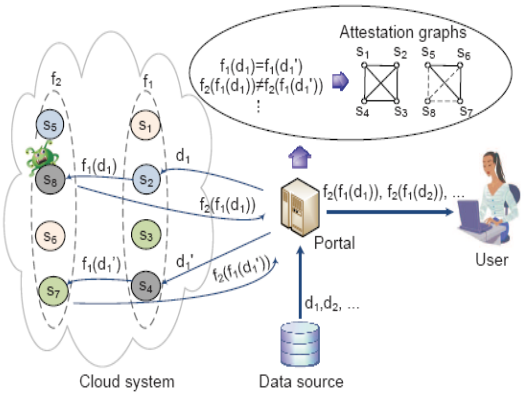

Figure 2: Data-driven service integrity attestation.

DOI 10.17148/IJARCCE.2015.4710

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 7, July 2015

3.2 Weighted Attestation Graph

AdapTest endeavors to pinpoint pernicious administration occurrences without making any former suspicion about the trustiness of any administration occasion. Also, vindictive assailants can perform particular duping amid long-running information handling administrations, which implies the trust score of an administration example must be constantly observed and upgraded. Subsequently, AdapTest utilizes a weighted validation chart to total past confirmation results and powerfully infers an arrangement of trust scores for every administration case, represented by Figure 3. We formally characterize the weighted authentication diagram as takes after.

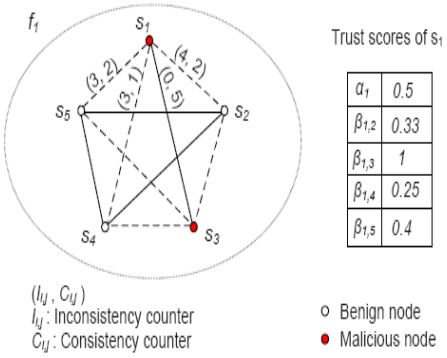

Figure 3: Weighted attestation graph

A weighted authentication diagram is an undirected complete chart comprising of all practically identical administration examples as hubs. The heaviness of every edge comprises of a couple of counters meaning the quantity of conflicting results and the quantity of steady results individually.

3.3 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 (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 in figure 4 below.

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
```

3.4 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 in Figure 5.

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

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.

IV.ANALYSIS

To evaluate the performance of the primitives, it measured their baseline execution time, and studied how the execution time of these primitives depended on their input parameters.

To better understand the cost of cross-world communication, Figure 5.1 plots the execution time of our method invocation benchmark while varying the size of the parameters to be transferred between worlds. The total execution time increases linearly at an approximate rate of 5.6ms/KB.

Finally, to shed some light on the performance impact of cryptographic operations in the attested primitives, Figure 5.2 shows 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.

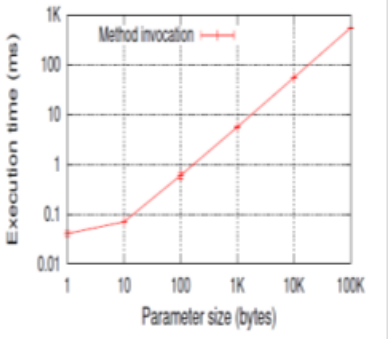

Figure 5.1 Performance of cross world method invocation varying the size of the method parameters

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 7, July 2015

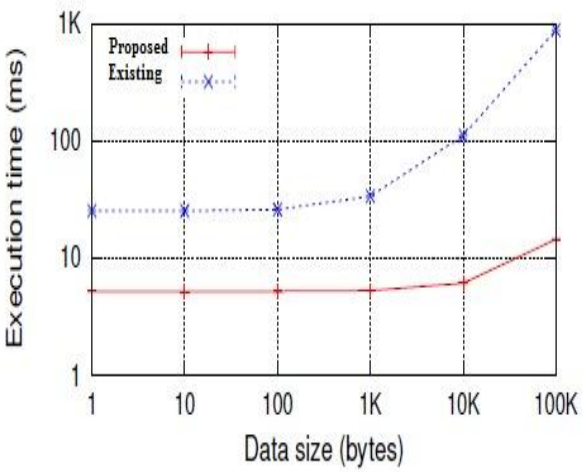

Figure 5.2: Performance of seal and unseal primitives varying the size of sealed and unsealed data, respectively.

V.CONCLUSION

This paper implemented multiple systems aimed at reinforcing user trust in computing platforms. For their popularity and impact, this paper targeted cloud, enterprise, and mobile platforms. This paper showed that, in spite of the diversity of these systems, a common twofold strategy can be adopted for building user trust: (i) enhance the security of their software to provide confidentiality and integrity of user computations, and (ii) provide tangible hardware-based guarantees that such a software is really deployed. The core principles to implement this strategy were borrowed from trusted computing, but the specific techniques had to be tailored for each platform. This is because each platform has unique characteristics and usage models that create specific challenges. This paper provides a scalable and efficient distributed service integrity attestation framework for large scale cloud computing infrastructures. This paper gives a novel integrated service integrity attestation scheme that can achieve higher pinpointing accuracy than previous techniques. It describe a result auto-correction technique that can automatically correct the corrupted results produced by malicious attackers. It conducted both analytical study and experimental evaluation to quantify the accuracy and overhead of the integrated service integrity attestation scheme.

REFERENCES

[1] Juan Du, Daniel J. Dean, "Scalable Distributed Service Integrity Attestation for Software-as-a-Service Clouds", IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 25, NO. 3, MARCH 2014, pp.730-739.

[2] Marty Humphrey and Mary R. Thompson. Security Implications of Typical Grid Computing Usage Scenarios. Cluster Computing, **5**(3):257–264, 2002.

[3] Andrew Martin and Po-Wah Yau. Grid security: Next steps. Information Security Technical Report, **12**(3):113 – 122, 2007.

[4] United StatesHouse of Representatives. Health Insurance Portability and Accountability Act. In Congressional Reports, number H. Rept. 104-736 in Congressional Committee Materials. U.S. Government Printing Once, July 1996.

[5] M. A.Crook. The Caldicott report and patient confidentiality. Journal of Clinical Pathology, **56**(6):426–428, 2003.

[6] ¨UlleMadise and TarviMartens. E-voting in Estonia 2005. The first practice of countrywide binding Internet voting in the world. In Robert Krimmer, editor, Proceedings of the 2nd InternationalWorkshop on Electronic Voting, **86** of GI Lecture Notes in informatics, 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.