A Novel Remote Attestation Platform for SOA

Songzhu Mei, Haihe Ba, Jiangchun Ren, Zhiying Wang, Geming Xia and Huaizhe zhou
College of Computer
National University of Defense Technology
Changsha, P. R. China
{sz.mei, haiheba, jcren, zywang, gmxia, hzzhou}@nudt.edu.cn

***Abstract*—Service-oriented architecture is a design paradigm for the larger-scaled heterogeneous enterprise information systems, and it is the enabling technology for remote and cross-organization e-business transaction. Trust and reputation among the cooperating parties are becoming increasingly vital in this scene, where services from different administration domains are deployed and cooperated. This paper presents a remote attestation platform for SOA. It gives out a complete solution for trust establishing in the SOA enabled IT systems with the supporting of the trusted platform module. The platform provides technique means for the trust management and remote attestation for service composition in the SOA environment.**

Keywords-trusted computing; service composition; trust management; remote attestation

I. INTRODUCTION

Service-oriented architecture (SOA) is a software design and software architecture design pattern based on structured collections of discrete software modules, known as services, that collectively provide the complete functionality of a large software application[1]. Unlike Traditional point-to-point architectures, SOAs comprise loosely coupled, highly inter-operable application services. These services cooperate based on a formal definition which is independent of the underlying platform and programming language. The interface definition encapsulates, or hides, the vendor and language-specific implementation. A SOA is independent of development technology. The software components become very reusable and the system can be more and more heter-ogeneous because the interface is defined in a standards-compliant manner.

Today, with the expansion of service-oriented applications, more and more researchers have realized that trust has become a crucial aspect in service-oriented systems[2-6]. However, the dynamic nature of service-oriented systems makes trust evaluation a challenging task. First, in many service-oriented systems, there is no centralized mechanism for controlling and coordinating the interactions among agents. Hence, the decision making for selecting suitable providers can only be based on information provided by partners (agents) [7], the local view of the agent and/or the experience from previous interactions. Second, in some complex environments, a service-oriented system may contain agents with different trust evaluation criterion and selfish goals.

The rest of the paper is organized as follows. Section II introduces the background of our proposition. Then we provide the main motivation of the platform in Section III. Section IV gives out the design and implementation of platform. Section V overviews the related work.

II. BACKGROUND

A. Security in SOA

Service-oriented architecture (SOA) allows different ways to develop applications by combining services. The main premise of SOA is to erase application boundaries and technology differences. As applications are opened up, how we can combine these services securely becomes an issue. Traditionally, security models have been hardcoded into applications and when capabilities of an application are opened up for use by other applications, the security models built into each application may not be good enough.

Several emerging technologies and standards address different aspects of the problem of security in SOA. Standards such as WS-Security, SAML, WS-Trust, WS-SecureConversation and WS-SecurityPolicy focus on the security and identity management aspects of SOA implementations that use Web services. Technologies such as virtual organization in grid computing, application-oriented networking (AON) and XML gateways are addressing the problem of SOA security in the larger context.

B. Trusted Computing and Remote Attestation

Trusted computing wants to add components and mechanisms to commodity systems to bestow on them some of the properties of high-assurance closed systems like ATM. It requires three core mechanisms [8]:

Secure boot. Make sure that the system is booted into a trusted operating system that adheres to some given security policy.

Strong isolation. Prevent the system from being compromised after it has been booted, and to prevent applications from tampering with each other.

Remote attestation. Certify the authenticity of software being run by a remote party.

The remote attestation is used to attest the configuration of an entity to a remote entity. This procedure is widely used to get integrity information before a client proceeds with the communication in order to use a service or receive data, e.g., digital content. This mechanism is referred as integrity

reporting and can be applied in many scenarios and different applications.

The basic assumption of remote attestation is trusted server, and untrusted clients. In conventional C/S or B/S deployment model, this assumption work well. But in the era of cloud computing, everyone can publish service for the public. These services may be well-designed. They also can have plenty of flaws and even be designed for malicious usage. The assumption can the conventional remote attestation method cannot fit cloud computing's needs well.

Another significant shortcoming is that the remote attestation is always static. It measures only the binary image of an application. This is an "all or nothing" manner. This manner is not suitable for a service-side application. Its availability cannot be assured. It is also a burden to do upgrade and patching for an application

III. MOTIVATION

The basic SOA architecture, in Web services, models the interactions between three roles: the service provider, service discovery agency, and service requestor. The interactions involve the publish, find, and bind operations. These roles and operations act upon the service artifacts: the service software module and its description. In a typical scenario a service provider hosts a network accessible software module (an implementation of a service). The service provider defines a service description for the service and publishes it to a requestor or service discovery agency. The service requestor uses a find operation to retrieve the service description locally or from the discovery agency (i.e. a registry or repository) and uses the service description to bind with the service provider and invoke or interact with the service implementation. Service provider and service requestor roles are logical constructs and a service may exhibit characteristics of both.

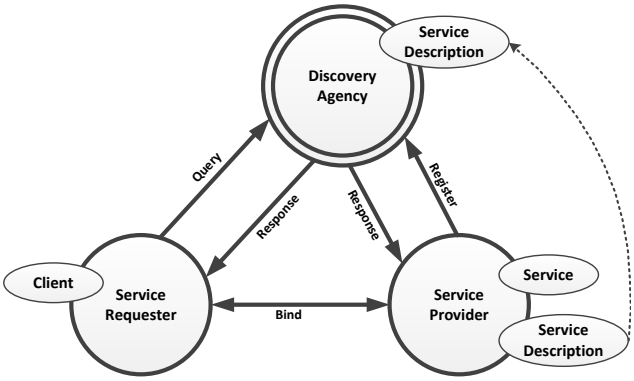

Figure 1. Basic concept model of SOA

Fig. 1 illustrates the basic SOA architecture, in which a service requestor and service provider interact based on the service's description information published by the provider and discovered by the requester through some form of discovery agency. Service requesters and providers interact by exchanging messages, using UDDI and WSDL.

The trust relationship among these three parties is established on the certification and the simply belief of the central CA. This trust is ideogenetic. A service can be forged by a malicious users or attackers. Trusted computing provides the information system the ability of attestation. We will use the Trusted Platform Module and Java virtual machine to build a trusted remote attestation mechanism for SOA.

IV. DESIGN AND IMPLEMENTATION

A. Architecture of Trust Framework

Traditional SOA security methods concern only about the application or message level. They protect the service entity by signature and encryption. These methods ignore the inherent attributes of the service entities. Trusted Computing could be used to guarantee participants in a distributed computing system are returning the results of the computations they claim to be instead of forging them. This would allow the system to be run without expensive redundant computations to guarantee malicious hosts are not undermining the results to achieve the conclusion they want.

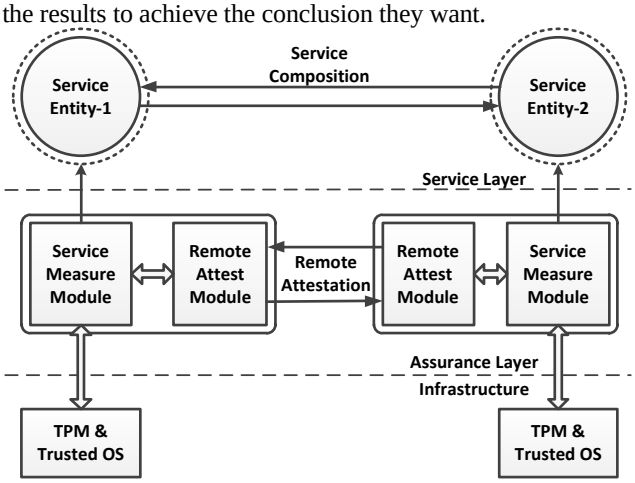

Figure 2. Architecture of Trust Framework for SOA

Fig. 2 illustrates the architecture of trust framework for SOA. For convenience sake, we use two abstract parties to show our ideas. Service entity in Fig 2 can be seen as either the requestor or the provider, or even the discovery agency.

We divides the architecture into three layers.

- **Service Layer** indicates the abstract place where services are deployed and where the service entities communicating and cooperating.
- **Assurance Layer** measure the service layer, including the service entities, service containers, runtime environment et al., and do remote attestation for the remote party to estimate the trust degree of its co-operators.
- **Infrastructure Layer** includes the Trusted Platform Module (TPM) and the trusted operating system. This layer is the support layer which provides the upper layer the basic ability of trusted computing.

In [9-11], we have propose our trusted java virtual machine (TJVM), with the help of TJVM, we can easily grab the static or runtime states of the service entities and give the cooperating parties am encapsulated information about the

service entity. In the next subsection, we will use service discovery as an example to show our idea about remote attestation.

B. Remote Attestation for Service Discovery

We implement the remote attestation mechanism as an individual layer for the reason of high cohesion. Assurance layer, which is separated from service layer, will not invade the normal procedure of traditional service lifecycle. With the help of assurance layer, the cooperation of different parties can be more safety and security.

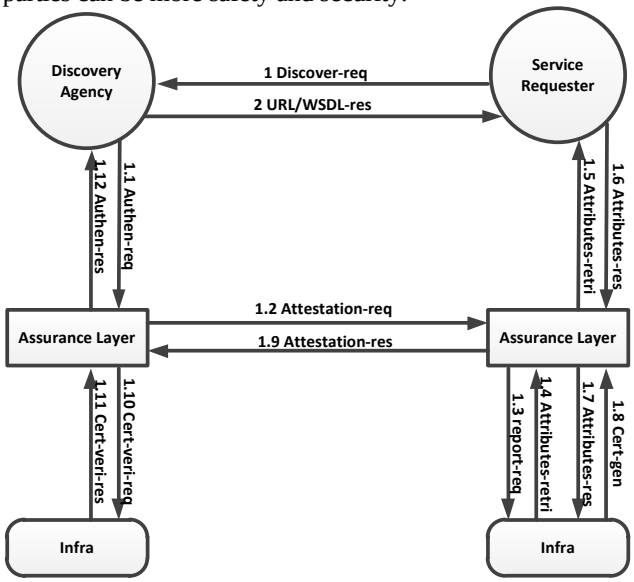

Figure 3. Procedure of Service Discovery with Remote Attestation

Fig.3 shows the procedure of service discovery with remote attestation. In traditional service discovery process, service requestor send a discovery requirement to discovery agency, and the agency response the URL and WSDL documents to the requestor so that the requestor can bind with and access to the provider with the description of WSDL.

After the insertion of assurance layer and infrastructure layer, we add some more steps shown in Fig 3 as step 1.1 to 1.12, when the discovery request is initiated.

Assurance layer, received authentication request from the discovery agency, will send the attestation request to the remote party, the counterpart of the remote party will retrieve the properties and state of the service requestor. Then put the properties and state into infrastructure layer. A report will be generated and encapsulated by the infrastructure and then send back to the discovery agency party. The assurance layer and infrastructure will parse the report and response the results to the discovery agency. The agency will decide whether to send the WSDL document to the requestor according to the result.

In this remote attestation platform the assurance layer plays two roles. The first one is the bridge between the infrastructure and the service. It connects the two layers and provides the ability of hardware enabled measurement and encryption. The second is that it is an implicit, distributed third party for the service layer. It is protected by the lower level infrastructure so that the services and their owners cannot influence it.

C. Bayesian Trust Estimation

Trust is a social concept; SOA endows the information system the society property. So pure technique methods can only represents one aspect of the trust, we still need. Trust is also a hypothesis about a future behavior. It must be inferred with the current state of the composited service, the behaviors we now captured, and the policy the composited service must confine, et al..

A Bayesian network is an appropriate model which provides a statistic method to calculate the probability of a hypothesis under different conditions. The theoretical background of Bayesian network is the rule:

$$P(h|c) = P(c|h) \pounds P(h) = P(c)$$
$$P(h|c) \pounds P(c) = P(c|h) \pounds P(h)$$

For estimating the trust-degree of a composited service, we build a three-level Bayesian network. As depicted in Fig. 4, the root node of the Bayesian network represents the overall trusted-degree T of a composited service. The trusted-degree $T \in [0;1]$, where 0 means the composited service cannot be trusted at all and 1 means the composited service can be totally trusted. The children-nodes of the root node represent the basic beliefs of the composited service. In our model, they are trust of behavior (TOB), trust of state (TOS) and trust of policy confine (TOP). The leaf nodes of the Bayesian network are fine-grinded belief source including most important runtime properties of the composited service.

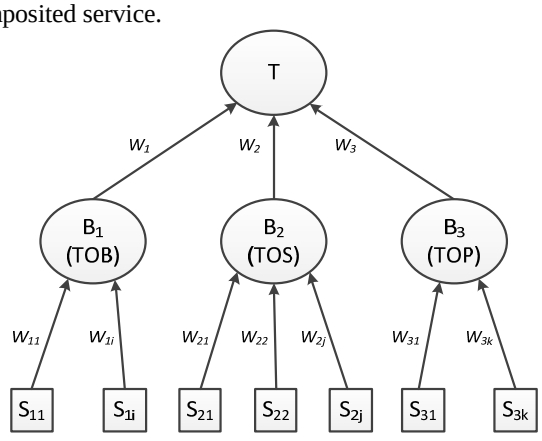

Figure 4. Bayesian network modeling

We can calculate the trusted-degree of the composited service using these formulas:

$$T_{app} = \sum_{1 \le i \le 3} W_{1i} \pounds T_{B_1}$$
$$T_{B_1} = \sum_{1 \le i \le m} W_{1i} \pounds T_{S_{1m}}$$

In the formulas above, W_i is the weight of each basic belief and W_{ij} represents the weight of every belief source.

When estimating the trusted-value of a belief source, we use $P(x_{k+1}|x_k)$ as the Kalman filter's state model, where x_k is the state of a belief source. So we can model the trusted-

value estimating process of a belief source with the formulas below:

$$T_{k+1_i} = P(T_{k_i}) \pounds P(T_{k_i} \, jx_{k+1}) \pounds P(x_{k+1} \, jx_k)$$
$$P(x_k \, jT_{k_i}) = P(x_k) P(x_k \, jT_{k_i \, 1_i})$$

The former one is the prediction process and the latter is the correction process.

With the help of Kalman filter, we can make sure the trusted-degree estimation process itself is trusted and controllable.

V. Related Work

Some research have been done on trusted computing technology which focus on the improvement of trusted computing technology itself and the application of trusted computing technology to distributed systems and cloud environments.

There is some prior work aims to make the mechanism of remote attestation more fine-grained, dynamic. In [13], Sailer et al. propose an integrity measurement architecture (IMA) based on TC technology, which extends the remote attestation mechanism to the application layer of the system by maintaining a measurement list in the kernel. Jaeger et al. use CW-Lite model to reduce the IMA architecture[14], so that only the integrity of the trusted subjects and the information flow between them will be monitored, which resolves the problem caused by wrong input data in IMA. This technology is not suitable to attestation for web-based applications which are always run on the top of JVM or CLR. The applications themselves are treated as the input data of the virtual machines. Halda et al. propose semantic remote attestation[15], which is very similar to our work. But the semantic remote attestation does not support users to establish their security policy, and cannot fit the various applications in cloud environment. Sadeghi et al. propose the remote attestation based on properties[16]. It provides an alternative to the binary attestation. A Trusted Third Party (TTP) translates the actual system configuration into a set of properties and issues certificates for those properties so that to preserve the system's privacy.

A lot of research have been done to solve the problem that how to estimate the trust of an agent or a group of agents. Melaye et al. proposed a Bayesian dynamic trust model for the computational grid[17], but they only give a theoretical model and did not imple-ment their model. Wang et al. propose Cloud-dls, a trust model to make the optimal resource schedule in cloud environment using Bayesian method[18]. Sun et al. proposed an entropy-based trust model for Ad-Hoc network [19]. But these methods are all focus on the reputation and peer-to-peer evaluation in the overall system. They all underestimate the impact to the trust relationship with the state and behavior of the agent itself.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under grant No.60903204, No.61272144 and No. 61070201

References

[1] K. Christopher, A. van der Merwe, A. de la Harpe, "The Role of Service Oriented Architecture as An Enabler for Enterprise Architecture," in Proceedings of AMCIS 2012, 2012.

[2] X. Fan, M. Li, J. Ma, Y. Ren, H. Zhao, Z. Su, Behavior-based reputation management in P2P file-sharing networks, J. Comput. System Sci. 78 (6), 2012, pp. 1737–1750.

[3] S. Ramchurn, D. Huynh, N. Jennings, Trust in multi-agent systems, Knowl. Eng. Rev. 19, 2004, pp. 1–25.

[4] K. Macarthur, "Trust and reputation in multi-agent systems," in Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'08), Estoril, Portugal, 2008.

[5] W. Lin, C. Lo, K. Chao, M. Younas, Consumer-centric QoS-aware selection of web services, J. Comput. System Sci. 74 (2), 2008, pp.211–231.

[6] P. Varalakshmil, S.T. Selvil, A.J. Ashrafe, K. Karthick, B-tree based trust model for resource selection in grid.

[7] V. Lesser, Cooperative multiagent systems: A personal view of the state of the art, IEEE Trans. Knowl. Data Eng. 11, 1999, pp. 133–142.

[8] Martin, M.: The Ten-Page Introduction to Trusted Computing. Research Report CS-RR-08-11, Computing Laboratory, Oxford University, Oxford, 2008

[9] S. Mei, Z. Wang, Y. Cheng, J. Ren, J. Wu & J. Zhou. "Trusted Bytecode Virtual Machine Module: A Novel Method for Dynamic Remote Attestation in Cloud Computing," International Journal of Computational Intelligence Systems, 2012(5), pp. 924-932

[10] S. Mei, Y. Cheng, J. Ren, Z. Wang, J.Ma, J. Wu, Y. Zhao, "TCFI4J: A Trust Enhanced Control Flow Integrity in Java Virtual Machine," in Proceedings of the International Conference on Advances in Industrial Control, Electronics and Computer Engineering (AICECE 13), Kington, Canada, 2013.

[11] H. Ba, S. Mei, J. Ren, Z. Wang, "TEM: A Novel Measurement Method for Java Applications on demand in Cloud Computing," in Proceedings of the International Workshop on Cloud Computing and Information Security (CCIS 2013), Shanghai, China, 2013.

[12] G. Welch, G. Bishop, "An Introduction to the Kalman Filter," Technical Report 95-041, Department of Computer Science, University of North Carolina at Chapel Hill, 1995.

[13] R. Sailer, X. Zhang, T. Jaeger and L. Doorn, "Design and Implementation of a TCG-based Integrity Measurement Architecture." In Proceedings of 13th USENIX Security Symposium, 2004, pp. 223–238.

[14] T. Jaeger, R. Sailer, U. Shankar, "Prima: policy-reduced integrity measurement architecture." In Proceedings of the 11th ACM symposium on Access control models and technologies, 2006, pp. 19–28.

[15] V. Haldar, D. Chandra, M. Franz, "Semantic Remote Attestation: a Virtual Machine Directed Approach to Trusted Computing." In Proceedings of the 3rd Conference on Virtual Machine Research and Technology Symposium, 2004.

[16] A. R. Sadeghi, C. Stuble, "Property-based attestation for computing platforms: caring about properties, not mechanisms." In Proceedings of the 2004 workshop on New security paradigms, 2004, pp. 67–77.

[17] D. Melaye, Y. Demazeau, "Bayesian Dynamic Trust Model, Multi-Agent Systems and Applications," 2005, pp. 480-489.

[18] W. Wang, G. Zeng, D. Tang, J. Yao, "Cloud-DLS: Dynamic trusted scheduling for Cloud computing," in the Expert Syst. Appl. 39(3) 2012, pp.2321-2329.

[19] Y. Sun, W. Yu, Z. Han, K. J. R. Liu, "Trust Modeling and Evaluation in Ad Hoc Networks," In Proceedings of the IEEE Global Telecommunications Conference, 2005, pp. 1862-1867.