Distributed Integrity Measurement for Efficient Attestation of Service Requester Platform

T-06507

Developed By:

Israr Iqbal Awan

(179-FAS/MSCS/F04)

Supervised by

Dr. Muhammad Sher

Dr. Masoom Alam

Department of Computer Science
Faculty of Basic and Applied Sciences
International Islamic University Islamabad

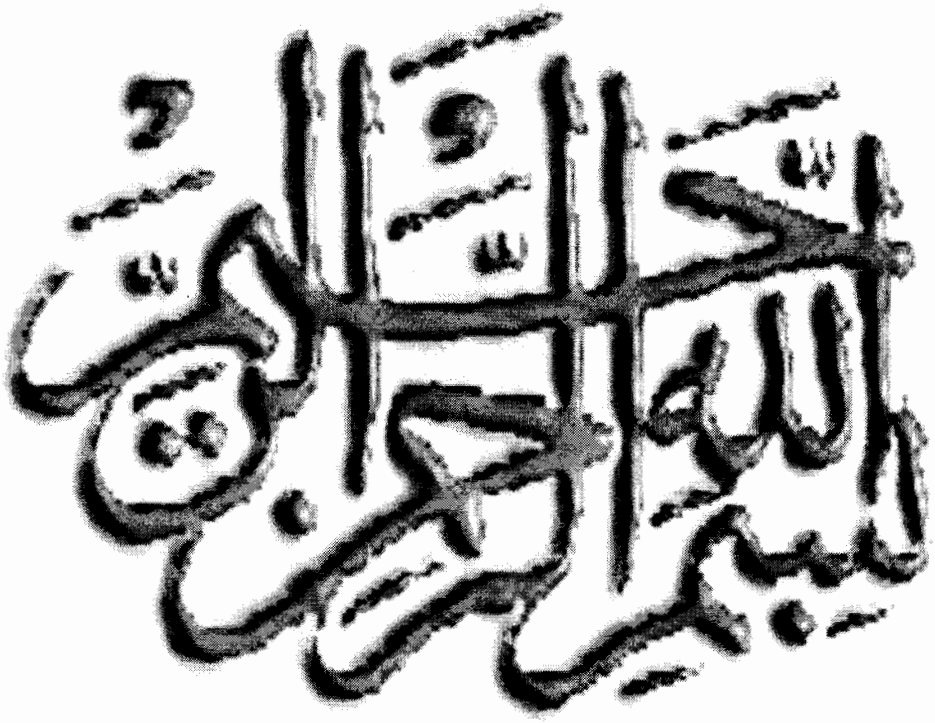
2009

MS
332.1
AWD

Accession No TH-6507

International Islamic Univ.
CENTRAL
LIBRARY
ISLAMABAD.

Banks management
Banks and banking - Computer network resources
Banks and banking - Technological innovations
Home banking services



Dedicated to ONE

Who has all the names, and who does not need any name

A thesis submitted to the

Department of Computer Science

International Islamic University Islamabad
As a partial fulfillment of requirements for the award of
The degree of

MS in Computer Science

Declaration

I hereby declare that this work, neither as a whole nor as a part has been copied out from any source. It is further declared that I have developed the proposed architecture and the accompanied report entirely on the basis of my own efforts and under the sincere guidance of my supervisor Mr. Muhammad Sher. If any part of this project is proved to be copied out from any source or found to be reproduction of some other project, I will stand by the consequences. No portion of the work presented in this dissertation has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

Israr Iqbal Awan

(179-FAS/MSCS/F04)

Acknowledgement

I am very thankful to Almighty Allah, the most beneficent, the most merciful and the most gracious, for enhancing our courage for the completion of this work successfully. It is matter of great honor and pleasure for us to express our ineffable gratitude and profound indebtedness to *Dr. Mohammad Sher and Dr. Masoom Alam* for his kind supervision valuable suggestions and professional activities and inexhaustible energy to steer forth the students. We also appreciate efforts and cooperation of management of Department of Computer Sciences, International Islamic University Islamabad, Pakistan, that made it possible for us to complete our research. I am also greatly obliged to member SERG Group especially *MR Mohammad Nauman* and *Mr Tamleek Ali* for their keen interest, guidance and moral support. I am also very thankful to Mr. Zeshan Shafi and Mr Muazzim Khattak who helped me whenever needed.

Words are lacking to express our humble obligation to our parents whose hands always rise in prayers for our success. Lastly, we thank all our family members and friends who helped us during our study.

Israr Iqbal Awan

Project in Brief

Project Title:	**Distributed Integrity Measurement for Efficient Attestation of Service Requester Platform**
Undertaken By:	**Israr Iqbal Awan**
Supervised By:	**Dr. Muhammad Sher**
Start Date:	**January 2009**
Completion Date:	**August 2009**
Tools and Technologies:	**Java Programming Language**
Documentation Tools:	**MS word, MS Excel**
Operating System:	**MS Windows XP Professional, MS Windows Server 2003 Server**
System Used:	**Dell PowerEdge 2950 Server (Intel Xeon Quad Core).**

International Islamic University
Islamabad

Dated: 12th September 2009

Final Approval

It is certified that the research work presented in this thesis, entitled "***Distributed Integrity Measurement for Efficient Attestation of Service Requester Platform***" is carried out by Israr Iqbal Awan Reg No: 178-FAS/MSCS/F04.

Committee

External Examiner

Dr. Mehboob Yasin

Chairman Department of Computer Science

Comsats Institute of Information Technology Islamabad

Internal Examiner

Mr. Qaiser Javed

Assistant Professor

International Islamic University Islamabad

Supervisor

Dr. Mohammad Sher

Chairman Department of Computer Sciences

International Islamic University Islamabad

Abstract

For online transactions, security is a primary concern for organizations, because large number of computers system connected to Internet is attacked continuously from malevolent software's and viruses. Current security infracts on the cyberspace seldom involve yielding safe connections, as user machine can be easily compromise. If we are employing a safe path to end user machine which is having unknown integrity is highly futile. For the purpose to provide improved security while offering e-Services, it is very important to be aware of that the client system is not tampered with and the system is not compromised.

Current Web Services standards only deal with the security at service provider side. The security at service requester platform is till now is active research domain. Remote Attestation is comparatively new domain of research .Remote Attestation helps authorized party to affirm that trustworthy environment exists on client platform.

To integrate this concept in web services domain, a novel technique WS-Attestation has been introduced however WS Attestation only provide functional prototype upon which more fine-grained attestation mechanism can be design .Further WS-Attestation not explicitly specify requirement for the trustworthiness of client platform, and uses single validation service which led to problem of Privacy, Scalability and Performance.

In this research we propose the distributive and scalable architecture for efficient integrity measurement of service requester platform. We will use XACML behavior policy, we will use distributed validation service for integrity measurement of different domain. In this way we will implement distributive architecture that is scalable in terms of performance and flexible. It is applicable in heterogeneous environment for the integrity measurement of service requester platform.

Tables of Contents

List of Figures

1. Introduction

Chapter 1

Introduction

1.1 Motivation and Challenges

Nowadays hackers focused are client machines as security of client machines can be easily breach. As Safford [24] points out: "In 80's attacker mostly snipped the network connection and actively hijacking sessions of network. Since applications started to use encrypted channels over the network therefore efforts of attackers diverted towards attacking server platforms, those are not configured properly. When companies started to use firewalls, security auditing tools and intrusion detection systems, then attacker turned their attention towards attacking client machine".

The survey of Symantec shows that in last six month of 2007 year 499,811 new malicious code threats detected (Fig 1.1). This is more than 100 percent increase over the previous period and up to 2007 over all number of malicious code threat identified by Symantec is 1,122311,this means that two third of all malicious code threat detected were created during 2007. Other survey shows that approximately half of the desktop computers are infected with viruses [1].

Trojans is also use as one of the tool by hacker to steal information that an attacker can sell or profit from it. For example, the Gampass Trojan [9] was used to steal a user's online gaming account information, which can then be sold to other gamers. Similarly Silentbanker Trojan [10] can be used to steal a user online banking credentials and avert legitimate transactions. This Trojan includes more advanced mechanisms to steal funds from users of online banking.

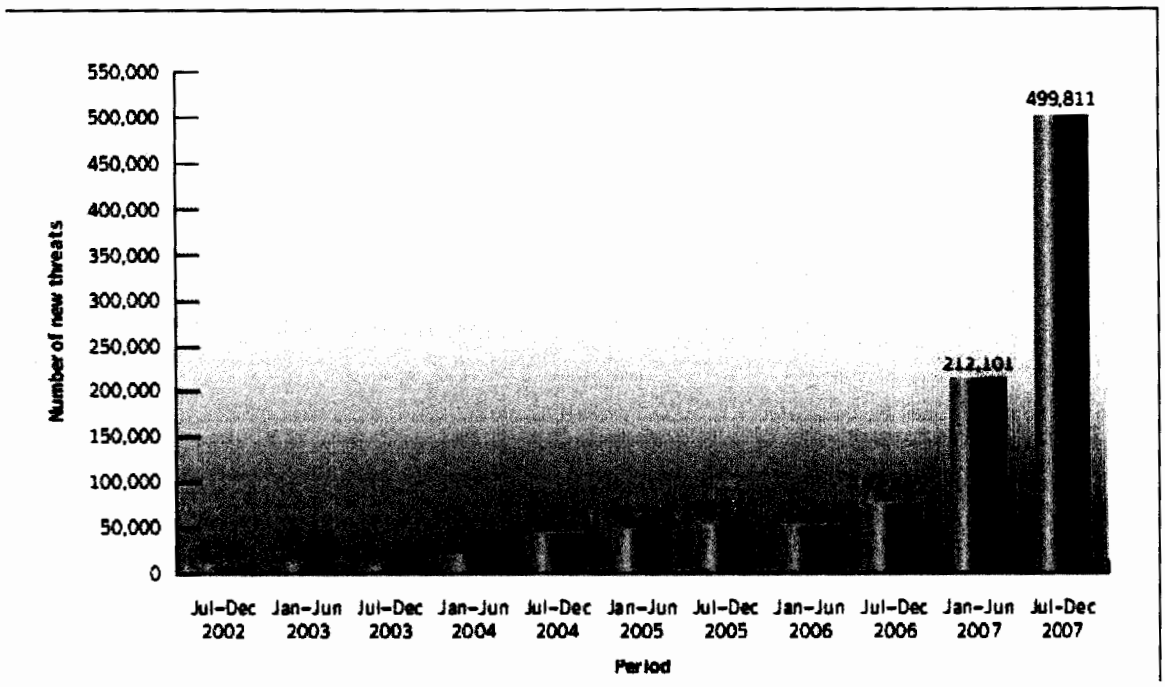


Figure 1: New malicious code threats (source: [8])

Symantec reported 61,940 active bot-infected computers per day During the period of July 1 and December 31 2007, a approximately 20 percent increase from the previous reported period .An active bot-infected computers or machines are defined as those computers that launch usually of at least one attack per day(figure1). Symantec also reported that during this period 5, 06,187 distinct bot-infected computers exist. Distinct bot-infected computers are those computers that were active at least one throughout the observed time [8].Initially the focus of the security researchers has been on the boundary line servers, firewalls. Now days the focus of research is vulnerabilities and attacks on client machines. During the last 6 month of year 2007 there found threat to confidential information made up to 68% of the top 50 malicious programs. Current access control and authentication techniques is not providing any knowledge about the current status of the of the client computer mechanism. it does not know whether any malicious program(virus ,Trojan ,bots) is executing on the client platform and can lead to the leakage of the confidential information. Trusted Computing Group[13] is a association that is currently working on these issues and they are developing the mechanism t o protect the system from these malicious program and if any malicious code is running

on the machine a remote computer can find out this misbehavior .In the subsequent section we address the concepts and technologies that are associated to these issues.

1.2 Background

To know the theory we need to understand the associated technologies those are Trusted Computing, Remote Attestation, Integrity Measurement Architecture (IMA), WS-Attestation and state of the art access control mechanism in brief.

1.2.1 Trusted Computing

The term "Trusted Computing " is define as technology in which personal computers, consumer devices and personal digital assistant are equipped with special security chip to support cryptographic mechanisms to help in verifying the application software integrity or system software integrity running on the client platform.TC also protect I/O operation and storage of data within the device. This mechanism is used to provide protection against the malicious programs. The motivation behind the evolution of this technology is that current techniques for fighting against the malicious program and other threats are at the software level. The solution at the software level is always tend to be compromised in some way it has been found out from the past experiences that trusted and tamperproof security is difficult to achieve only using software level solution [1].

For the purpose to shape these efforts leading hardware manufacturers have formed the consortium recognized as Trusted computing Group[16] the motive was to designed trusted personal computers , PDA's and mobile phone that enable them for all e-applications for example e-health , e-govt , e-commerce, m-commerce in a trust worthy way . According to the specification that is defined by Trusted Computing Group (TCG) for the Trusted Computing enabled devices, they have the four technical functionalities.

- **Protected I/O:** All the communication between the I/O devices such as monitors, sound cards should be encrypted so in that a way only intended applications can read and [process this information.

- **Curtaining Memory:** Data that is stored in the memory should be strictly isolated from other applications by using encryption mechanism.

- **Sealed Storage:** Data stored in Permanent storage media such as hard-drive should be in a sealed (encrypted) form by this way only the originating application or device can read it. The motive is that if the data is moved from the sealed storage maliciously to another device, the data is not readable any more because the data is sealed through the encryption mechanism by application and no other application knows the key for successfully decrypting the data.

- **Remote Attestation:** The important feature of trusted is to verify form remote machine in enciphered form about the type of software running , whether malicious program has changed the corresponding software, current status of the hardware components etc. This capability helps the service providers to provide their services over geographical boundaries.

Researchers in information security reach a conclusion that all security problems is difficult to solve by software alone and most of the conventional operating system provide security relying on hardware characteristics for the separation of supervisor and user modes. Trusted Computing helps in assuring platform behavior to the user of that computer and to manufacturer/vendor of company PC. Trusted Computing also provides assurance to the local user and remote party over network by mechanism known as Remote attestation.

When a system boots initially the BIOS is in control. That system carries out enough subsystems to do basic I/O and initialize cards. After this it transfers control to the BIOS of different installed cards and then receives control back for the completion of further task. After this is done, boot loader takes the control, which in turn passes control to the operating system kernel. The operating system initialize different needed device drivers and services and then may start up a program, or be asked to start a program by the user of a system . By the time the user can direct the operating system to perform something; we can observe control over the machine has transferred between a numbers of different components. In this scenario, how can a user know that the system has not been compromised or hacked any number of times that enable cracker access to everything he wants to do with the compromised system.

1.2.2 Trusted Platform Module

The key element in Trusted Computing is the Trusted Platform Module chip. The TPM provide secure data storage and secure storage of secret keys. The TPM is uniquely identified by its Endorsement Key. The manufacturer burn EK at the time of manufacturing and cannot be altered. Public key is used when important data is sent to TPM and for attestation purposes. The secret key never leaves the TPM. Attestation Identity Key is generated by TPM and signed by trusted party distinctly identifies the particular user on a specific machine.

The TPM include register those are specifically designed for purpose to store 160-bit hash value of any software running on a platform and are known as Platform Configuration Register. When system boots PCR are cleared and each software component of the computer is mapped to hash value of 160-bit that constitutes good state of that software component. Every newly computed hash value is concatenated with current hash value in the PCR.

The new calculated hash value stored in the same PCR, so the small number of PCR will use to store measurement information of the whole system .SML stores information of the sequence of measurement in the PCR. Remote Attestation is a technique in which service requester platform presents its PCR measurement to a challenging party. Normally in a typical remote attestation scenario platform TPM collect current PCR measurement and sign it by using AIK. The signed PCR is then forwarded to the challenger along with SML.

The challenger by using SML and PCR supplied by target platform re-compute value of PCR and compare expected value with the value supplied by target platform. After making comparison, the challenger can find out whether the integrity of application/system software running on target platform is valid or not. A comprehensive introduction to remote attestation and trusted computing can be found at [13]. The TPM has been designed to protect security by ensuring the following.

- Private keys can't be sneaked or given away.
- Modification in the software code is always noticed.
- Private keys are saved from being used by illegitimate program
- Sealing keys are not easily accessible to a hacker

The TPM achieves these motives by using three main mechanisms those are the following.

- Authentication through public key cryptography
- Measuring the integrity of the software
- Remote Attestation mechanism.

By using Public key authentication security mechanism the objective of on-board key pair generation and with public key signature, verification, encryption is achieved .As the key pair is generated in the chip and when any time they are transferred outside the chip these keys are encrypted therefore malicious programs cannot access the keys.
Even the owner of the computer system if want to find out the keys cannot access the keys because the keys are never visible outside the TPM .Malevolent programs could use the private

keys on the chip so some technique must be used to protect the keys so malicious code cannot use the private keys if they capture the keys .

The IMA is designed for such a purpose it has the capability to provide protection for private keys so it cannot be accessed by malicious code. when the system boots up the TPM chip saves the hashes of the configuration information as the boot sequence proceed .once the system boots up the keys are sealed in the program configuration register .the information that is once sealed in the platform configuration register can be unsealed if the platform configuration register has the same value as at the time when they are sealed. so if hacker made effort to boot the an alternative system or in other scenario if virus has backdoored the system in someway ,the value stored in the platform configuration register will not match and process of unsealing the data will fail so in this way the data will be protected from unauthorized access.

The attestation functions maintain a record of all measurement of the software those are stored in the platform configuration register and alter can be sign with secret key of the trusted platform module. In the distributed environment the trusted client can verified by the service provider that the client has compromised software or not compromised software. It is very difficult to fix the current client software, because of the complexity, compatibility reasons. First reason is that modern systems are very complex and consist of million of lines of code and it is very difficult to fix this code. A typical Linux system with standard applications having 100 millions lines of code. If we take the example of the large application that often have hundred of million lines of code. It is found out from the recent surveys that product level software has approximately at least one security related bug per/thousand lines of code during its life time. So we can say that in typical system we have hundred thousand security bugs and we are finding approximately six thousand bugs per year.

The second reason is the compatibility requirement of the client system. even if we are succeeded in building a secure software system, the amount of time and effort required to fix the security bugs in the billions lines of code is simply prohibitive.

The third reason is that without the support of the hardware it is very difficult to discover the existence of the malicious programs in the system.

1.2.3 XACML

XACML is an OASIS standard for the specification of composite access control policies. It is consist of three main components: 1) an XML based language for the definition of complex access control scenarios, 2) a request/response protocol which helps in insulation from the underlying environment and 3) an abstraction layer in the form of a data flow model which gives distributed access control [30] across diverse platforms and any customization for a particular environment

1.2.4 XACML BEHAVIORAL ATTESTATION

Author [29] proposed new framework, Behavioral Attestation for Web Services, in which XACML is built on top of WS-Attestation. They propose a type of XACML policy called XACML behavior policy, which defines the expected behavior of a partner platform. The author used Existing web service standards to integrate remote attestation at the web services level. The author also presented the prototype of the system which implements XACML behavior policy using low-level attestation techniques.

1.3 Thesis Outline

This thesis is divided into three phases. In the first part, In-depth literature review on the existing Trusted computing, Remote attestation was carried out. In the second phase, an analysis of the current remote attestation architecture with proposed distributed IMA for remote attestation was conducted. The system design and its implementation with a possible solution are proposed to solve the problems in the previous proposed remote attestation architecture. In the last portion the testing and performance evaluation are examined and also proposed the future work.

2. Literature Survey

Chapter 2

Literature Survey

2.1 Introduction

Basically Remote Attestation is comparatively new area of research. We can find rich material exist on system security, Trusted Computing (TC) and IMA. I studied a large number of thesis, research publications, articles, books etc, so that I can find out what has been done in this area in which our problem domain resides. Some of the current research is reviewed as follows.

2.2 Related Research

Schoen [31] in "Trusted Computing: Promise and risk" explains "Trusted computing" based architecture for solving some of today's security problems by using hardware component support to the personal computer. As we know there is a strong debate on the TC controversies that are going on for the last few years in the industry. The paper has some strong opinion in favour of Trusted Computing.

Shapiro[20] "Establishing the Genuinity of remote computer system" the author identified basic problem in distributed system environment that is to find out whether the compute rat remote site can be trusted that is autonomously accessing organization secure resources through a network. They discussed a mechanism by which a computer system at remote location can be challenged to show that it is genuine and trustworthy .after proving itself a legitimate client, the client can be granted access to distributed resources. This work is good research contribution and gives very good understanding of attestation

of attestation of remote system through cryptographic mechanism using software systems. The paper basically explains general purpose genuinity check and not discussed the whole system status, without using any hardware component for storing secret keys. So the system can be vulnerable to different type of software attacks.

Sailer [25] in "Design and implementation of TCG-based Integrity Measurement Architecture "proposed the basic design and implementation of a new and novel secure integrity measurement system. All the executable component of the Linux operating system is measured and these measurements are saved before execution. These saved measurements are protected by the TPM, which is main part of the TCG standard. The IMA is the first to extend the concept of TCG trust measurement concept to dynamic executable content. it means from the Basic Input Output System (BIOS) all the way up into the application layer. They propose trust measurement paradigm for a web server application, where they demonstrate how the proposed architecture can detect undesirable invocations, for example rootkit program. They show that the measurement mechanism is practical with respect to performance of the system.

Heroshi [21] in "Trusted Platform on Demand (TPoD)" an author propose a new architecture called trusted platform on demand and also implement the proposed architecture. The architecture increases the trustworthiness of networked system by using dedicated security hardware. The proposed architecture uses a secure operating system kernel with open security protocol which provides a secure platform that may be used for hosting a wide range of distributed applications. The main significant of the proposed architecture is that application can effectively protect itself even in situations where there is vulnerabilities in application or system software. This new paradigm provide a good understanding of maintaining and also checking integrity of platform through dedicated hardware .TPoD architecture use a trusted boot sequence in which BIOS and boot loader is attested e.g.(GRUB integrity). The IMA extends and use the chain of trust proposed and implemented by TPoD by attesting load time integrity of Linux operating system and application loaded in it.

Sadeghai in [32] has discussed the limitation of attestation mechanism and sealing process proposed by the existing specification of the TCG. They argued that these techniques can be negatively used to discriminate specific platforms i.e. their operating system and eventually the specific vendors. They show that how their proposed methodology eliminates these problems by introducing a new paradigm in which the attestation of the platform should not rely on the specific hardware or software .but only on the platform properties. The properties based attestation only can check whether the platform properties are enough to fulfil certain security requirement of the challenger asking for the attestation .they discussed a number of solutions that is based on the existing TC hardware. The proposed methodology is very interesting, but when we require recognizing a wide range of configurations and then mapping them to a specific properties then problem arise.

Shi [26] proposed BIND (Binding Instructions aNd Data) a novel attestation technique for securing distributed system. They introduced the mechanism of code attestation. current code attestation technology is comparatively is not mature due to the vulnerabilities exist in software version and configuration .The main problem of time-of-use and time of attestation remained to be there , it means code may be correct when attested but it may be changed when it is used. Their main focus of research is t o address these problems and made effort to make code attestation more usable in securing distributed system. BIND proposed a fine grained attestation by proposing the ides that instead of attesting entire memory data, only the portion of code that challenger is concerned about that as a result it greatly simplifies verifications. Their effort is to narrow the difference as much as possible between time-of-attestation and time-of-use. BIND uses a technique in which it measures the piece of code immediately before it is executed .BIND introduced sand-boxing technique to give protection for the execution of the attested code. Beside this it couples the code attestation with the data that it produced as a result challenging party can point what code to be run to produce the result.

Li [33] in "An Efficient Attestation for Trustworthiness of Computing Platform" showed some fatal limitation that must be overcome for example leakage of privacy information regarding platform configuration. Author has proposed a new behavior based attestation. Behavior based attestation determine the trusted state of target platform by using trustworthiness related behavior of the system. The author claimed this gives privacy protection of system and has high feasibility. They were the first to propose attestation technique based on system behavior that can help him effectively reduce impacts that is caused by malicious programs (Trojan, virus).

Safford [19] in "Trusted Linux Client" presents the idea of trusted Linux client that was used to protect Linux client from offline and online integrity attacks. This technique is also transparent to the end user that is accomplished by using TPM security chip, digital signatures and verification of extensible trust characteristics of all contents. The author proposed a technique that how we can make Linux as trusted system. The author explains new technologies that are available on Linux operating system to make it trusted platform.

Yoshihama in [28] "WS-Attestation: Enabling Trusted Computing on Web Services. Test and Analysis of Web Services "propose a new architecture that is called WS-Attestation. This architecture is built on top of the framework provided by web services. the author claims that proposed architecture is software oriented .dynamic and fine-granular mechanism for attestation , which help TCG and WS-Security technologies for increasing trust integrity reporting. The proposed technique also helps in binding of attestation with application context. The author explains that if a user wants Online-purchasing and trust online shopping service .when user submits his credit card number even the service provider organization is trustworthy and honest with clients .the server might be infected with a malicious program such as Trojan horse. The malicious program surreptitiously send the credit card number to a remote attacker .in another scenario server platform software might have vulnerability that attacker can use to obtain the super user privilege and steal customer credit card information. Therefore it is very important to ensure that services running on a platform that is trustworthy. it means we must ensure about the services provide by the platform ,it is running on the hardware that it claims to be and

operating system and application running on it are not infected by malicious programs and has no vulnerability.

Author in [29] proposed new framework behavioral attestation in which XACML is used on top of WS-Attestation .they propose XACML policy which define expected behavior of target platform called XACML behavior policy. Current Web Services standards are used to incorporate Remote Attestation at the Web Services level. Author presented model that implements XACML behavior policy.

Current TCG specification defines only primitive attestation mechanism that has many limitations for use in practical scenario. Attestation mechanism is coarse grained, dynamic states of the system are not captured and it is complicated to validate integrity metrics. The platform attestation status is not bound to the status as of communication. The target platform configuration information is not protected from hackers.

Cryptographic protocols such as Secure Socket Layer(SSL) [5] or those used in Virtual Private Networks (VPNs) that gives end-to-end channel security, but say nothing about the integrity of the platform on either side of the communicating parties.
For example, it is possible for a compromised client to initiate an SSL connection as the SSL connection only encrypts network traffic Consider an office employee connecting to his organization network from outside it through VPN software. The VPN software only helps in authenticating the employee, but cannot check platform integrity of the particular client system with which the connection is being made. There is a possibility that client system may be compromised, or may leak company private information once they are downloaded on it. It means, cryptography alone is not a complete solution to security because it only secures the channel or authenticates the client but does not say anything about the behavior of the system.

Current Web Services standards only deal with the security at service provider side. The security at service requester platform is till now is active research domain.

Remote Attestation is comparatively new domain of research. Remote Attestation helps authorized party to affirm that trustworthy environment exists on client platform.

To integrate this concept in Web Services domain, a novel technique WS-Attestation has been introduced [28] .yet, WS Attestation only provide functional prototype upon which more fine-grained attestation mechanism can be design. Further WS-Attestation not explicitly specify requirement for the trustworthiness of client platform.

More specifically, in WS-Attestation, the design of explicit criteria for the trustworthiness of a SR or client platform is left unspecified, and uses single Validation Service (VS) which led to problem of Privacy, Scalability, and Performance. Privacy is primary security concern in remote attestation using VS, because if we use single VS, we disclose the whole configuration of SR platform (i.e. which operating system and Database Management System, Security Application). If the VS are compromised then attacker has the whole configuration of the SR platform that can be used for attack. Using single VS is not scalable as if thousands of client accessing the services of the VS, so this architecture is not feasible it can lead VS unresponsive in some situation and unreliable. VS is calculating hash for thousands of users simultaneously so the performance will also lack, if we use single VS for all domains performance of the system will be not satisfactory.

2.3 Summary

With the growing reliance of societies on usage of information systems, the necessity and need for security is rapidly increasing. The increase in private and sensitive information on interconnected networks has given rise to the need for the development of mechanism which helps in protecting privacy of individuals on the Internet. TC is one such paradigm geared for the assurance of security through a hardware-based solution. In the next chapter we described the problem scenario and focus of research of the proposed system.

3. Requirement Analysis

Chapter 3

Requirement Analysis

3.1 Introduction

In order to provide better security while providing online services, we have some basic component of the service structure. The main and basic components of any online service provider are server or service provider communication channel or protocol used and consumer of the service or service requester. We can define the following main component of service architecture.

- Service Provider Server(online server providing services to the clients)
- TCP/IP being the communication protocol
- Service Requester(client machine)
- Validation Server(trusted third-party which can perform attestation on behalf of the Service Provider)

As been discussed in the previous chapter the attestation architecture given by WS-attestation using single VS, we have issues related to the privacy of the client or service requester platform configuration and architecture using single validation is also not scalable and performance will also lack, When thousand of client accessing the services of validation server. Now after analyzing the need of such architecture that solves the above discussed problem. Our proposed system should have the following capabilities.

- Provide improved privacy for the service requester platform
- Provide improved performance when large no of client accessing the services of service provider
- Architecture must be scalable
- Incorporate delegated model.

3.2 Problem Scenario

Online service provider still has reservation regarding integrity of the service requester platform. Making the appropriate resource such as data availability, but only to right people; avoiding data leaks; securing digital assets and remaining complaint with financial, medical or corporate regulation are main reason for widespread adoption of remote worker programs.

Cryptographic protocols for example SSL [5], used in VPNs, only provide end-to-end security, but cannot guarantee about the integrity of the platform on either ends. For example, it is possible for a compromised client to initiate an SSL connection that only encrypts network traffic. Consider an office working employee connecting to his organization network server from outside it by using VPN software. The VPN software only helps in authenticating the employee, whether the employee can access the secure data of organization or not but cannot check platform integrity of the particular client system with which the connection is being made. There is a possibility that client system may be compromised, or may leak company private information once they are downloaded on it. It means, cryptography alone is not a complete solution to security because it only secures the channel or authenticates the client but does not say anything about the behavior of the system. The solution to this problem is Remote Attestation that helps authorized party to affirm that trustworthy environment exists on client platform.
To integrate this concept in Web Services domain, a novel technique WS-Attestation has been introduced [28] .yet, WS Attestation only provide functional prototype upon which more fine-grained attestation mechanism can be design. Further WS-Attestation not explicitly specify requirement for the trustworthiness of client platform and uses single

VS which led to problem of Privacy, Scalability and Performance. Privacy is primary security concern in remote attestation using VS, because if we use single VS, we disclose the whole configuration of Service Requester platform (i.e. which operating system and DBMS, Security Application). If the VS is compromised then attacker has the whole configuration of the Service Requester platform that can be used for attack. Using single VS is not scalable as if thousands of client accessing the services of the VS, so this architecture is not feasible it can lead VS unresponsive in some situation and unreliable. VS is calculating hash for thousands of users simultaneously so the performance will also lack, if we use single VS for all domains performance of the system will be not satisfactory.

3.3 Summary

The requirement for the distributed integrity measurement architecture is as follows:

1. Service provider must be able to specify the protected resource.
2. It must have capability to generate distributed request to different validation servers.
3. It must have capability to process behavior policy.
4. Based on the integrity status of the appropriate resource the response should be sent to the client or service requester.

Similarly validation server has the following requirement as follows.

1. Validation server must be able to validate request by using its database of good known hashes of the application for which the validation is requested.
2. Validation server must be able to send response according to the result evaluated.

4. System Design

Chapter 4

System Design

4.1 Introduction

In the design phase I will cover all the aspects of the research project. This is then further helpful in coding and the implementation. In the first phase of this thesis I have gather requirement analysis of the proposed research topic i.e. about the Trusted computing, WS-Attestation. Review the literature related to Remote Attestation and different technique for attesting the code and new challenges in the remote attestation. In second phase analyses the existing Remote Attestation paradigm and problems in the proposed model. In last phase, possible solution is proposed for the problems in previously used remote attestation architecture.

4.2 Research Method

In the design stage all the characteristic of the proposed architecture are included and coding, implementation in performed on these procedures. I have used conductive research method. Understanding and insight study of the problem was presented by involving qualitative techniques. During the research, literature survey was completed; using the Internet and IEEE's website, where I have found the most recent reports about the current research is going on the current remote attestation architecture available.

4.3 Methodology / Algorithm

In our proposed architecture as shown in figure 3 we have embedded the new component in remote attestation architecture that is distributed integrity evaluation manager. WS-Attestation proposes three architectural models (Fig 2). In the first model (Fig 2a), a SR takes an attestation credential from a VS and provide to the service provider is known as pushed model .In the Pulled Model (Fig 2b), the SR embed its own PCRs and SML

information with the service request when sending request to service provider. The service provider then send these credentials to the VS for the verification of different properties of the SR platform using sent PCRs. The third model is the Delegated Model (cf. Fig 2) and in this model service provider requests the VS for attestation on behalf of

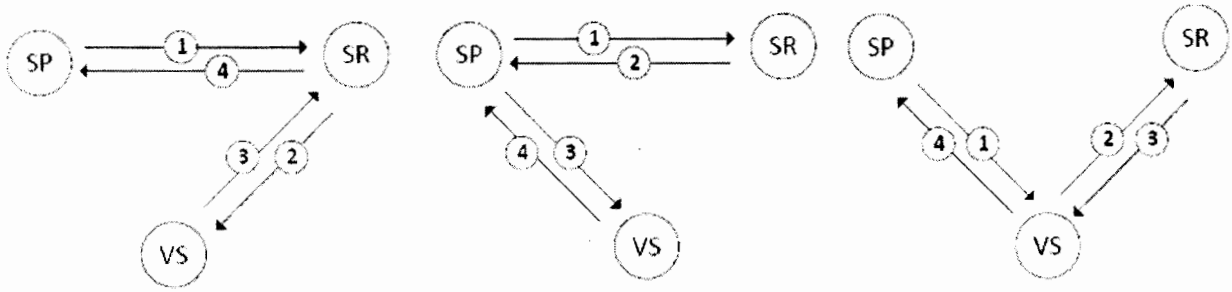


Figure 2: Attestation Models (Reproduced from [23]

the service provider. We have utilized delegated model. The advantages of delegated model are that 1) the integrity privacy of the attested platform is protected; 2) the attestation requester does not have to be capable of validating attestation. Delegating model is the most important model for incorporating attestation in heterogeneous environment and distributed environment. In our proposed architecture we have three main parties that communicate with each other and those are SR, service provider and VS. SR is client that is accessing some resource on the server or need some services of the service provider. Service provider is the party that provides some kind of services to their clients. VS is trusted third party service, which provides attestation parameters each of which asserts a security properties that represents the state of a measured platform. As shown in the given figure 4.2 SR or client want to access the service of the online service provider generates access requests that is intercept by policy enforcement point of the service provider that eventually the policy decision point try to evaluate the request. The distributed integrity evaluation manager in the policy decision point checks behavior policy to verify that whether the integrity measurement is needed for the client. If behavioral policy is enabled and there is a need of integrity measurement of the SR then required information extracted from the request and distributed requests are generated to different VSs those have the capability for measuring integrity of application of different

domains. After generating the distributed request the distributed integrity evaluation manager wait for responses from different VSs. When response arrives from VSs then information about the integrity measurement is used for evaluating the request of the SR.

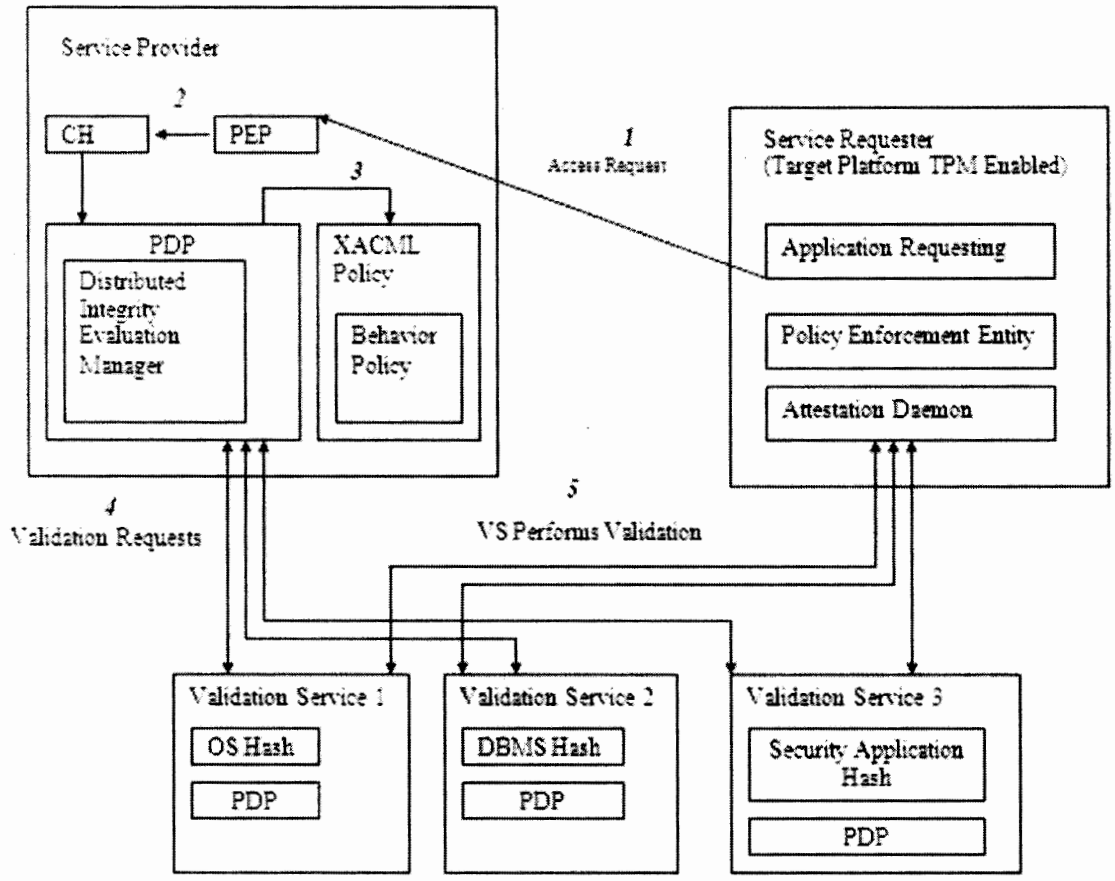


Figure 3: Architectural Enhancements for Distributed Integrity Measurement

The algorithm of the proposed design is as below.

Algorithm for Service Provider

{

Wait for request of SR

Accept the request

If (request received)

Parse the request file

Parse the policy file

If (behavior_policy=true in policy)

{

1. Extract the information about the identity of the client from the request file.
2. Send extracted information to the distributed integrity evaluation manger.
3. Distributed integrity evaluation manger will use the provided information and passed these parameters for further processing to validation servers.

}

Else

{

1. Policy decision point evaluate the request locally
2. Generate the response file
3. Make socket connection with SR
4. Send response to the client
5. close socket connection

}

Algorithm for the DIEM (Distributed Integrity Evaluation Manager)

{

1. Extract the information about the client requesting resources from the request.
2. Make socket connection with VSs for sending request.
3. Send distributed request to different Validation Servers.
4. Wait for response until response arrive from all validation servers to whom request are sent.
5. If(Response arrive from all validation servers)

{

1. Combine all the responses received from different validation servers.
2. Send response to Policy decision point for evaluating the request.

}

else

{

Generate Error message

}

}

Algorithm for the Validation Server

{

1. Calculate 10th PCR value by using hashes in the SML.
2. Match the expected PCR with the value supplied by the SR.
3. Check the nonce supplied by the SR with the nonce sent to it.
4. If the expected PCR and nonce is valid

Then jump to 1:

OR

Jump to 2:

1:

{

Access the database in which good known hashes is saved

Check whether hash is valid

Distributed Integrity Evaluation Manager

Send response to Distributed Integrity Evaluation Manager

}

2:

{

Create response for the Distributed Integrity Evaluation Manager

Send response to Distributed Integrity Evaluation Manager

}

}

4.4 Operational Details

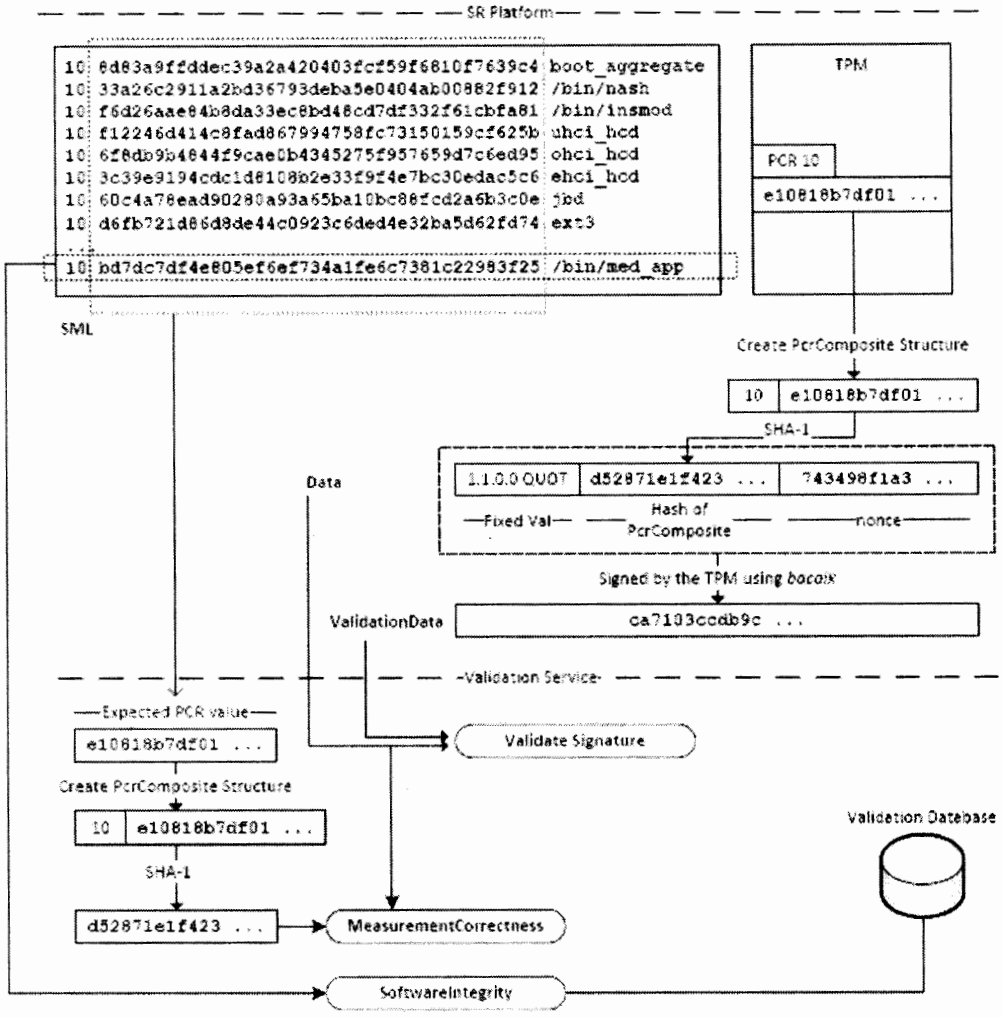


Figure 4: Operational Details

4.5 Summary

The most important issues which were considered in the design of the proposed architecture is the design of distributed integrity evaluation manager that must be able to process behavior policy and generate distributed XACML request to different validation severs and receive responses from them. Our proposed architecture is scalable, privacy issues are solved and performance is also improved using the distributed architecture.

5. Implementation

Chapter 5

Implementation

Introduction:

This phase of the project will include the algorithm, explanation of classes its implementation in Java environment. I have chosen java because of it platform independency. I have mentioned the functionality and requirements. The pseudo code of the classes will be shown at abstract level of all the method of the classes used in the project. The functional description of the data flow diagram of the proposed architecture will be included.

5.1 Deployment Environment

The system development, implementation and testing is very critical phase of the development of the proposed system. During this phase the needed application programs are developed and tested.

5.1.1 Tool/Language Selection

The selection of software is key factor to be considered in the development phase of a new proposed system, because the choice of the software depends upon many factors. Those factors are requirement of the proposed system, current environment (i.e. existing software), amount of data to handle and the cost of programming. After analyzing the nature of the problem and considering the requirement, I choose the Java due to its novel features.

Our developed system will be run on any platform. Therefore, the developed system can be used in any operating environment. For java programming we used Kawa 5.0 professional edition IDE and used Sun's XACML implementation.

5.1.2 Features of Java

The basic features that make Java programming language powerful and popular are:

1. Platform Independence
2. Security
3. Dynamic Binding
4. Good Performance
5. Multi Threading
6. Built-in Networking
7. Object Oriented
8. Compiler/Interpreter Combo
9. Robust
10. Several dangerous features of C & C++ eliminated:
11. Automatic Memory Management

5.1.3 OPERATING SYSTEM

Our developed system can be used in any operating system environment for example windows and Linux, Mach.

Hardware requirements

The following are hardware requirements:

- Any Intel based computer especially, P-IV with a 256 MB RAM recommended.
- Monitor (monochrome or colored).
- Hard disk having capacity of 40 GB or greater.
- Ethernet Network Interface Cards
- Connectors
- Cables

5.2 System Flow Chart Diagram:

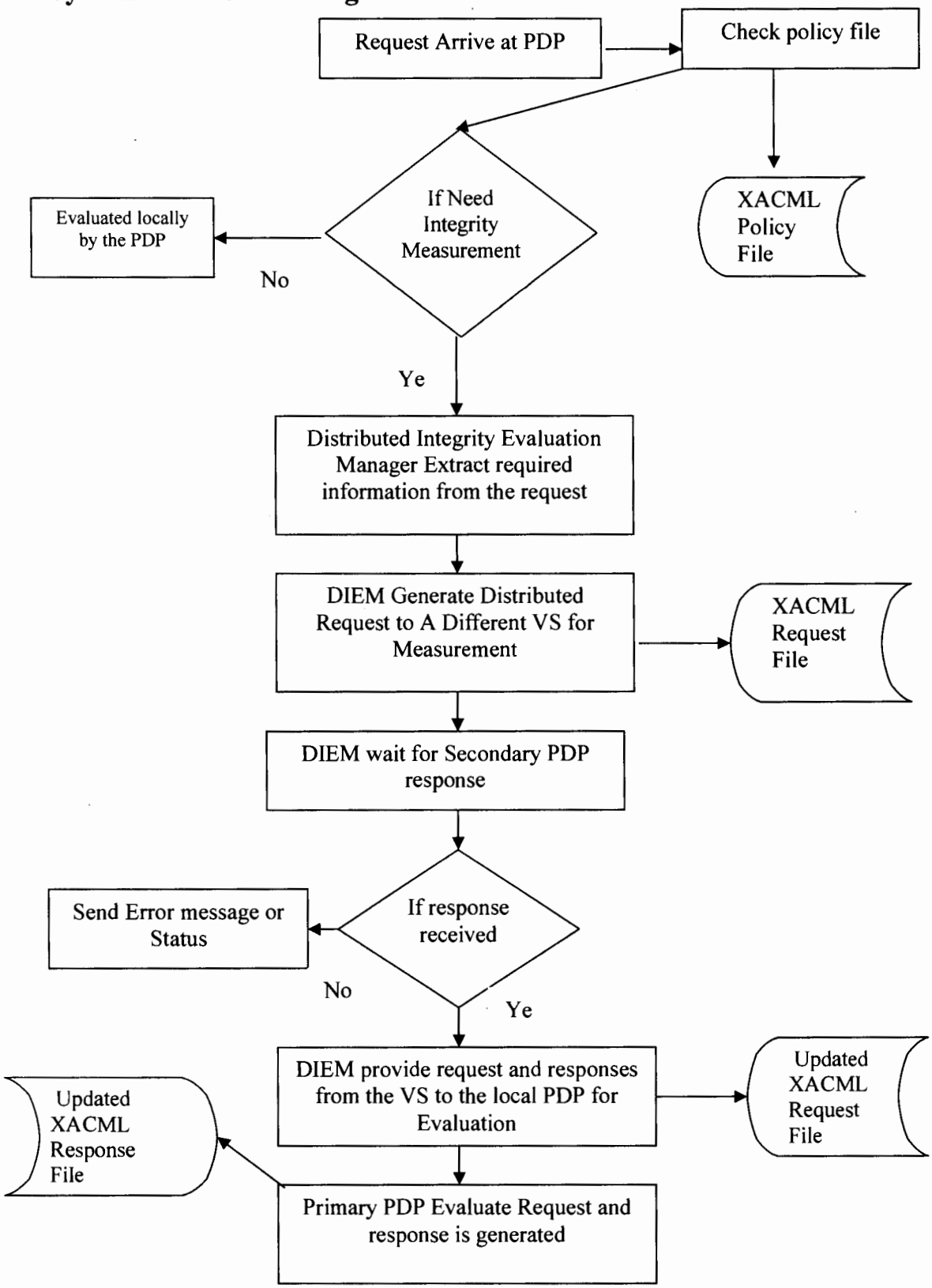


Figure 5: System Flow Chart Diagram for DIEM

5.3 Algorithm / Pseudo code

The pseudo code of the Service Provider and Validation Server Distributed Integrity Evaluation Manager in our developed distributed integrity measurement architecture is as;

Service Provider

```
{
Wait for request of SR
If (request received)
Parse the request file
Check policy if need integrity measurement
Then jump to 1:
OR
Jump to 2:
1:
        {
        Send request to Distributed Integrity Evaluation Manager
        Wait for the response of DIEM
        }

2:

    {
    Evaluate the request locally
    Send response to SR

    }

}
```

DIEM (Distributed Integrity Evaluation Manager)

```
{
Parse the policy file
Extract the required information from the request
Send distributed request to different Validation Servers
Wait for response
If(Response arrive)
Send response to PDP

        }
```

Validation Server

{

1. Calculate 10th PCR value by using hashes in the SML.
2. Match the expected PCR with the value supplied by the SR.
3. Check the nonce supplied by the SR with the nonce sent to it.
4. If the expected PCR and nonce is valid

Then jump to 1:
OR
Jump to 2:

1:

{
Access the database in which good known hashes is saved
Check whether hash is valid
Distributed Integrity Evaluation Manager
Send response to Distributed Integrity Evaluation Manager
}

2:

{
Create response for the Distributed Integrity Evaluation Manager
Send response to Distributed Integrity Evaluation Manager
}

}

TH-6507.

5.3.1 Classes and their Method:

Following classes are developed and used in the proposed system.

Class ValidationServer

Methods:

startServer() Creates and start an object of ValidationServer class. This will be used to start the validation server that starts listing requests from clients.

parseRequestFile() Method that parse the XACML request received from the client and create the required objects for parsing the request received from the clients.

performIntegrityMeasurement() This method basically perform actual integrity measurement that involves the SML and PCR verification checking and validating of hashes supplied by the integrity measurement requester and preparing response that will be sent to the service provider.

recievefile() This method is used for creating sockets and receiving request files from the client requesting for integrity measurement

sendfile () This method is used for creating sockets and sending response after processing the request of the clients for integrity measurement

Class ValidationRequester

Methods:

generateRequest() This method is used to Creates and start an object of ValidationRequester class. This will be used to generate request for integrity measurement that will be sent to different validation servers.

distributedIntegrityEvaluationManager() This is the main method that will using policy file generates distributed request to different validation servers and after receiving response from the validation servers provide to the PDP for evaluating request and generating response.

5.4 Summary

This chapter mainly focuses on the implementation phase of the proposed research project. In this chapter I have listed the tools and technologies used for developing the proposed system. The system flow chart diagram illustrates the flow of data in the system. The pseudo code and algorithm of the system is also explained in depth. Finally classes and method involved in the implementation of this research project is listed. The next chapter discusses and explains the testing and performance evaluation of the proposed system.

6. Testing and Performance Evaluation

Chapter 6

Testing and Performance Evaluation

We have used a Dell PowerEdge 2950 Server (Intel Xeon Quad Core). The server has two 3.0 GHz processors and 4GB memory. A normal validation time for single request is 1.422 second that is basically includes measurement correctness process of the SR platform by measuring SML and values of PCR that is signed by the TPM. The VS asks the SR for attestation of required parameters and nonce (to verify freshness of the parameters returned).

VS also ask the TPM of the SR to supply a quote over the value of its current tenth PCR. The PCRs value is attested by TPM using quote. By taking index of PCRs TPM takes quote and also take nonce for assuring the freshness of AIK and quote is used for signing. TPM digitally sign result by concatenating current PCR value and nonce. The result is known as TPM quote over a PCR value. Upon receiving this quote from the SR, the VS check Measurement and digital signature provided to verify that it has indeed been signed by the TPM of the client or SR. Afterwards, the expected value of the PCR is calculated by using the SML.

The algorithm used in this computation is given below.

1. Calculate the expected value of the 10th PCR value by accumulating the hashes in the SML.
2. Match the expected value of the PCR that is computed with the value supplied by the SR platform.
3. Match and check the nonce supplied by the SR platform with the nonce sent to it.

If the expected value of the PCR and those supplied by the SR is same, if the nonce same is returned in the quote, the VS can conclude that:

- SML supplied by the client is indeed valid as it is signed by TPM
- The values of the SML are fresh because TPM has signed the nonce while performing the quote.

Since the validity and freshness of the software hash is guaranteed with the help of TPM signature, the only condition for finding out the integrity of the software is that the hash be of a known good value. As SHA-1 generates hash of the specific application that is not reversible and the hash provided of the client platform corresponds to a known good value already present in the database. The integrity of the application which is running on client platform is certified by validation service which is running on a client platform.

6.1 Test Scenario

According to proposed architectural attestation models I have used delegated model in which attestation is carried out by validation service on behalf of service provider, when attestation requestor asks for it and result is sent in the form of credentials. The improvements which are observed by applying this model are:

1) The privacy of the platform which requires attestation is protected.

2) For the validation of attestation, attestation requestor did not have the capability.

In distributed and heterogeneous environment the most important model for attestation is delegating model.

6.2 Performance and Evaluation

Our proposed system has following advantages over the previous work.

1. Performance is improved
2. Privacy issue is solved
3. proposed architecture is scalable
4. Improved response time

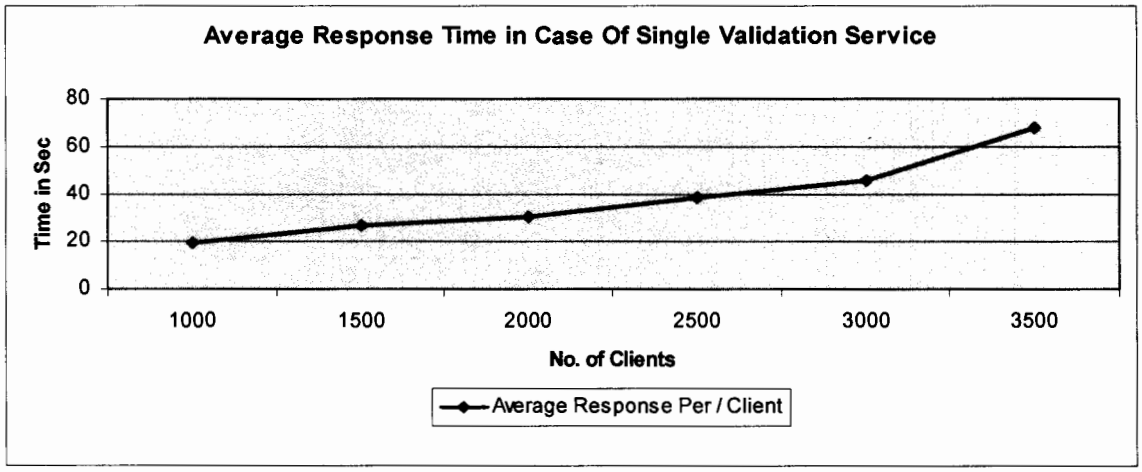


Figure 6: Average Response Time in case of single VS

First we have evaluated the performance of the system using single VS

In this case we generated 1000, 1500, 2000,2500,3000,3500 validation requests to server. We can analyze the average response time of the server .by analyzing the results we can easily conclude that as no of clients sending request for the validation is increasing as a result average response time is also increasing.

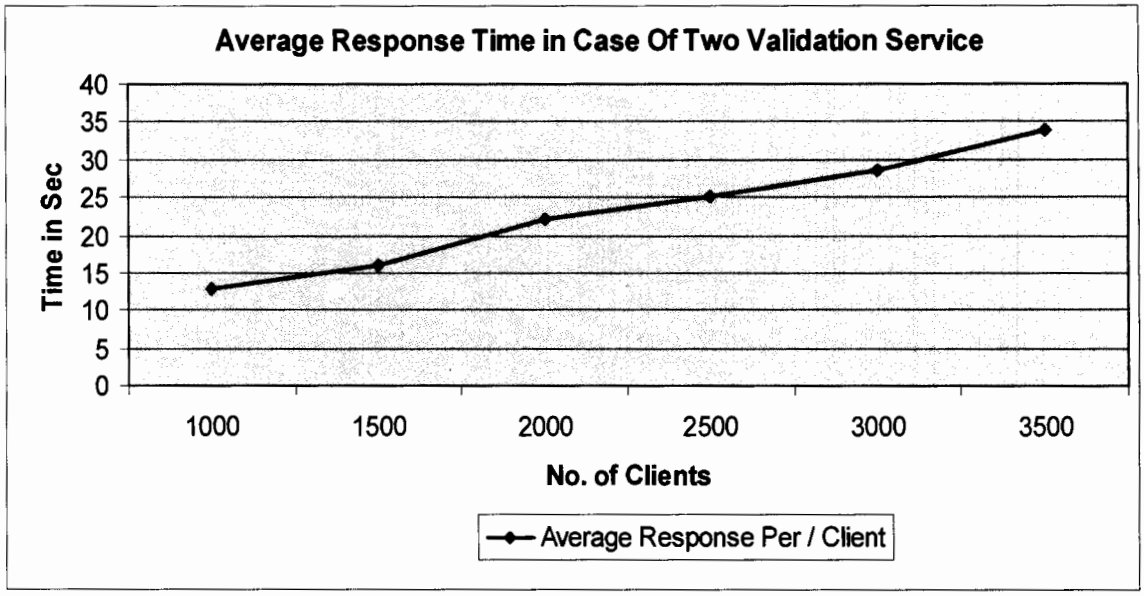


Figure 7: Average Response Time in case of two VS

In the second scenario, we have evaluated the performance of the system using two VS. In this case we generated distributed request for two different server means when we generated 1000 request then half of the request to validation server 1 and validation server 2. in this manner we have generated 1500, 2000,2500,3000,3500 validation requests that distributed to validation server 1 and validation server 2. We can analyze the average response time in case when requests are distributed between two servers .we can easily conclude that the response time in case of distributed VS is improved.

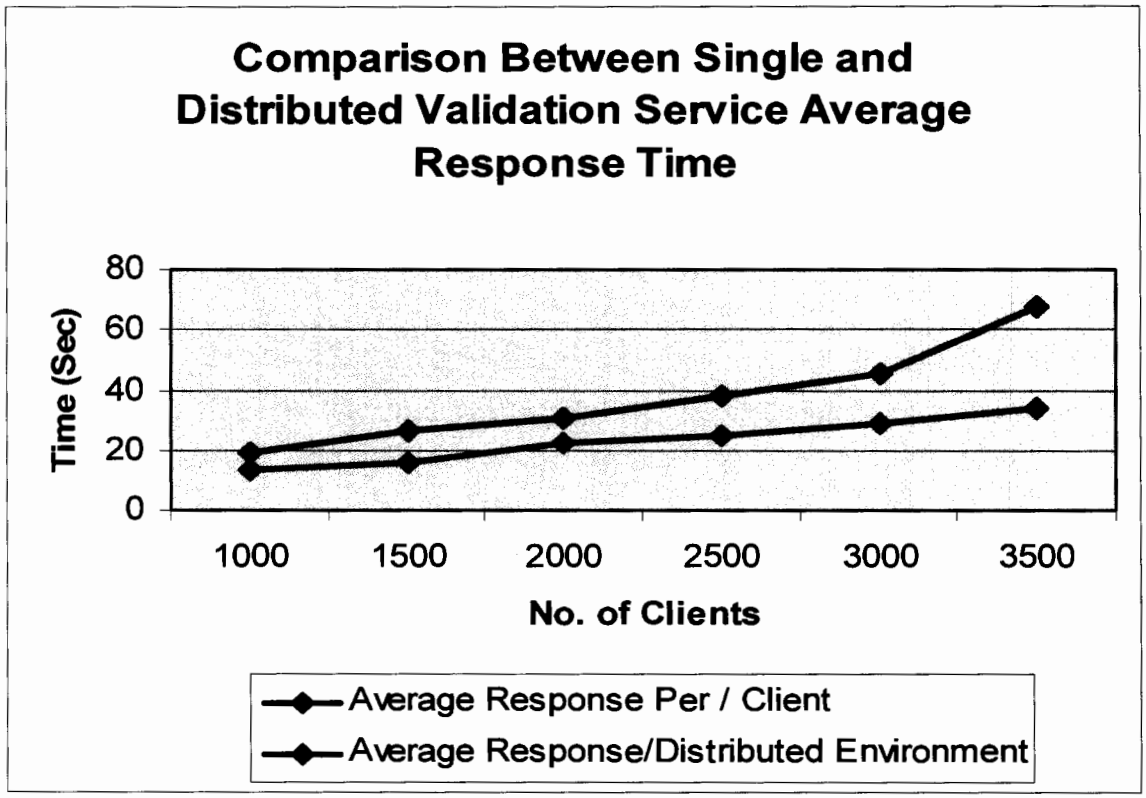


Figure 8: comparison between single and distributed VS average response time

The above graph shows comparison between averages responses time of in case of single VS its comparison with average response time in distributed environment. The graph shows that proposed distributed architecture has improved response time incredibly.

6.3 Summary

This chapter focuses on the testing and performance evaluation of the proposed distributed architecture. First it briefly explained the working environment of the proposed architecture. Then test scenario is mentioned and at the end results of both in case of single VS is compared with proposed distributed architecture and theses results are expressed in form of graphs. In the next chapter we give the concluding remarks and finally proposed some future recommendations.

7. Conclusion and Outlook

Chapter 7

7. Conclusion and Outlook

Remote Attestation is comparatively new domain of research. Remote Attestation helps authorized party to affirm that trustworthy environment exists on client platform. To integrate this concept in Web Services domain, a novel technique WS-Attestation has been introduced .yet , WS Attestation only provide functional prototype upon which more fine-grained attestation mechanism can be design. Behavioral Attestation for Web Services is proposed, in which XACML is built on the top of WS-Attestation in the proposed Web Service based behavioral attestation, to enable more flexible remote attestation mechanism at services level. XACML behavior policy defines the expected behavior of a SR platform. As proposed, the Distributed Integrity Measurement Architecture in which XACML behavior policy is used to define the expected behavior of SR platform and use distributed architecture for attestation purposes.

7.1 Achievements

Our proposed will improve the performance of overall remote attestation mechanism as in the previous work done in WS-Attestation uses single VS architecture that has various shortcomings. Our distributed architecture uses different VSs for validating integrity of different domain. Because of the distributed architecture the proposed system is also scalable means the system will work in situation in which large no of clients is accessing the services of the validation server. the architecture proposed by the WS-Attestation also lead to the problem of privacy means if validating server is compromised the whole configuration information of the SR or client can be used by the attacker to attack the specific client, because when the attacker has the information about the operating system client is using and security application running and if the known operating has the vulnerability. So the client configuration information can be use by the attacker to plan the attack. In our proposed architecture we have distributed the task of validating the integrity of different application running on the client machine to different VS as a result

the whole configuration information of the client will not disclose to the attacker if one of validation server is compromised.

7.2 Improvements

Following improvements have been observed

1. Performance will be improve
2. Privacy issue will be solved
3. Proposed architecture will be scalable
4. Response time of overall validation process will be improve

7.3 Future Recommendations/Outlook

In this thesis we propose a novel distributed integrity measurement architecture for the remote attestation. Our proposed architecture will solve the problem in the WS-Attestation architecture. Those were the privacy, performance and scalability problems as discussed in the previous chapters.

Remote attestation is at present a hot research area. Researchers are working to develop the better techniques of measuring, communicating and verifying the integrity of a remote platform. I have developed the prototype of the distributed integrity measurement architecture and we are working to enhance the architecture which involves the development of the some sort of discovery protocol for the discovery of validations servers for some specific domains and there is also a need of load balancing mechanism if we have no of validations services for integrity measurement of one specific domain.

7.4 Summary

This chapter contains the final concluding remarks of the research work. In first section, we briefly describe the need of the new architecture that should solve the problems exist in the previous WS-Attestation architecture. We proposed new paradigm which will solve the problem that exist in the previous architecture. In the next section we states achievements which we obtain from the proposed Architecture. Finally we propose a number of recommendations and directions which require more concentration in the future.

References and Bibliography

[1] AOL/NCSA Online Safety Study. http://www.staysafe on- line.info/news/ NCSA-AOLInome StudyRelease.pdf.

[2] IAIK at Graz University of Technology. http://iaik.tugraz.at/.

[3] IAIK: Institute for Applied Information Processing and Communications, Graz University of Technology. http://www. iaik. tugraz. at/.

[4] Jetty-Java-based Open SourceWeb Server. http:// www.mortbay.org/jetty-6/.

[5] Netscape, SSL 3.0 Speci_cation. . http://www .netscape.com/ eng/ssl3.

[6] Secure Hash Algorithm 1. http://www.ietf.org/rfc/rfc3174.txt.

[7] Security Assertion Markup Language (SAML) v2.0. http://docs.oasis-open.org/ security/saml/v2.0/ saml-core-2.0-os.pdf.

[8] Symantec Internet Security Threat Report, 2008. http://eval.symantec.com/ mktginfo/ enterprise/white-papers/b-whitepaper-internet -security-threat -report-xiii-04- 2008.en-us.pdf.

[9] Symantec Security Response. http://www. symantec.com/ security-response/ writeup.jsp? docid=2006- 111201-3853-99.

[10] Symantec Security Response (Silentbanker).http://www.symantec.com/ security-response/writeup.jsp?docid=2007-121718-1009-99)

[11] The Open Trusted Computing (OpenTC) consortium. http://www.opentc.net/.

[12] Trusted Computing for the Java(tm) Platform. available at,. http://trustedjava. sourceforge. net/.

[13] Trusted Computing Group (TCG). https:// www.trusted computing-group .org/.

[14] Trusted Platform Module (TPM) Specifications. https://www. trust-edcomputing group.org/ specs/TPM/.

[15] M. Alam, X. Zhang, M. Nauman, T. Ali, and J.P. Seifert.Model-based Behavioral Attestation. In Accepted for publication in SACMAT '08: Proceedings of the thirteenth ACM symposium on Access control models and technologies. Available at:http://serg.imsciences.edu.pk/pub/mba-sacmat08.pdf, New York, NY, USA, 2008. ACM Press.

[16] Masoom Alam, Qi Li, Xinwen Zhang, and Jean-Pierre Seifert. Usage control platformization via trustworthy selinux. In ASIACCS'08:Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security, 2008.

[17] Masoom Alam, Jean-Pierre Seifert, and Xinwen Zhang. A model-driven framework for trusted computing based systems. In EDOC '07:Proceedings of the 11th IEEE International Enterprise Distributed Object Computing Conference, page 75, Washington, DC, USA, 2007.IEEE Computer Society.

[18] Ryan Catherman David Safford Leendert van Doorn David Challene, Kent Yoder. A Practical Guide to Trusted Computing. IBM Press, IBM T. J Watson Labs, USA, 2008.

[19] Mimi Zohar David Safford. A trusted linux client. IBM Journal, 2005.

[20] Rick Kennell and Leah H. Jamieson. Establishing the genuinity of remote computer systems. In SSYM'03: Proceedings of the 12th conference on USENIX Security Symposium, pages 21{21, Berkeley, CA,USA, 2003. USENIX Association.

[21] Yoshihama Sachiko Ebringer T Maruyama Hirosh, Muneto Seiji. Tpod-trusted platform on demand. Joho Shori Gakkai Kenkyu Hokoku, pages 181{186, August 2004.

[22] F. Mayer, K. MacMillan, and D. Caplan. SELinux by Example: Using Security Enhanced Linux. Prentice Hall, 2006.

[23] Megumi Nakamura Sachiko Yoshihama, Tim Ebringer. Test and Analysis of Web Services. Springer Verlag, Springer Berlin Heidelberg, 2007.

[24] David Safford. Need for tcpa, March 2002.

[25] Reiner Sailer, Xiaolan Zhang, Trent Jaeger, and Leendert van Doorn. Design and implementation of a tcg-based integrity measurement architecture. In SSYM'04: Proceedings of the 13th conference on USENIX Security Symposium, pages 16{28, Berkeley, CA, USA, 2004. USENIX Association.

[26] Elaine Shi, Adrian Perrig, and Leendert Van Doorn. BIND: A Fine-Grained Attestation Service for Secure Distributed Systems. In SP '05: Proceedings of the

2005 IEEE Symposium on Security and Privacy, pages 154{168, Washington, DC, USA, 2005. IEEE Computer Society.

[27] Zishuang (Eileen) Ye, Sean Smith, and Denise Anthony. Trusted paths for browsers. ACM Trans. Inf. Syst. Secur., 8(2):153{186, 2005.

[28] S. Yoshihama, T. Ebringer, M. Nakamura, S. Munetoh, T. Mishina, and H. Maruyama. WS-Attestation: Enabling Trusted Computing on Web Services. Test and Analysis of Web Services, pages 441{469, 2007.

[29] M Alam, X Zhang, M Nauman, T Ali. Behavioral Attestation for Web Services (BA4WS)- Proceedings of the 2008 ACM workshop on Secure web services, Alexandria, Virginia, USA Pages 21-28 2008

[30] Markus Lorch, Seth Proctor, Rebekah Lepro, Dennis Kafura, and Sumit Shah. First experiences using xacml for access control in distributed systems. In XMLSEC '03: Proceedings of the 2003 ACM workshop on XML security, pages 25–37, New York, NY, USA, 2003. ACM.

[31] Seth Schoen. Trusted Computing: Promise and Risk. http://www.eff.org/Infrastructure/trusted computing/20031001 tc.php.

[32] Ahmad-Reza Sadeghi and Christian Property-based Attestation for Computing Platforms: Caring about Properties, not Mechanisms. In *NSPW '04: Proceedings of the 2004 Workshop on New Security Paradigms*, pages 67–77, New York, NY, USA, 2004. ACM Press.

[33] Xiao-Yong Li, Chang xiang Shen, and Xiao-Dong Zuo. An Efficient Attestation for Trustworthiness of Computing Platform. In *IIH-MSP*, pages 625–630, 2006.

Acronyms

AIKs	**Attestation Identity Keys**
BIND	**Binding Instructions and Data**
BIOS	**Basic Input Output System**
EK	**Endorsement Key**
BWA	**Broadband Wireless Access**
DBMS	**Database Management System**
DIEM	**Distributed Integrity Evaluation Manager**
IMA	**Integrity Measurement Architecture**
IEEE	**Institute of Electrical and Electronic Engineers**
MAN	**Metropolitan Area Network**
Mb/s	**Megabit Per Second**
PCRs	**Platform Configuration Registers**
PDA	**Personal Digital Assistant**
PKI	**Public Key Infrastructure,**
PKM	**Privacy and Key Management**
PKMv2	**Privacy and Key Management Version 2**
PPP	**Point-To-Point Protocol**
SML	**Stored Measurement Log**
SR	**Service Requester**
SSL	**Secure Socket Layer**
TC	**Trusted Computing**

TCG	**Trusted Computing Group**
TPM	**Trusted Platform Module**
TPoD	**Trusted Platform on Demand**
TDMA	**Time Division Multiple Access**
VPN	**Virtual Private Network**
VS	**Validation Service**
VR	**Validation Requester**
WMAN	**Wireless Metropolitan Area Networks**
XACML	**Extensible Access Control Markup Language**