

Mälardalen University Press Dissertations
No. 161

BRINGING VISIBILITY IN THE CLOUDS
USING SECURITY, TRANSPARENCY AND ASSURANCE SERVICES

Mudassar Aslam

2014



School of Innovation, Design and Engineering

Copyright © Mudassar Aslam, 2014
ISBN 978-91-7485-156-4
ISSN 1651-4238
Printed by Arkitektkopia, Västerås, Sweden

Mälardalen University Press Dissertations
No. 161

BRINGING VISIBILITY IN THE CLOUDS
USING SECURITY, TRANSPARENCY AND ASSURANCE SERVICES

Mudassar Aslam

Akademisk avhandling

som för avläggande av teknologie doktorsexamen i datavetenskap vid
Akademin för innovation, design och teknik kommer att offentligen försvaras
fredagen den 5 september 2014, 10.00 i Kappa, Mälardalen University, Västerås.

Fakultetsopponent: Professor Chris Mitchell, Royal Holloway, University of London



Akademin för innovation, design och teknik

Abstract

The evolution of cloud computing allows the provisioning of IT resources over the Internet and promises many benefits for both - the service users and providers. Despite various benefits offered by cloud based services, many users hesitate in moving their IT systems to the cloud mainly due to many new security problems introduced by cloud environments. In fact, the characteristics of cloud computing become basis of new problems, for example, support of third party hosting introduces loss of user control on the hardware; similarly, on-demand availability requires reliance on complex and possibly insecure API interfaces; seamless scalability relies on the use of sub-providers; global access over public Internet exposes to broader attack surface; and use of shared resources for better resource utilization introduces isolation problems in a multi-tenant environment. These new security issues in addition to existing security challenges (that exist in today's classic IT environments) become major reasons for the lack of user trust in cloud based services categorized in Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS) or Infrastructure-as-a-Service (IaaS).

The focus of this thesis is on IaaS model which allows users to lease IT resources (e.g. computing power, memory, storage, etc.) from a public cloud to create Virtual Machine (VM) instances. The public cloud deployment model considered in this thesis exhibits most elasticity (i.e. degree of freedom to lease/release IT resources according to user demand) but is least secure as compared to private or hybrid models. As a result, public clouds are not trusted for many use cases which involve processing of security critical data such as health records, financial data, government data, etc. However, public IaaS clouds can also be made trustworthy and viable for these use cases by providing better transparency and security assurance services for the user. In this thesis, we consider such assurance services and identify security aspects which are important for making public clouds trustworthy. Based upon our findings, we propose solutions which promise to improve cloud transparency thereby realizing trustworthy clouds.

The solutions presented in this thesis mainly deal with the secure life cycle management of the user VM which include protocols and their implementation for secure VM launch and migration. The VM launch and migration solutions ensure that the user VM is always hosted on correct cloud platforms which are setup according to a profile that fulfills the use case relevant security requirements. This is done by using an automated platform security audit and certification mechanism which uses trusted computing and security automation techniques in an integrated solution. In addition to provide the assurance about the cloud platforms, we also propose a solution which provides assurance about the placement of user data in correct and approved geographical locations which is critical from many legal aspects and usually an important requirement of the user. Finally, the assurance solutions provided in this thesis increase cloud transparency which is important for user trust and to realize trustworthy clouds.

SICS Swedish ICT
Doctoral Thesis
SICS Dissertation Series 70

Bringing Visibility in the Clouds
using Security, Transparency and Assurance Services

Mudassar Aslam

2014



Swedish Institute of Computer Science(SICS)
SICS Swedish ICT, Kista
Stockholm, Sweden

Copyright © Mudassar Aslam, 2014

ISSN 1101-1335

Printed by Mälardalen University, Västerås, Sweden

Abstract

The evolution of cloud computing allows the provisioning of IT resources over the Internet and promises many benefits for both - the service users and providers. Despite various benefits offered by cloud based services, many users hesitate in moving their IT systems to the cloud mainly due to many new security problems introduced by cloud environments. In fact, the characteristics of cloud computing become basis of new problems, for example, support of *third party hosting* introduces loss of user control on the hardware; similarly, *on-demand availability* requires reliance on complex and possibly insecure API interfaces; seamless *scalability* relies on the use of sub-providers; *global access* over public Internet exposes to broader attack surface; and use of *shared resources* for better resource utilization introduces isolation problems in a multi-tenant environment. These new security issues in addition to existing security challenges (that exist in today's classic IT environments) become major reasons for the lack of user trust in cloud based services categorized in Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS) or Infrastructure-as-a-Service (IaaS).

The focus of this thesis is on IaaS model which allows users to lease IT resources (e.g. computing power, memory, storage, etc.) from a *public* cloud to create Virtual Machine (VM) instances. The public cloud deployment model considered in this thesis exhibits most *elasticity* (i.e. degree of freedom to lease/release IT resources according to user demand) but is least secure as compared to *private* or *hybrid* models. As a result, public clouds are not trusted for many use cases which involve processing of security critical data such as health records, financial data, government data, etc. However, public IaaS clouds can also be made trustworthy and viable for these use cases by providing better transparency and security assurance services for the user. In this thesis, we

consider such assurance services and identify security aspects which are important for making public clouds trustworthy. Based upon our findings, we propose solutions which promise to improve cloud transparency thereby realizing trustworthy clouds.

The solutions presented in this thesis mainly deal with the secure life cycle management of the user VM which include protocols and their implementation for secure *VM launch* and *migration*. The VM launch and migration solutions ensure that the user VM is always hosted on correct cloud platforms which are setup according to a profile that fulfills the use case relevant security requirements. This is done by using an automated platform security audit and certification mechanism which uses trusted computing and security automation techniques in an integrated solution. In addition to provide the assurance about the cloud platforms, we also propose a solution which provides assurance about the placement of user data in correct and approved geographical locations which is critical from many legal aspects and usually an important requirement of the user. Finally, the assurance solutions provided in this thesis increase cloud transparency which is important for user trust and to realize *trustworthy clouds*.

Populärvetenskaplig sammanfattning

Utvecklingen av cloud computing tillåter användning av IT-resurser över Internet och kan innebära många fördelar för både användare och leverantörer. Trots fördelarna med molnbaserade tjänster tvekar många användare att flytta sina IT-system till molnet främst på grund av många nya säkerhetsproblem som tillkommer i molnmiljöer. Faktum är att egenskaperna hos cloud computing blir grunden för nya problem, till exempel introducerar stöd för *tredjeparts hosting* förlust av användarens kontroll över hårdvaran; på samma sätt förutsätter *on-demand tillgänglighet* komplicerade och eventuellt osäkra API-gränssnitt; *sömnlös skalbarhet* förlitar sig på användning av underleverantörer; *global tillgång* via det publika Internet exponerar en bredare attackyta; och användning av *delade resurser* för bättre resursutnyttjande introducerar isoleringsproblem i en miljö med flera molnkunder. Dessa nya säkerhetsfrågor, utöver de befintliga säkerhetsutmaningar som redan finns i dagens IT-miljöer, är viktiga skäl till användarens brist på förtroende för molnbaserade tjänster, kategoriserade i Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS) respektive Infrastructure-as-a-Service (IaaS).

Fokus i denna avhandling är på IaaS-modellen, vilken tillåter användare att hyra IT-resurser (t.ex. datorkraft, minne, lagring, etc.) från ett *publikt* moln för att skapa instanser av virtuella maskiner (VM). Den publika molndistributionsmodellen som betraktas i denna avhandling uppvisar störst *elasticitet* (dvs grad av frihet att hyra / frigöra IT-resurser i enlighet med användarnas efterfrågan) men är minst säker jämfört med *privata* eller *hybrid*-modeller. Som ett resultat är publika moln

inte pålitliga för de många användningsfall vilka innebär behandling av säkerhetskritiska uppgifter såsom patientjournaler, finansiella data, offentlig statistik, etc. Däremot kan publika IaaS-moln göras pålitliga och lönsamma för användning genom bättre insyn och säkerhetsgarantier för användaren. I denna avhandling identifierar vi sådana säkerhetsaspekter som är viktiga för att göra publika moln pålitliga. Baserat på våra resultat föreslår vi lösningar som förbättrar insyn i molnet och därmed möjliggör pålitliga publika moln.

De lösningar som presenteras i denna avhandling behandlar främst säker livscykelhantering för användarens VM, vilket innefattar protokoll och dess tillämpning för säker *VM-start* och *VM-migrering*. Lösningar för VM-start och VM-migrering säkerställer att användarens VM alltid befinner sig på rätt molnplattform och att denna är inställd enligt en profil som uppfyller användningsfallets relevanta säkerhetskrav. Detta görs med hjälp av en automatiserad säkerhetsgranskning av plattformen och ett certifieringssystem som använder Trusted Computing och tekniker för säkerhetsautomation i en integrerad lösning. Förutom att ge en försäkran om molnplattformar, föreslår vi också en lösning som ger säkerhet vad gäller placeringen av användardata i korrekta och godkända geografiska platser, vilket är viktigt ut flera juridiska aspekter och ofta ett viktigt krav från användaren. Slutligen, de lösningar som ges i denna avhandling ökar molnets öppenhet, vilket är viktigt för användarens förtroende och vilket möjliggör *pålitliga moln*.

Acknowledgments

First of all, I am really thankful to **Almighty Allah** who gave me perseverance, knowledge and strength to achieve this milestone. I pray Him to make my knowledge beneficial for others and fulfill my responsibilities that lie on me due to His blessings.

I am grateful to all people in SICS, MDH and Ericsson who supported and guided me in doing this work; especially, my co-supervisor Dr. **Christian Gehrman** who provided me the opportunity to work in an esteemed research environment at SICS. I am indebted to all the efforts and valuable time that Christian has spent on me for guiding, improving and polishing my research skills right from the very first day. I also want to express my sincere regards and gratitude for my main supervisor Prof. **Mats Björkman** who provided me the much needed motivation, inspiration and guidance in achieving this milestone.

I feel happy, satisfied and proud to get the opportunity to work with the learned researchers from SICS and Ericsson who provided very useful feedback to improve my work and tune it according to the current and future industrial demands. I express my gratitude to **András Méhes** who provided his insightful criticism to remove the lacunae in the early stages of this work; **Lars Rasmusson**, **Fredric Morenius** and **Nicolae Paladi** for their collaborative research and development activities; and **Rolf Blom** for his useful research directions.

I am really thankful to all my current and former co-workers in Security (SEC) and Networks (NETS) lab: **Antonis Michalas**, **Marco Tiloca**, **Ludwig Seitz**, **Arash Vahidi**, **Anders Lindgren**, **Bengt Ahlgren**, **Björn Grönvall**, **Henrik Abrahamsson**, **Ian Marsh**, **Laura Feeney** and **Maria Holm** who provided a unique professional and research environment for me. I would specially like to thank **Oliver Schwarz** for his discussions (technical and social) and valuable suggestions whenever solicited.

Finally, I would like to thank all my friends and colleagues including Shahzad Saleem, Shahid Raza, Rashdan, Zeeshan Ali Shah and many others who were always there to extend their help and support in times of need.

I would like to dedicate this work to **my parents** and family who supported me throughout my academic and professional carrier with their prayers, love, guidance and sacrifices whenever required.

Mudassar Aslam
Stockholm, September, 2014

*This work has been performed in the Security Lab in SICS Swedish ICT. Other partners that were involved in various projects include Ericsson, Saab, Telia-Sonera and T2Data. The funding for this work has mainly been provided by **VINNOVA** through different research project grants DNr: 2009-02959, DNr: 2010-02098, DNr: 2012-01519; and also by the **Higher Education Commission (HEC), Pakistan** in the form of scholarship grant for my PhD studies (PM-2007-1/Overseas Phase-II/Sweden/231).*

The SICS Swedish ICT is jointly sponsored by the Swedish government and the Industry partners which include TeliaSonera, Ericsson, Saab AB, FMV (Defense Materiel Administration), Green Cargo (Swedish freight railway operator), ABB, and Bombardier Transportation.

List of Publications

Papers included in the thesis¹

1. Mudassar Aslam, Christian Gehrman. *Security Considerations for Virtual Platform Provisioning*. In 10th European Conference on Information Warfare and Security (ECIW), 7-8 July 2011, Tallin, Estonia.
2. Mudassar Aslam, Christian Gehrman, Lars Rasmusson, Mats Björkman. *Securely Launching Virtual Machines on Trustworthy Platforms in a Public Cloud*. In 2nd International Conference on Cloud Computing and Services Science (CLOSER), 18-21 April 2012, Porto, Portugal.
3. Nicolae Paladi, Christian Gehrman, Mudassar Aslam, Fredric Morenius. *Trusted Launch of Virtual Machine Instances in Public IaaS Environments*. In 15th Annual International Conference on Information Security and Cryptology (ICISC), 28-30 Nov 2012, Seoul, Korea.
4. Mudassar Aslam, Christian Gehrman, Mats Björkman. *Security and Trust Preserving VM Migrations in Public Clouds*. In 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 25-27 June 2012, Liverpool, UK.
5. Mudassar Aslam, Christian Gehrman, Mats Björkman. *ASArP: Automated Security Assessment & Audit of Remote Platforms -*

¹The included papers have been reformatted to comply with the thesis layout

using *TCG-SCAP synergies*. In Journal of Information Security and Applications (to appear).

6. Nicolae Paladi, Mudassar Aslam, Christian Gehrman. *Trusted Geolocation Aware Data Placement in Infrastructure Clouds*. 13th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 2014.

Additional publications, not included in the thesis

1. Mudassar Aslam, Christian Gehrman, Mats Björkman. *Continuous Security Evaluation and Auditing of Remote Platforms by Combining Trusted Computing and Security Automation Techniques*. In: Proceedings of the 6th International Conference on Security of Information and Networks (SIN), pages 136-143, New York, NY, USA, 2013. ACM. **BEST PAPER AWARD**
2. Lars Rasmusson, Mudassar Aslam. *Protecting Private Data in the Cloud*. In: The 2nd International Conference on Cloud Computing and Services Science (CLOSER), 18-21 April 2012, Porto, Portugal.
3. Mudassar Aslam. *Security Considerations for Virtual Platform Provisioning*. In: Workshop on Cryptography and Security in Clouds, March 15-16, 2011, Zurich, Switzerland.
4. Mudassar Aslam, Christian Gehrman. *TCG Based Approach for Secure Management of Virtualized Platforms: state-of-the-art*. SICS Technical Report (T2010:05), 2010, ISSN No. 1100-3154. Available at <http://soda.swedish-ict.se/3993/>
5. Mudassar Aslam, Christian Gehrman. *Deploying Virtual Machines on Shared Platforms*. SICS Technical Report (T2011:07), 2011, ISSN No. 1100-3154. Available at <http://soda.swedish-ict.se/4170/>

List of Acronyms

<i>AIK</i>	Attestation Identity Key
<i>API</i>	Application Programming Interface
<i>CAPEX</i>	Capital Expenditure
<i>CCE</i>	Common Configuration Enumeration
<i>CCSS</i>	Common Configuration Scoring System
<i>Client</i>	See <i>User</i>
<i>CPE</i>	Common Platform Enumeration
<i>CSA</i>	Cloud Security Alliance
<i>CSP</i>	Cloud Service Provider
<i>CVE</i>	Common Vulnerabilities and Exposures
<i>CVSS</i>	Common Vulnerability Scoring System
<i>EK</i>	Endorsement Key
<i>FedRAMP</i>	Federal Risk and Authorization Management Program
<i>FISMA</i>	Federal Information Security Management Act
<i>GPS</i>	Global Positioning System
<i>GuestOS</i>	Guest Operating System
<i>HIPAA</i>	Health Insurance Portability and Accountability Act

<i>IaaS</i>	Infrastructure-as-a-Service
<i>NIST</i>	National Institute of Standards and Technology
<i>NVD</i>	National Vulnerability Database
<i>OVAL</i>	Open Vulnerability and Assessment Language
<i>PaaS</i>	Platform-as-a-Service
<i>PCA</i>	Privacy CA
<i>PCI – DSS</i>	Payment Card Industry Data Security Standard
<i>PCR</i>	Platform Configuration Registers
<i>Provider</i>	Cloud Service Provider
<i>PTAA</i>	Platform Trust Assurance Authority
<i>SaaS</i>	Software-as-a-Service
<i>SCAP</i>	Security Content Automation Protocol
<i>SecaaS</i>	Security-as-a-Service
<i>SLA</i>	Service Level Agreement
<i>SRK</i>	Storage Root Key
<i>STAR</i>	Security, Trust and Assurance Registry
<i>TAL</i>	Trust Assurance Level
<i>TCG</i>	Trusted Computing Group
<i>TPM</i>	Trusted Platform Module
<i>TSPI</i>	TCG Service Provider Interface
<i>TSS</i>	TCG Software Stack
<i>TTP</i>	Trusted Third Party
<i>User</i>	Cloud Service User
<i>VM</i>	Virtual Machine
<i>VMM</i>	Virtual Machine Monitor
<i>XCCDF</i>	Extensible Configuration Checklist Description Format

Contents

I	Thesis	1
1	Introduction	3
1.1	Research Goals	4
1.2	Research Problem	5
1.3	Research Methodology	6
1.4	Contributions	7
1.5	Thesis Outline	10
2	Background	11
2.1	Cloud Computing	11
2.1.1	Service Models	12
2.1.2	Deployment Models	15
2.2	The Trusted Computing Group (TCG)	16
2.2.1	Trusted Platform Module (TPM)	17
2.2.2	TPM - Key Management	18
2.2.3	TPM Data Protection	20
2.2.4	Sealing Data Remotely	21
2.2.5	TPM 2.0	22
2.3	Security Content Automation Protocol (SCAP)	25
2.3.1	Languages	25
2.3.2	Enumerations	26
2.3.3	Measurement and Scoring Systems	26
3	Trustworthy Clouds - State-of-the-Art	27
3.1	State-of-the-Art	27
3.1.1	Trustworthy VM Launch and Migration	28
3.1.2	Cloud Transparency through Audits	29

3.2	State of Practice	31
3.2.1	FedRAMP	31
3.2.2	European Union Cloud Strategy	33
3.2.3	Security, Trust and Assurance Registry (STAR)	33
3.2.4	Relevance with our work	35
4	Putting the Pieces Together	37
4.1	Scenario	37
4.2	Cloud Trust Management Model	38
4.2.1	Define Cloud Platform Baseline	38
4.2.2	Setup Cloud Platforms	39
4.2.3	Cloud Platform Certification	40
4.2.4	Continuous Platform Audit & Certification	42
4.2.5	Trusted VM Life Cycle	42
5	Conclusions and Future Work	43
5.1	Conclusion	43
5.2	Future Work	45
5.2.1	Sealing Data to a Platform State	45
5.2.2	Implementation of a Standards-based Trustworthy Clouds	46
6	Overview of Papers	47
6.1	Security Considerations for Virtual Platform Provisioning	47
6.2	Securely Launching Virtual Machines on Trustworthy Plat- forms in a Public Cloud	49
6.3	Trusted Launch of Virtual Machine Instances in Public IaaS Environments	51
6.4	Security and Trust Preserving VM Migrations in Public Clouds	52
6.5	ASArP: Automated Security Assessment & Audit of Re- mote Platforms	53
6.6	Trusted Geolocation Aware Data Placement in Public In- frastructure Clouds	54
	Bibliography	57

II	Included Papers	65
7	Paper A: Security Considerations for Virtual Platform Provisioning	67
7.1	Introduction	69
7.2	Scenario - A Telecommunication Cloud Use Case	70
7.3	Threat and Security Requirements	73
7.3.1	Provider Network Authentication	74
7.3.2	Platform Integrity and Authentication	74
7.3.3	Authentication, Attestation and VM Launch Protocol	75
7.3.4	VM Isolation	75
7.3.5	Confidentiality	76
7.3.6	Secure VM Migration	76
7.4	Related Work	77
7.5	Conclusion	79
	Bibliography	81
8	Paper B: Securely Launching Virtual Machines on Trustworthy Platforms in a Public Cloud	85
8.1	Introduction	87
8.2	Scenario and Threats	88
8.2.1	Scenario	88
8.2.2	Threats	89
8.3	Related Work	90
8.4	Cloud Platform Security Architecture	92
8.4.1	Cloud Platform Architecture	92
8.4.2	Platform Credentials	94
8.4.3	Platform Secure Boot	95
8.5	Secure VM Launch Protocol	95
8.5.1	Connect and Discovery	97
8.5.2	Platform Integrity-Verification	97
8.5.3	VM Launch	98
8.6	Implementation	99
8.6.1	User-Provider Client	100
8.6.2	Procurement Server	100
8.6.3	Cloud Platform: Management Agent	101
8.7	Security Analysis	102
8.7.1	Authentication	103

8.7.2	Confidentiality	104
8.7.3	Integrity	104
8.7.4	Non-Repudiation	105
8.7.5	Replay Protection	105
8.8	Conclusion	105
	Bibliography	107
9	Paper C:	
	Trusted Launch of Virtual Machine Instances in Public IaaS Environments	111
9.1	Introduction	113
9.2	Trust and attack models, problem description and requirements	114
9.2.1	Trust and attack models	114
9.2.2	Virtual machine images	116
9.2.3	Requirements for a trusted VM launch protocol	116
9.3	A trusted launch protocol for virtual machine images in IaaS environments	117
9.3.1	Platform-agnostic protocol description	119
9.4	Protocol security analysis	123
9.4.1	<i>TTP</i> verification model	124
9.4.2	Protocol caveats	124
9.5	Protocol implementation	124
9.5.1	OpenStack IaaS platform	125
9.5.2	Prototype implementation	125
9.6	Related work	127
9.7	Conclusion	129
	Bibliography	131
10	Paper D:	
	Security and Trust Preserving VM Migrations in Public Clouds	135
10.1	Introduction	137
10.2	Background	138
10.2.1	VM Migration	139
10.3	Related Work	140
10.4	Attack Model and Security Requirements	142
10.4.1	Attack Model	142
10.4.2	Requirements Analysis	143

10.5 Proposed Trusted VM Migration	144
10.5.1 Platform Trust Certification	145
10.5.2 VM Migration Protocol	151
10.6 Requirements Review	152
10.7 Conclusion and Future Work	153
Bibliography	155

11 Paper E:

ASArP: Automated Security Assessment & Audit of Remote Platforms - using TCG-SCAP synergies	159
11.1 Introduction	161
11.2 Threat Model	163
11.2.1 Compromising Software Stack Integrity	163
11.2.2 Misconfiguration of the running software	164
11.3 Background	164
11.3.1 State of Practice	165
11.3.2 State of the Art	166
11.3.3 Limitations of SCAP and TCG	167
11.4 Automated Security Assessment and Auditing of Remote Platforms (<i>ASArP</i>)	168
11.4.1 Phase I - Software Stack Integrity	168
11.4.2 Phase II - Software Stack Vulnerability Assessment	169
11.4.3 Phase III - Software Configuration Compliance . .	171
11.5 Implementation	172
11.5.1 Environment Setup	173
11.5.2 ASArP Performance Analysis	175
11.6 Discussion	180
11.6.1 Security Analysis	180
11.6.2 Implementation Feasibility	182
11.7 Related Work	183
11.8 Conclusion and Future Work	185
Bibliography	187

12 Paper F:

Trusted Geolocation Aware Data Placement in Infrastructure Clouds	193
12.1 Introduction	195
12.1.1 Contribution	196
12.1.2 Organization	196

12.2	Background and related work	196
12.3	Preliminaries	199
12.3.1	Definitions	199
12.3.2	Adversary model	201
12.3.3	Problem Statement	202
12.4	System model	202
12.4.1	Distributed object storage	203
12.5	Protocol description	204
12.5.1	Geolocation	205
12.5.2	Storage Protection Protocol	208
12.6	Prototype implementation	211
12.7	Security Analysis	214
12.8	Performance	215
12.9	Conclusion	216
	Bibliography	219

I

Thesis

Chapter 1

Introduction

Traditionally, companies and organizations procure physical hardware resources to set up their IT infrastructure. However, the availability of multi-core, high performance computing platforms together with the advancements in virtualization technologies and high-speed data networks have created new possibilities for the procurement of IT resources usually known as *cloud services*. Buying a cloud service can be a quicker, more dynamic and cost effective way to run an IT service compared to using own hardware. Costs can also be saved due to less administration as well as energy savings. Well established cloud service categories are Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS) and Infrastructure-as-a-Service (IaaS) which are sometimes commonly called SPI service models [1]¹. A cloud service is not only limited to SPI service models rather Anything-as-a-Service (XaaS) can be offered, for example, Storage-as-a-Service, Database-as-a-Service, Security-as-a-Service, etc. We focus on security and trust related aspects in an Infrastructure-as-a-Service (IaaS) model in which a cloud service provider (CSP) uses platform virtualization by means of a hypervisor (e.g. XEN [2]) to create and run multiple virtual instances called Virtual Machines (VMs) for many clients on a single physical machine. The use of virtualization and its ability to create multiple VMs of different specifications (attached memory, storage and processing power etc.) on a single powerful physical machine allows maximum utilization of that machine by providing the service (a VM in IaaS case) to many cloud users simultaneously.

¹SPI models will be described in chapter 2 (§2.1)

Many possible cloud deployment models exist depending upon the nature and requirements of the cloud users. Well known examples are *Private*, *Public* and *Hybrid* clouds. A private cloud is usually deployed by an organization on premise and its services are provisioned to the users who belong to the same organization. On the other hand, a public cloud is setup by an independent CSP and its services are open for any individual or an organization. The hybrid cloud model integrates both public and private cloud models to overcome the shortcomings of both but involves various challenges to keep a balance [3]. The resource *elasticity* – which allows the cloud users to lease and release computing services dynamically and robustly on-demand without worrying about the up-front capital expenditure (CAPEX) – is one of the main benefits of cloud computing. However, the elasticity payoff varies inversely with the security requirements between different cloud deployment models, that is, the private clouds are most secure but least elastic whereas the public clouds are most elastic but least secure [4]. One of the main reasons that the public clouds are (or perceived to be) less secure, is the lack of user visibility in the cloud infrastructure, its operations and processes. Due to this shortcoming, a security sensitive cloud user (e.g. a public department, company or organization) can not trust a public cloud service. This thesis focuses on improving the visibility of the cloud infrastructure by proposing automated and continuous remote audit techniques which increase transparency thereby *bringing visibility in the clouds*.

1.1 Research Goals

The aim of this research is to propose technological solutions to:

“Increase User Trust in the Public Clouds Services”

with a special focus on IaaS cloud model. In order to achieve this broader goal, we define following sub-goals:

- G1. Identify security issues that inhibit the wide-scale adoption of public cloud services by the security sensitive users [5, 6] due to lack of technical ways for verifying the security claims stated by the cloud service providers (CSPs).
- G2. Propose new design principles, security protocols and models to address the identified problems thereby providing transparency and assurance services to increase user trust in the public clouds.
- G3. Implement prototype(s) of the proposed solutions to validate their feasibility and practicability.

1.2 Research Problem

The research problems focused in this thesis are drawn from the broader research goal stated above. The IaaS model considered in this thesis requires the user VM to run "*in the cloud*" on remote platforms in a CSP owned network. The remote cloud platforms are managed by the CSP who is also responsible for their security protection. Traditionally, the users are assured about the employed security controls through contracts called Service Level Agreements (SLAs) which only provide a *reactive* approach in case of a breach. While contractual guarantees are useful, a *proactive* approach is also needed to increase the user trust; for example, by providing technological mechanisms to the user (or a user delegated authority like a trusted third party) to verify that the target cloud platform fulfills the security requirements of the user *before* using the service. Moreover, it is also important that the user gets an assurance that his/her VM will always be hosted on a platform which meets his security requirements. In addition to the protection of user VM, the geographical location of user data in the cloud is also important (e.g. due to legal requirements). In this thesis, we consider following research problems which result in the lack of user trust in public IaaS clouds:

- P1. Existing IaaS cloud providers do not provide any techniques to the user which can assure the launch of VM on a cloud platform which fulfills the security requirements of the user such as regulatory compliance. This implies that the user can only rely on contracts and courts to guard against any provider malfeasance. There is a need to provide tools to the user which can guarantee that his/her VM will be launched on a right (secure/trusted) cloud platform.
- P2. The cloud user cannot get any assurance that his/her VM will never be migrated to an insecure platform within or outside of the physically secured datacenter perimeter.
- P3. While there are many data security issues in the cloud, we focus on the *geolocation* of the user data which is currently administered through contractual guarantees (SLAs). User data can be initially stored (like VM launch) or moved (like VM migration) to a storage host which resides out of user approved jurisdiction. There are no technological solutions which can enforce and provide assurance to the user that his/her data will only be available (as plaintext) in a user approved geolocation.

- P4. The existing external audit and certification process – used to increase user trust – does not guarantee that the specific cloud platform which hosts the user VM also meets the certification criteria granted to the CSP after a selective platform audit. Moreover, due to infeasibility of frequent and random audits, the traditional audit and certification process still relies on the CSPs internal security mechanisms which eventually makes the third party audits less significant.

1.3 Research Methodology

Constructive Research method [7] is followed during this work which is based upon the key idea of *construction* that is done by using the existing knowledge in novel ways, and adding the missing links where needed. Principally, the constructive research is a hybrid of traditional *scientific research*, with respect to learning existing systems, and *engineering research* which focuses on creating new artifacts. Following the constructive research hybrid scheme, the scientific research methods were used to study the current state-of-the-art, whereas the engineering research methods were used to produce new results (i.e. models and protocols) based on the existing knowledge base. Finally, the newly created artifacts were implemented to validate the proof-of-concept and its feasibility.

The overall research goal, i.e. *to enable user trust in the public clouds*, is derived and motivated by the importance of research problem faced by the potential security sensitive cloud users [5, 6]. Using traditional scientific research approach, a thorough literature review from various sources - including peer reviewed academic and research publications indexed by publishers like IEEE², Springer³, ACM⁴, etc. - produced the current state-of-the-art which allowed us to define specific research problems needed to be addressed for achieving the overall goal of creating trustworthy clouds. The initial outcome, i.e. a peer reviewed security requirements publication (included in chapter 7), provided an initial set of research problems for further research.

²<http://www.ieee.org/index.html>

³<http://www.springer.com/>

⁴<http://www.acm.org/>

The problems identified in the initial review were further explored with the objective of providing their solutions which are the main research contributions of this thesis (chapter 8-12). Each research problem was further investigated and refined using the scientific research methods which was followed by providing their solutions in the form new models, protocols and prototypes that extend the existing body of research. During the course of research, new problems were identified (e.g. automated platform assessment and rating, geolocation sealed cloud data) which were addressed where important for the fulfillment of overall stated goal.

The proposed solutions are validated by logical argumentation and reasoning approach to discuss their security effectiveness. This is done by providing the security analysis of the proposed solutions. Moreover, the feasibility of the proposed artifacts is also validated through prototype implementations. Finally, the solutions presented in this thesis contribute towards the research goal of enabling user trust in the public clouds. (See Chapter 5-Table 5.1 for a summary of each contribution and achieved research goals)

1.4 Contributions

This thesis provides protocols and models to increase user trust in public cloud services offered according to the IaaS model. The solutions suggested in this thesis are built upon mechanisms proposed by Trusted Computing Group (TCG) which utilize hardware security features offered by a Trusted Platform Module (TPM)⁵. The main contributions of this thesis are the following.

1. Security Requirements Analysis of IaaS Public Clouds

We present a threat analysis for virtual machine provisioning in IaaS clouds to define security requirements for trustworthy public IaaS clouds. It provides an insight on several unsolved security research issues along with providing the foundations for further research in this direction. The threats analysis results are published in ECIW'11 [8] (see chapter 7).

⁵See chapter. 2 for an overview of TCG and TPM

2. Secure Launch of user VM in the cloud

The secure launch of a user-bundled VM image uses trusted computing techniques to allow the cloud user to remotely verify the security properties of the target cloud platform in the provider infrastructure. The proposed solution focuses on building user trust by sealing user VM to a verified cloud platform at launch time. A prototype implementation validates the implementation feasibility of the suggested mechanisms. This work has been published in CLOSER'12 conference which covers various aspects of cloud computing including *security* [9](see chapter 8).

3. Trusted Launch of generic VM instances in Public cloud

The trusted launch of a user VM from a generic VM image requires that the user gets mechanisms to verify that a trusted VM image is used to launch his/her VM and that too on a trusted target host. Using the TPM's sealing mechanisms, the user is assured by a trusted third party that his/her VM is launched on a trusted host. The user can also verify that a trusted VM image is used to launch the user VM instance. The proposed protocol is implemented in an OpenStack IaaS cloud deployment. This work has been published in ICISC'13 [10] (see chapter 9).

4. Secure VM migrations

We present design principles and mechanisms to allow user VMs to migrate across cloud platforms according to user specified security policies about the host platform configurations. Our suggested protocol enforces these policies on platform level preventing migration to untrusted/miss-configured host platforms. The proposed VM migration solution complements and extends our earlier launch protocol and makes the suggested trust model and protocols applicable to a large set of IaaS usage scenarios. The secure VM migration results are published in TrustCom'12 [11](see chapter 10).

5. Continuous Integrity Monitoring and Management of Remote Cloud Platforms

Continuous monitoring and assessment of the cloud platforms is necessary for multiple reasons such as to keep regulatory compliance which consequently increases user trust. We propose automated mechanisms to remotely assess and frequently audit-certify a platform if it fulfills the required security criteria. This is done using novel techniques which integrate trusted computing and security automation solutions such as Security Content Automation Protocol (SCAP)⁶. The TCG-SCAP integration provides the first solution (to best of our knowledge) which addresses the fundamental shortcomings of both TCG and SCAP. The paper published on this work in SIN'13 conference received the **Best Paper Award** [12]. An extended version of this paper including the implementation results is also accepted for the Journal of Information Security and Applications (JISA) [13](included in chapter 11).

6. Geolocation Aware Placement of User Data in the Clouds

The current contractual guarantees that the user data will only be placed in a correct jurisdiction (i.e. user defined geolocation) are useful but not enough to convince the users to send his/her data to the cloud. We propose to complement the existing approaches with a technological solution that gives the assurance that the plain-text user data is only stored and processed in a correct jurisdiction while still providing flexibility to the CSPs to store the encrypted-data anywhere if required for administrative reasons (e.g. cost, performance, redundancy, etc.). This is achieved by using the *sealing* capability of the TPM to seal data to a geolocation which is stored in the TPM by a daemon process which gets the location data from a navigation device (e.g. a GPS receiver which is either natively available to the platform or to a central master node which sends the location data to the platform). A prototype implementation is used to validate the proof-of-concept and performance is measured using Swift distributed object store in a test OpenStack deployment. The results are accepted for publication in TrustCom'14 (included in chapter 12).

⁶A brief overview of SCAP framework is presented in section 2.3

1.5 Thesis Outline

This thesis is in the form of collection of six papers which are included in Chapters 7-12. The outline of the thesis is as follows. A brief over of technologies used in this thesis – that is, cloud computing, trusted computing and security automation standards – is given in Chapter 2. In Chapter 3, we present the current state-of-the-art in the area of focus for this thesis. Chapter 4 presents a complete solution in the form of *Cloud Trust Management Model* which is based upon the solutions proposed in this thesis (i.e. the paper collection included in the second part of the thesis – Chapter 7 to 12). The conclusion and potential future areas of work are presented in Chapter 5. A brief overview of the included papers is given in Chapter 6 which also explicitly identifies the author's contributions in each paper.

Chapter 2

Background

In this chapter, we introduce the concepts which are fundamental for better understanding of the solutions we present in this thesis. We start with an overview of the cloud computing, its stakeholders and common cloud service delivery models (§ 2.1). We also provide an overview of two important technology standards – Trusted Computing (TCG) in § 2.2, and Security Content Automation Protocol (SCAP) in § 2.3 – which are fundamental in our proposed solutions.

2.1 Cloud Computing

A *cloud* is a pool of IT resources – usually virtualized – which are available over the Internet. Virtualization allows to configure computing resources dynamically for the client which can then be leased following the pay-per-use payment model. Other than dynamicity and elasticity provided by virtualization, there are many other characteristics of cloud computing which are defined in various definitions available today. One of the most comprehensive definitions is provided by The National Institute of Standards and Technology (NIST) [1] which states that:

“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.”

Benefit	Description
Flexibility	The underlying virtualization technologies allow cloud users to request for the exact amount of required resources (e.g. a VM with quad-core processing and 8GB memory etc.)
Scalability	The service can be leased/released dynamically on demand
No upfront investment (CAPEX)	The cloud service can be leased dynamically which means that CAPEX costs to start/expand the business can be reduced a lot
Less operating costs (OPEX)	The physical cloud platforms which are used for the provisioned service are managed by the CSP which saves many operational expenses for the cloud user business
Ubiquitous access	The cloud services are usually web bases which allows them to be accessed from variety of devices e.g. laptops, mobile devices, etc.

Table 2.1: Benefits of using cloud based services

The characteristics like elasticity, rapid provisioning, minimal management effort, etc. promise many business benefits [14] (major benefits are listed in Table 2.1) which are instrumental in the recent surge towards the demand for using different cloud services (see § 2.1.1). On the other hand, there are still many unresolved security issues [15, 16] which must be addressed for wide-scale adoption of the cloud services by many potential security sensitive users.

2.1.1 Service Models

The cloud computing aims at providing different IT services (e.g. storage, database, applications, VM, etc.) over the Internet so that the user does not need to worry about establishing and managing the underlying infrastructure. Numerous types of such cloud services have emerged in the recent years. In the following sections, we present a brief summary of the well known categories usually referred to as SPI service models [1] followed by a more detailed use case of the IaaS model that we consider in this thesis.

Software-as-a-Service: Software-as-a-Service cloud model refers to the provisioning of any useful application/software over the Internet. There are many SaaS offerings to date, e.g. Google provides office suite in the form of Google Docs¹. Similarly, SAP provides many business process applications in their cloud solutions². Other SaaS cloud providers are Rackspace³, Salesforce⁴ etc.

Platform-as-a-Service: Platform-as-a-Service cloud model refers to the provisioning of provider managed platforms which provide development frameworks for the cloud users. The operating system and the required development environment is setup and managed by the cloud provider. The cloud user builds up his/her applications using such platforms. Examples of PaaS providers include Microsoft Windows Azure⁵, Google App Engine⁶ and salesfore.

Infrastructure-as-a-Service: Infrastructure-as-a-Service cloud model refers to the provisioning of the computing resources (e.g. CPU, memory, storage, etc.) that are used to create a user Virtual Machine (VM) which belongs to the cloud user and runs just like a physical machine. In an IaaS model, the cloud user takes the responsibility of managing his/her VM by installing the desired operating system and other software. Some examples of IaaS providers are Amazon EC2⁷, GoGrid⁸ and FlexiScale⁹.

Our Use Case: In this thesis, we consider a use case in which the Cloud Service Provider (CSP) sells its computing resources, self-owned or leased from a third party called an infrastructure provider, to the users (or clients) according to an IaaS model (see figure 2.1). There are usually two or sometimes three main entities with respect to their roles as defined in Table 2.2. However, many other business models also exist

¹<http://www.docs.google.com/>

²<http://www.sap.com/pc/tech/cloud/software/cloud-applications/index.html>

³www.rackspace.com/

⁴<http://www.salesforce.com/>

⁵<http://azure.microsoft.com/en-us/>

⁶<https://developers.google.com/appengine/>

⁷<http://aws.amazon.com/ec2/>

⁸<http://www.gogrid.com/>

⁹<http://www.flexiscale.com/>

which can have even more entities (e.g. Amazon Simple Storage Service (Amazon S3¹⁰) offers cloud storage in large quantities called *buckets* which are used by other storage service providers like Dropbox¹¹ to offer services to the end-users).

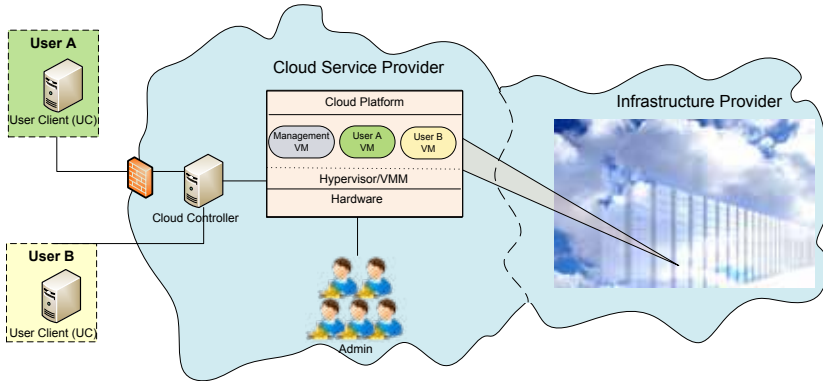


Figure 2.1: Infrastructure-as-a-Service (IaaS) Cloud Model

In the IaaS model, the provider uses virtualization to allocate platform resources (e.g. CPU, memory, etc.) to the user according to his/her needs. Using these resources, the user can create an instance of his/her virtual machine. The use of virtualization allows multiple users to run their VMs on a single physical platform. According to the existing cloud infrastructures, a cloud user performs the following three main steps in creating his/her VM:

1. **Service Request** is done by visiting the provider portal/website
2. **Service Select** involves the selection of the required configuration (number of required CPU cores, required memory and storage etc.) by the user
3. **Service Start** which creates and starts the user VM in the provider network.

There are also other intermediate steps like payment, VM image transfer/selection etc. In this thesis, we analyze this process with respect to

¹⁰<http://aws.amazon.com/s3/>

¹¹<https://www.dropbox.com/help/7/en>

security and focus on the the limitation that the user has not option other than to trust the cloud provider that his/her VM will only be hosted on secure platforms during its life time. We address this limitation and extend the existing process of VM life cycle management by embedding transparency and assurance services into it – by using Cloud Trust Management Model presented in chapter 4 – which alleviates many user concerns.

Entity	Role	Example
Infrastructure Provider	Manage physical cloud platforms	Datacenters
Cloud Service Provider (CSP)	Rent resources from one or many infrastructure providers to render IT services (<i>SaaS, PaaS, IaaS</i>) to the end user	Amazon Inc., Google Inc., Microsoft Corporation
Cloud User or Client	lease or release IT services dynamically on-demand from a CSP for own business operations	An individual or a business like web hosting company etc.

Table 2.2: Stakeholders in a typical cloud model

2.1.2 Deployment Models

Different motivations to use/provide cloud services result in different cloud deployment models. For example, an organization may be interested in setting up a local cloud called *Private Cloud*, for better resource utilization, and hence deliver IT services (XaaS) to various departments (e.g. human resource, finance, etc.). On the other hand, a new/small company might not want to invest heavily on procuring computing resources for its required IT services, rather prefer to lease the required IT services from a third party cloud provider called a *Public Cloud* provider. Both models have their pros and cons which makes room for a *Hybrid Cloud* which tries to maximize the benefits of both models and explore possibilities to minimize the disadvantages of either model (public and private). However, selecting the right balance in utilizing a private or public resource remains an administrative and management challenge in hybrid clouds. We consider the security perspective of the public clouds which promise most business benefits but need better security solutions to fulfill the requirements of many security sensitive users.

Public Clouds - The Security Perspective

The emergence of cloud computing has brought many benefits for both, the user and provider of the service. A large majority of normal IT users (e.g. home users) is already using one or more public cloud services at very less cost¹² (e.g. storage, email, etc.). However, many organizations and businesses – despite recognizing many business benefits promised by the public clouds – hesitate in using public cloud services due to their security concerns. Some companies tend to setup their own private cloud for better internal resource utilization, and also to make their existing infrastructures *cloud-compatible* for any possible future migration to the public clouds. However, large scale migration to public clouds by such organizations can only be made possible if public clouds become more transparent in providing security assurances. The goal of this thesis is to provide the required assurance so that the users can check the implemented security controls in a transparent way.

2.2 The Trusted Computing Group (TCG)

Trust¹³ plays an important role in a stable and long lasting business relationship. In the information technology field, the strengthening of trust is not limited to social aspects such as one-to-one relationship, past experiences of both parties etc (*social trust*). Instead, a remote client also needs technological mechanisms to assess the security of the underlying infrastructure. These mechanisms form the basis of *technical* or *digital trust* [4]. The technology initiative to realize this digital trust was taken in 1999 by a consortium called Trusted Computing Platform Alliance (TCPA). The TCPA consortium, initially having five members¹⁴, was succeeded by Trusted Computing Group (TCG) which has over 190 members to date. The distributed architecture of cloud computing, which is also a reason for lack of user trust in the cloud services, makes good use case and potential area for trusted computing to strengthen user trust.

The Trusted Computing Group (TCG) is a non-profit, industry-standards organization¹⁵ with the mission of proposing specifications and technol-

¹²Sometimes these services are available for free with some usage limitations

¹³"firm belief in the reliability, truth, or ability of someone or something" (*Oxford Dictionary*)

¹⁴Compaq, HP, IBM, Intel, Microsoft

¹⁵http://www.trustedcomputinggroup.org/about_tcg

ogy for *trusted computing platforms*. A trusted computing platform is embedded with a tamper-resistant security chip called the Trusted Platform Module (TPM) which implements the most important TCG specification [17]. The TCG proposed mechanisms can be used for:

- the security of information assets by using securely stored cryptographic keys in the TPM's '*Protected Storage*'.
- securely storing the current platform state in TPM's Platform Configuration Registers (PCR). The platform state can later be reported to a remote client for the platform integrity verification through a process called *Remote Attestation*.

2.2.1 Trusted Platform Module (TPM)

The TCG defined Trusted Platform Module specifications¹⁶ can be implemented in hardware or software; however, a hardware based TPM is considered more secure. TPM is a fundamental security building block in the TCG architecture and therefore its components must function as intended. This can be ensured by following good engineering practices, manufacturing process and industry reviews. The main components of a TPM are shown in Figure 2.2.

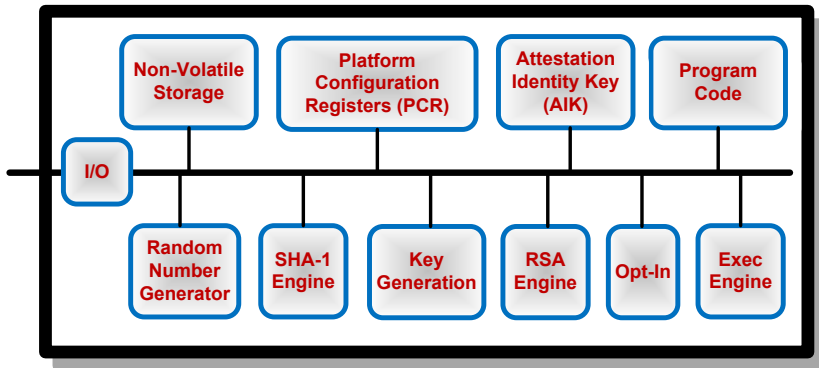


Figure 2.2: TPM Component Architecture

¹⁶The existing chips are built upon TPM 1.2 specification which is used in this thesis; however TCG has recently published TPM 2.0 (a brief overview presented in § 2.2.5)

2.2.2 TPM - Key Management

The TCG proposed security and trust enhancing mechanisms are based upon various protocols and principles supported by the TPM. One of the main capabilities of the TPM is to provide data protection using various types of TPM keys. TPM supports secure ways for its key management which includes the following:

Key Creation: Asymmetric keys of various sizes and attributes can be created using `TPM_CreateKey` command [18] which uses random number generator in the TPM.

Key Storage: Some root keys are stored in the TPM non-volatile storage (NV-storage) but that storage is not sufficient for all TPM keys; therefore other keys are usually stored out of TPM in a *Key Cache* after in-chip encryption called *key wrapping* with one of the TPM-resident root key (e.g. Storage Root Key)

Key Usage: A TPM key can only be used if it is already *loaded* in the TPM. In order to use a wrapped key (which is stored out of the TPM), it is first decrypted with its root key and then loaded for the required cryptographic operation.

A TPM supports many cryptographic operations such as *encryption/decryption*, *signing*, *quote* (reporting platform state), etc. As a basic security principle, different types of keys are used for different purposes for which the TCG classifies TPM keys into following 7 categories. These keys make a key hierarchy as shown in Figure 2.3.

1. **Endorsement Key (EK):** Every TPM has a unique asymmetric key pair called Endorsement Key (EK). The EK is a non-migratable key, that is, its private part never leaves the TPM. The EK is usually generated by the TPM manufacturer and should ideally be supplemented with a certificate called *Endorsement Credential* that is stored in the TPM's NV-storage. The EK is never used for encryption/decryption or signing purposes rather its use is limited to only two tasks. First, to get the platform ownership which is a process of generating a Storage Root Key (SRK) and is only possible with owner's physical presence. Second, the EK is used in the protocol for platform identity registration with the Privacy-CA (PCA) who certifies the Attestation Identity Keys (AIK) discussed below.

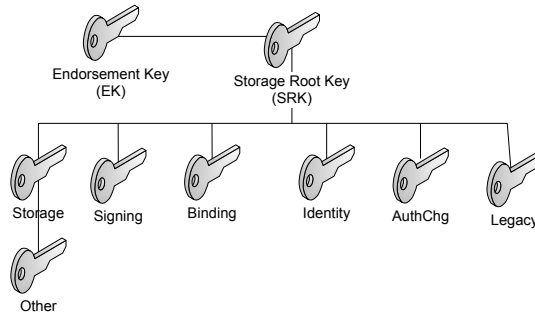


Figure 2.3: TPM Key Hierarchy

2. **Storage Keys:** These are asymmetric keys used for storing other keys or data securely external to TPM. The 'Storage Root Key (SRK)' is an important non-migratable storage key which is the root of all TPM protected keys, that is, every TPM key is wrapped (encrypted) with the SRK before it is stored out of TPM. As stated earlier, SRK is generated when the TPM ownership is taken.
3. **Signing Keys:** These are asymmetric keys which can only be used for signing application data or messages.
4. **Binding Keys:** The bind keys are used to encrypt (or *bind*) small amount of data (e.g. a symmetric key) on one platform and decrypt (or *unbind*) it on a remote platform.
5. **Identity Keys:** These are non-migratable keys and are also called Attestation Identity keys (AIK). These are exclusively used for signing TPM data like PCR register values. An AIK signed message also accompanies an *AIK certificate* which is issued by a mutually trusted CA and certifies that the AIK is a non-migratable TPM key.
6. **Authentication Keys:** These are symmetric keys which are used to secure communication channel with TPM.
7. **Legacy Keys:** These are legacy system keys which are created out of TPM and then imported in the TPM for signing or encryption operations.

2.2.3 TPM Data Protection

In this section, we first introduce the TPM supported operations for data protection and later give a brief introduction of technique we used in our solutions for intended purpose. The TCG classifies TPM protected data in to the following four categories [19]:

1. **Binding/Unbinding:** The TCG notion of binding actually refers to the encryption of data using a public key whose corresponding private key is stored in the recipient's TPM (usually in key cache). The platform with TPM unbinds (decrypts) the received data using TPM_UnBind command after loading the private key in the TPM. It is noteworthy here that the TPM does not have any command to do binding which means that encryption is always performed out of TPM. The binding operation is either performed by using the TCG Service Provider Interface (TSPI) which is a part of TCG Software Stack (TSS) or by any other available cryptographic library (e.g. JavaTM Cryptographic Extension¹⁷, OpenSSL¹⁸, etc.).
2. **Signing:** Signing is also a traditional cryptographic operation to ensure the integrity and authenticity of the data. The TPM supports TPM_Sign command that uses a signing key to return the required digital signature. The recipient of the data can use standard signature verification process to verify the authenticity and integrity of the received data.
3. **Sealing/Unsealing:** Sealing is similar to binding in terms of encryption but it also includes platform configuration which must be satisfied to be able to unseal the data. Since sealing is a TPM operation therefore for performance reasons usually small data like a symmetric key is sealed/unsealed. Hence, in order to seal large amount of data it is first encrypted using a symmetric key which is then sealed (encrypted) using a non-migratable asymmetric key by also providing the platform configuration (PCR values). The platform can only unseal (decrypt) the data if it has the corresponding private key and also the platform configuration matches the values specified in the data sealing. Two important points must be

¹⁷<http://docs.oracle.com/javase/7/docs/technotes/guides/security/jsse/JSSERefGuide.html>

¹⁸<http://www.openssl.org/docs/crypto/rsa.html>

remembered while using the seal/unseal operation. *First*, in the seal operation any platform configuration can be specified, current or future, which is needed for unsealing the data successfully; this implies that the platform is not required to be in the *trusted* state when sealing is done. However, the `TPM_Seal` command also includes the current platform state which is returned by `TPM_UnSeal` command and may, or may not, be of the user's interest. *Second*, the seal and unseal operations must be performed on the same platform. This limitation restricts remote users to seal their data to the destination platform.

4. **Sealed-Signing:** The sealed-signing combines the effects of both operations by including the PCR values in computing the digest to sign. This feature can be used to inspect the sender's platform configuration at the time of data signing.

2.2.4 Sealing Data Remotely

As highlighted earlier, the seal and unseal operations must be performed on the same platform which means that a remote party cannot benefit from the protections provided by the seal operation. However, in distributed systems such as clouds where the remote user owns the resource (a VM) which is managed by another entity, it can be very useful to seal user data *remotely* to the correct platform in trusted state. This requirement also applies to a cloud user who wants his/her VM to run only on known, trusted cloud platforms. We achieve the properties of seal operation in our solutions – described in Chapter 8-12 – by using a TPM key (we call it, a bind key) which is locked to a set of PCRs. We create this key with `TPM_CreateKey` command which allows to specify the required platform state (represented by PCRs) to use that key. The public part of this key is then sent to the cloud user who can use the public bind key to encrypt his information (VM image, token, symmetric key, etc). Since the private bind key is only available to the intended platform in correct configuration, the user information is considered *sealed* to that platform.

2.2.5 TPM 2.0

The TCG, after a span of roughly 10 years, has recently published a new TPM specification which is currently in *draft* state. The TCG describes TPM 2.0 as “a *library specification, which means that it supports a wide variety of functions, algorithms and capabilities upon which future platform-specific specifications will be based. Basically, it is the core capabilities and commands of the TPM. The TPM 2.0 specification will be used as the basis for creation of TPM specifications for different platforms.*”¹⁹ The new specification is divided into four parts (just like TPM 1.2 which has three parts).

- The first part gives an *informative* description of the **TPM Architecture** (similar to TPM 1.2 Design Specification) which includes its properties, functions and methods [20].
- The second part contains a *normative* description of the **Structures** that are used in the TPM commands. The specification provides C-code for all structures (**data types**, **constants**, and **unions**) [21].
- The third part provides a *normative* description of **TPM Commands** along with their C-code [22].
- The final part gives algorithms and methods (as C-code) used by TPM commands [23].

New Features

TPM 2.0 introduces many improvements and new features as compared to TPM 1.2 (see Table 2.3). A list of some key features is given below.

1. **Algorithm Agility:** The current TPM 1.2 supports **SHA-1** and **HMAC** for hash functions while **RSA** is the only available asymmetric key algorithm. The current hash functions are no longer considered secure whereas there are two main problems with the **RSA**. First, it is not feasible for constrained devices due to performance reasons, and second, different geographies and markets prefer their own algorithms. TPM 2.0 addresses these both issues by introducing support for additional set of algorithms such as **SHA-256**, **Elliptic Curve** algorithms (**EC Cryptography**, **EC Diffie-Hellman**).

¹⁹http://www.trustedcomputinggroup.org/resources/tpm_library_specification

2. **Concept of PCR Banks:** The support for new algorithms allows the introduction of *PCR Banks*, that is, it will be possible to have multiple sets of PCRs for each (hash) algorithm. Consequently, the TPM 2.0 commands like `TPM2_Extend` [22, 23] are also extended to specify the correct PCR Bank by an algorithm reference. Hence, a SHA-1 digest will be extended to a SHA-1 PCR Bank, while a SHA-256 digest will be extended to a SHA-256 PCR Bank.
3. **Additional Cryptographic Services by the TPM:** While TPM 1.2 only supports asymmetric key operations (sign, seal/unseal, unbind), TPM 2.0 provides ability to perform more general cryptographic operations such as symmetric encryption (AES), signature verification, etc. As a result, TPM 2.0 can be used as a hardware crypto-module allowing much broader range of usage scenarios.
4. **Support for Multitude of Devices:** When TPM 1.2 specification was released, the main target profile was PC Client which resulted in certain platform specific assumptions (e.g. only support of RSA). However, with the exponential growth of new platforms – such as mobile devices, embedded systems, virtualized systems – a new specification was required. TPM 2.0 is designed to support a broad range of platforms made possible due to main architecture changes stated above (where ECC support is most important for constrained devices).
5. **Improved Management for Better Usability:** TPM 1.2 management (enable, activate, own) and usage is controlled by the owner (who is usually the end user). The management of controlling the *security* and *privacy* in using the TPM becomes so complex for the end-user that he/she never uses it. TPM 2.0 separates these functions by defining three domains namely *Security* (to protect user security), *Privacy* (to protect the identity of the user/platform), and *Platform* (to protect the integrity of the platform services). These domains are realized by introducing three distinct hierarchies (instead of one in TPM 1.2) each having its own resources and controls (authorization, enable/disable).

	TPM 1.2	TPM 2.0
Name	TPM Main Specification	TPM Library Specification
Released	2003	2014
Parts	3	4
Status	Release (in production)	Draft (Work in Progress)
Algorithms	RSA SHA-1, HMAC One-time-pad with XOR	All in TPM 1.2 ECC, ECDH SHA-256 AES More if needed (SM-series)
PCR Banks	single	multiple
Management	Single Hierarchy (storage)	Three Hierarchies (Platform, Storage, Endorsement)

Table 2.3: Comparison of TPM 1.2 and TPM 2.0

Road Map and Availability

Along with the release of TPM 2.0 Library Specification, TCG has also published a *draft* PC Client Platform specific TPM Profile (PTP) Specification [24] which is also available for public review. The PTP specification is the first in a series of (most probable) upcoming specifications for a wide array of platforms such as mobile devices, embedded systems, servers, etc. Additionally, release of supporting specification documents like TCG Software Stack (TSS) is also in pipeline (but no time-line is available).

The availability of products based upon TPM 2.0 depends upon the vendors whether they release products according to *draft* specifications or wait until the release of *final* specifications by the TCG (TPM 2.0 specification, platform-specific specification, and compliance specifications). However, it is expected²⁰ that the vendors will release products as soon as they can which can later be upgraded (firmware upgrades are possible) when final specifications are released by the TCG. Due to the availability of C-code in TPM 2.0 specifications, one can expect an early release

²⁰http://www.trustedcomputinggroup.org/resources/tpm_20_library_specification_faq

of TPM 2.0 compliant products and APIs due to significantly less effort needed to implement TPM Commands. Moreover, even though the provided code is not meant to meet any level of conformance, it is still expected that different vendors will use it (making it a de facto standard) which will result in a consistent behavior by all TPM chips.

2.3 Security Content Automation Protocol (SCAP)

The Security Content Automation Protocol (SCAP) is a NIST standard [25] which defines the formats that can be used to define and communicate the security flaws in a software. The SCAP framework has eleven components (each defined as a NIST standard) in five categories – briefly described below – that can be used to enable automated vulnerability management, measurement, and policy compliance evaluation (e.g., FISMA compliance) [26]. The coordinated use of SCAP components also allows the security assessment and rating of a platform by an entity (e.g. an accredited third-party lab)

2.3.1 Languages

The SCAP languages provide standard way to express security policy, technical check mechanisms, and assessment results using Extensible Markup Language (XML) so that the resulting documents can be effectively used by the both, humans and machines. The following three SCAP languages have been standardized by the NIST:

- ***Extensible Configuration Checklist Description Format (XCCDF)***

The Extensible Configuration Checklist Description Format (XCCDF) is a specification language [27] expressed in XML that is used to write the security checklists, benchmarks, and related kinds of documents. An XCCDF document can be created by a government department to define the minimum security controls, by a industry consortium to express the baseline configuration for a particular platform (e.g cloud platform), or by a security expert team to define a checklist that is used by the security tools installed on the platforms to assess a specific vulnerability threat on that platform.

- ***Open Vulnerability and Assessment Language (OVAL®)***
Open Vulnerability Assessment Language (OVAL) [28] is an XML-based description format which is used to define security checks through definitions called *OVAL Classes*. There are four OVAL classes namely - 1) *compliance* which is used to define security settings, 2) a *vulnerability* class used to define steps needed for finding known vulnerabilities, 3) a *patch* class is used to remediate a vulnerability through relevant patch, and 4) an *inventory* class which is used to collect computer information (e.g. the operating system it runs, presence of an application or software, etc.)

2.3.2 Enumerations

The SCAP enumeration standards provide unique nomenclature to define universal names for various elements namely: 1) code-based software vulnerabilities using Common Vulnerabilities and Exposures (CVE) format [29], 2) configuration-based software vulnerabilities represented by Common Configuration Enumeration (CCE) [30], and 3) to define unique names for every software using Common Platform Enumeration (CPE) [31]. These vulnerabilities (represented by CVEs and CCEs) and the affected items (represented by CPES) are delivered to the end systems for necessary action through public repositories such as National Vulnerability Database (NVD) [32].

2.3.3 Measurement and Scoring Systems

The SCAP framework includes a scoring system which provides metrics to estimate the security impact due to a software vulnerability or a bad configurations. The software vulnerabilities are rated using Common Vulnerability Scoring System (CVSS) [33] whereas the configuration based vulnerabilities are represented and rated using Common Configuration Scoring System (CCSS) [34]. The CVSS and CCSS scores represent the severity of the vulnerability on a scale of 0.0 to 10.0 (where 10.0 represents the most severe vulnerability)

Chapter 3

Trustworthy Clouds - State-of-the-Art

The use of trusted computing and using security audits for making clouds trustworthy is an active topic of research. In this chapter, we first present the current *state-of-the-art* (§ 3.1) and later present the current *state of practice* (§ 3.2) to highlight the relevance and significance of our work according to the actual industry demands.

3.1 State-of-the-Art

Cloud security is an important and (therefore) active area of research [35]. The cloud security issues exist in many areas as described in a comprehensive guide [15] published by the Cloud Security Alliance (CSA). There are many other survey papers which identify different types of security issues at different layers [3, 5, 36, 37, 38]. These all security issues – combined with the fact that user has to delegate his control of data to a service provider – result in a lack of user trust in the cloud services despite many potential benefits. Increasing user confidence in using public cloud services is an important research challenge which demands solutions that make the basis for *trustworthy clouds* [39]. Historically, independent third party audit and certification mechanisms orchestrate the required trust in the traditional IT systems. Same approaches have been adopted in the cloud computing domain as best practices which

are good in principle but – due to the dynamics of cloud computing – traditional audits & certifications fail to achieve the same level of transparency [40, 41]. In response, the research community has proposed many partial solutions using latest technologies especially trusted computing [9, 10, 11, 42]. Moreover, efforts are being made to address the limitations of traditional audit schemes and propose much more flexible and dynamic audit approaches for the clouds [13, 41, 43, 44]. In the following sections, we highlight some important contributions in the existing body of research which define path towards trustworthy clouds.

3.1.1 Trustworthy VM Launch and Migration

The first operation – in moving from traditional self-hosted IT models to a service-based cloud model (i.e. IaaS) – is launching a user VM in the cloud. For a trustworthy cloud, many transparency enhancing services are needed including mechanisms to allow users to verify that his/her VM will be hosted on a *correct platform*; that is, a platform which meets user security requirements. Moreover, it is also important to ensure that the same security policy is used and enforced during the entire VM life cycle (launch, migration, etc.). This requirement of knowing the target cloud platform state before starting to use the cloud service is investigated by many researchers where the common denominator is the use of trusted computing mechanisms of *remote attestation*. There are different threat models which assume that the cloud platform could be compromised (deliberately, mistakenly, or due to changed IT environment) and cannot be trusted. For example, Santos et al. [42] assume threat from malicious insiders (platform administrators) who can trick user VM to run on a compromised platform. The authors propose to set up Trusted Cloud Computing Platforms (TCCP) which register with a Trusted Third Party (TTP) called External Trust Entity (ETE). The ETE performs remote attestation for the cloud users which defines trustworthiness of the platform during VM launch and migration. While the Santos et al. solution proposes good principles for trusting the host platforms, the solution uses rigid remote attestation technique; as a consequence, it overlooks the fact that different users have different security requirements [45]. Moreover, there are few other shortcomings which are considered in our solutions for secure VM launch [10] and migration [11] (see Chapter 9 and 10).

3.1.2 Cloud Transparency through Audits

The Audit & Certification process is an important part of the service industry to determine, demonstrate and communicate the correct state of business affairs (processes, safety and security controls, etc.) to their clients in order to increase clients' confidence in the service. Similar approaches can be used to improve cloud transparency to realize trustworthy clouds. While all major cloud providers employ audit and certification at present, the current systems are only replicated from classic IT infrastructures and therefore not feasible in the cloud environments. Many position and survey papers highlight the importance of cloud audits to improve cloud transparency - an important requirement for trustworthy clouds [40, 41, 43, 46].

Doelitzscher et al. in a recent research [41] provide a detailed insight into cloud audits wherein the authors study traditional audit approaches and identify their shortcomings in the cloud context. The authors identify cloud security issues (and how audits can help) and group them into two categories - amplified cloud security problems, and cloud specific problems. The first category includes the security issues which are already known and exist in traditional distributed systems environments but their importance is amplified in the cloud environments (e.g. malicious insiders, lack of transparency for applied security measures, etc.); while the later category only includes security issues which arise due to cloud-specific environment (unknown data location, nefarious use of cloud resources, insecure APIs, etc.). The paper proposes Security-Audit-as-a-Service (SAaaS) architecture which uses agents-based monitoring of the cloud environment. Multiple agents are used at different layers which continuously monitor the key points where they are placed. The monitoring is done according to a set of rules defined in the Security Service Level Agreements (SSLA). The proposed solution is in its initial stage (first prototype and basic architecture defined), therefore it lacks many detailed components; for example, the authors envisage tools for formal modeling of SSLAs for cloud environments for which complete modeling language and a graphical modeler is needed. While the paper provides a good degree of details in understanding cloud audits, the solution itself is not analyzed from a security perspective. For example, the protection or security of agents is not discussed which is very important for an audit framework because the audit becomes useless if it is not performed by a trusted entity - a physically present certified auditor or

a trustworthy software (e.g. TPM protected). Moreover, the placement of agents in some points – for example, in user VMs as suggested by the authors – can be problematic for users unless they get assurances that the correct and authorized agent is placed.

Other studies also consider the use of cloud based security assessment of cloud environments such as Risk Assessment-as-a-Service for cloud computing [40] wherein the authors describe the risk assessment and group it into three types. 1) *Security Assessment* which determines and provides the enterprise’s security knowledge, 2) *Privacy Assessment* which covers the unauthorized disclosures, and 3) the *Compliance Assessment* which deals with check the domain specific assessments (e.g. PCI DSS). The main focus of the paper is to show that the risk assessment (what and how to assess) needs to be redefined for clouds because conventional IT assessment models are based upon different primitives and assumptions such as single vs multi-tenancy, local vs global access, etc. which need a more careful consideration. The authors conclude the importance of real-time cloud assessments using automated audit solutions.

The use of automated audits makes it feasible (and possible) to schedule audits in a much more temporal and on-demand fashion instead of relying on less frequent (annual or biannual) audits done today. This opportunity is investigated in many recent studies [47, 48] and solutions for continuous audit schemes are presented [12, 13, 44, 49, 50]. In [49], Kazi et al. discuss the importance of verifying and expressing compliance and its role in bringing transparency for the clouds. The authors focus on the requirement of a tool for real time compliance verification and propose an Automated Security Compliance Tool (ASCT) for the Cloud. The ASCT uses three components - 1) *Data Collection Engine* which collects the audit critical data (e.g. vulnerability reports, logs, etc.), 2) a *Verification Engine* which processes the audit data and checks the compliance status, and a 3) *User Interface* which allows users to request for compliance test. The authors consider four mechanisms for collecting the audit data which include API, vulnerability scanning, log analysis and manual entry. The API approach uses custom built Audit APIs which collect specific system information (e.g. system time). The vulnerability scanning approach uses an existing vulnerability scanner (e.g. OpenVAS is used in the prototype) and uses its reports. The log analysis approach gathers system information by analyzing the system logs; the ASCT prototype does not use this approach but refers to other solutions

that suggest the use of log analysis. Finally, authors consider manual entry of audit information by the system administrators. The ASCT prototype – deployed in an OpenStack cloud – uses CloudAudit API for the user interface to execute a required audit. The ASCT solution has some shortcomings as well; first, a vulnerability scan is a time consuming task which can result in a denial of service (DOS, DDOS) attack if it is requested by multiple users simultaneously (or even if scheduled one after the other). Main reason for this shortcoming is the absence of a certification mechanism which can be reused after a successful audit – a technique we propose in the \mathcal{ASArP} solution [13]. Second, the authors identify the challenges of security and integrity of the audit critical data, assurance requirements about the correctness of ASCT itself etc, but do not provide a mechanism or discussion on the issue which is of fundamental importance for an audit framework. Third, while the provision of adding multiple audit data collection mechanisms is useful, the use of manual entry can be avoided to keep the solution fully automated. However, this does not assume that manual data is not important and can be part of traditional manual audits (which cannot be fully replaced even with the presence of a good automated solution).

3.2 State of Practice

This section describes the current state of practice which includes - 1) an overview of a security-focused cloud framework (§ 3.2.1) which aims at legalizing and realizing the use of cloud services by government organizations; and an overview of a trusted cloud framework (§ 3.2.3) by an open industry consortium.

3.2.1 FedRAMP

The Federal Risk and Authorization Management Program (**FedRAMP**) [51] is a security focused cloud framework which defines processes to guarantee that the CSPs certified by one of the accredited auditors, called Third Party Assessment Organizations (3PAO), implement the security controls that are required by the US federal agencies intending to use that cloud.

The Federal Risk and Authorization Management Program (**FedRAMP**), enacted in December 2011, is a collaboration of cyber-security and cloud

experts from various public, private and standardization organizations such as National Institute of Standards and Technology (NIST)¹, U.S. Department of Defense (DOD)², Nation Security Agency (NSA)³, etc. The FedRAMP provides a standardized approach to security assessment, authorization, and monitoring for cloud products and services [51] with a focus to ensure secure cloud computing that fulfills the security requirements of the US federal government as mandated in the Federal Information Security Management Act(FISMA)⁴. The FedRAMP program defines a process for the transparency, compliance and auditability of the cloud service to strengthen user trust in it. This process is briefly enumerated below.

Process

1. Step 1: The CSP implements the security mechanisms that fulfill the required security properties. This is done by following the recommended standard security controls as defined by the NIST Special Publication 800-53, Revision 3 [52].
2. Step 2: A FedRAMP accredited auditor called Third-Party Assessment Organization (3PAO) performs the security assessment of the CSP by following the guidelines for assessing security controls as defined by NIST [53] which is consistent with the NIST Special Publication 800-53, Revision 3. The outcome of this assessment is a report called *Security Assessment Package*.
3. Step 3: The security assessment package is reviewed by the Joint Authorization Board (JAB) that consists of the representatives from government/federal agencies. If approved, the CSP gets provisional authorization called *Authority to Operate (ATO)*⁵ which allows the federal agencies to use CSP's services. The CSP is required to maintain the compliance afterwards by performing a variety of continuous monitoring tasks.

It is worth mentioning here that the continuous monitoring for compliance is done by the CSP whereas the 3PAOs are not responsible for per-

¹<http://www.nist.gov/index.html>

²<http://www.defense.gov/>

³<http://www.nsa.gov/>

⁴<http://csrc.nist.gov/groups/SMA/fisma/index.html>

⁵List of FedRAMP compliant CSPs:<http://cloud.cio.gov/fedramp/cloud-systems>

forming continuous audits (regular audits are scheduled over months). As a result, the reason for third party audit – that is, to shift the source of trust from CSP to a trusted third party – is not fulfilled effectively.

3.2.2 European Union Cloud Strategy

While the FedRAMP framework is targeted for US state agencies/public offices, other regions have their own laws and regulations. For example, the European Union (EU) considering its own requirements [54] has devised a *European strategy for Cloud computing* [55] with a focus of promoting rapid adoption of cloud computing in all sectors of the economy. The EU communiqué identifies key areas needed to support wide scale cloud adoption including a goal to increase user confidence in public cloud services. The major tasks outlined in the EU cloud strategy include – a) the identification of proper standards for security, interoperability, data portability, b) definition of EU-level technical specifications which focus on EU regulations (privacy, data protection), and c) identification of a list of cloud certification schemes [45] and publish a list of such scheme⁶. The EU cloud initiative is an ongoing activity which involves many countries, standardization bodies, and companies⁷. The target is not to introduce a new set of standards, rather to identify and recommend and use existing solutions which fit well with the EU requirements.

3.2.3 Security, Trust and Assurance Registry (STAR)

The Security, Trust and Assurance Registry (STAR)⁸ is a Cloud Security Alliance (CSA) initiative which started in 2012. STAR is a public registry for users which lists the security, trust and assurance level of different CSPs⁹. There are three main STAR levels where each incremental level promises better transparency and assurance services. These levels are listed below:

⁶Cloud Computing Certification Schemes List (CCSL) deliverable is now available at <https://resilience.enisa.europa.eu/cloud-computing-certification> [Accessed 19-05-2014]

⁷Working Groups and their activities can be followed at <http://ec.europa.eu/digital-agenda/en/european-cloud-computing-strategy>

⁸<https://cloudsecurityalliance.org/star/>

⁹Number of CSPs in STAR registry = 64 (as on May 05, 2014)

1. **STAR Self-Assessment** is free and open for all CSPs who themselves evaluate and submit their services according to CSA published best practices (including major standards like FedRAMP, COBIT, PCI-DSS, ISO27001, etc)¹⁰
2. **STAR Certification** level is achieved by the CSP after a rigorous assessment of its security controls (according to ISO/IEC 27001:2005) by an independent and accredited CSA certification body (e.g. British Standard Institution (BSI))¹¹. This level is similar to FedRAMP certification except that the assessment is done based on different security standards (FISMA vs ISO/IEC).
3. **STAR Continuous** is based upon continuous assessment and auditing of relevant security properties. STAR Continuous is currently under development (target delivery date is 2015) but it will be based upon Cloud Trust Protocol (CTP) and CloudAudit¹² – two important components described in the following sections.

Cloud Trust Protocol

As stated earlier, transparency is an important requirement for trustworthy clouds. There are different factors that can contribute to improve the cloud transparency, such as knowing the configuration of the cloud platform, geographical location of the data, implemented mechanisms for data security, etc. These are usually referred to as *Elements of Transparency* (EoT) [56]. A user can request any or all EoT which can be communicated to the user using Cloud Trust Protocol (CTP).

Cloud Audit

CloudAudit – supported by over 250 stakeholders – provides an interface (programming API) which can be used for automated assessment and audit of the cloud platforms remotely. It is intended to be used by an authorized entity like cloud user, its duly authorized representative or an approved auditor.

¹⁰Number of CSPs at Self-Assessment Level = 57 (as on May 05, 2014)

¹¹Number of CSPs at STAR Certified Level = 7 (as on May 05, 2014)

¹²<http://cloudaudit.org>

3.2.4 Relevance with our work

The solutions we provide in this thesis and the complete Cloud Trust Management Model presented in chapter 4 is inline with the major industry initiatives discussed in this chapter. While the current FedRAMP framework lacks continuous audit, our solution can be used to improve the existing ways of doing audits, certifications and continuous risk monitoring. On the other hand, the STAR Continuous – which is under development – considers the requirement of continuous audits and proposes an automated audit interface which can be used to realize our solution. Moreover, we have considered different EoTs as proposed by the CTP which include : a) **Trust-Token** for platform configuration, b) geolocation (used for data placement in chapter 12), etc. Hence, our proposed Cloud Trust Management Model implements the CSA STAR Continuous conceptual framework.

Chapter 4

Putting the Pieces Together

In this chapter, we present a complete solution that we envisage in the form of a Cloud Trust Management Model (in Section 4.2). The presented solution uses the results included in this thesis (Chapter 7 to Chapter 12), however minor adaptations are done to integrate all pieces together.

4.1 Scenario

We keep our focus on an IaaS cloud model (i.e. the cloud deployment model considered in this thesis) in which we consider three main entities - the Cloud Service Provider (**CSP**), the cloud user (**user**), and a trusted third party called Trusted Trust Orchestrator (**TTO**) who brokers the cloud trustworthiness. The role of **TTO** can be considered similar, but not limited, to the third Party Assessment Organizations (3PAOs) in the FedRAMP framework [51], or the British Standard Institution (BSI) as a CSA accredited certification body (as discussed earlier in Chapter 3). Many other solutions for trustworthy clouds also rely on a similar TTP with different roles (e.g. auditing, CSP trust assurance, target platform integrity assurance, Transparency-as-a-Service, Audit-as-a-Service, etc.).

The **user** can be a government department or a small company like a merchant who handles the payment card information (e.g. debit, credit,

prepaid, etc.) and wants to host its IT system in a public cloud. However, in shifting the IT resources to the cloud, the **user** is still responsible to demonstrate the compliance of its host environment (e.g. cloud platforms) which is not in **user**'s control anymore. Therefore, the **user**'s concern is to get an assurance from the **CSP** that the required compliance requirements are fulfilled. The solutions proposed in this thesis are aimed at providing such assurances which can be combined together in a Cloud Trust Management Model presented in the following section.

4.2 Cloud Trust Management Model

The cloud trust management model describes the steps and mechanisms to improve the cloud transparency by providing security assurances to the remote user. While there are many cloud specific security properties, this thesis focuses on providing the following three assurances - 1) that the host platform runs a correct software stack (i.e. only runs good, known and updated software); 2) that the platform is configured correctly (i.e. according to a best known setting); and 3) the plaintext user data is available only at user defined geographical locations. As a result, these assurance mechanisms help in increasing user trust in public IaaS clouds. The following sections describe the processes in the envisioned cloud trust management model (see Figure 4.1).

4.2.1 Define Cloud Platform Baseline

We define *Cloud Platform Baseline* as a reference platform profile which specifies a thoroughly analyzed and tested software stack (from security perspective) as well as a secure configuration of important software components (antivirus, firewall, SELinux, etc.). Defining a single cloud platform baseline to satisfy the security requirements of every cloud user is impractical; however, it is still possible to thoroughly test and define a reference profile for a particular use case according to its specific security requirements. One working example of a similar reference profile (which only covers software configurations) is United States Government Configuration Baseline (USGCB)¹ which defines the required configuration for Information Technology products deployed across the federal agencies. In a same way, we consider that different cloud platform baselines are

¹http://usgcb.nist.gov/usgcb_content.html

defined for different cloud use cases by their relevant governing bodies (see Table 4.1 for different entities and their roles in each process of the Cloud Trust Management Model). The required software configurations can be defined according to the current industry practice of creating automated checklists using Extensible Configuration Checklist Description Format (XCCDF) [27]; whereas, the components of an approved software stack can be defined as an XML formatted white list. The resulting reference profiles for every use case (e.g. PCI DSS, HIPAA, FISMA, etc.) can be made public through online repositories such as the National Checklist Program (NCP)². These reference cloud platform profiles are implemented by the CSP (see Section 4.2.2) and later verified by the **user** himself or by the TTO on **user**'s behalf (see Section 4.2.3).

Entity	Examples	Role
Governing Body	FedRAMP, ENSIA, PCI	Define Cloud Platform Baselines
Cloud Service Provider	AWS, Windows Azure, Rackspace	Setup Cloud Platforms according to relevant baselines
Trusted Third Party	Accredited Auditors e.g. BSI	Audit & Certify Cloud Platforms according to relevant baselines
Cloud User	Public or Private Organization	Verify Cloud Platform compliance

Table 4.1: Cloud Trust Management Model - Entities and their Roles

4.2.2 Setup Cloud Platforms

The role of a CSP is to set up the cloud platforms and implement security controls for their security. The security controls can be same for all platforms, however in a more practical scenario a CSP may define different pools of platforms that are setup with different types of security requirements defined by the relevant regulatory authority (e.g. HIPAA, PCI DSS, FISMA). For example, a *PCI-compliant* pool will have all the platforms that fulfill the security requirements as defined by the PCI Data Security Standard. In our model, we assume that a CSP will setup

²<http://web.nvd.nist.gov/view/ncp/repository>

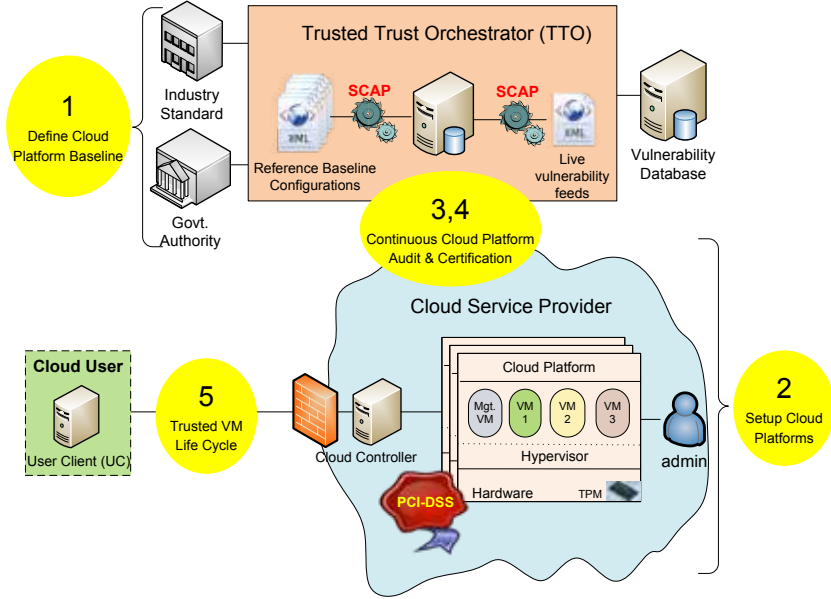
pools of platforms according to the defined cloud platform baselines (described in § 4.2.1). Furthermore, we assume that the cloud platforms have physical TPM and implement trusted computing primitives – such as trusted boot, remote attestation, etc. – which are needed for platform certification described in the next section (§4.2.3). As a standard practice, the CSP’s implement internal security control mechanisms such as using antivirus programs, firewall, intrusion detection and prevention systems, and continuous monitoring and remediation mechanisms (using SCAP) for newly reported threats and vulnerabilities. However, it is still important – for both the **user** and the **CSP** – that the **user** gets an assurance that such mechanisms are in place. This assurance is usually given by a third party audit and certification process.

4.2.3 Cloud Platform Certification

There are many shortcomings of existing audit and certification mechanisms as identified in the current state of the art (see § 3.1.2). Our solutions address two important issues namely *incomplete* and *infrequent* audits. While the problem of infrequent audits is solved through a continuous audit mechanism discussed in the next section (§ 4.2.4), the problem of incomplete audits is usually overlooked. We introduce the problem of incomplete audit which arises due the fact that the traditional audits are only performed on a selected subset of (representative) nodes out of thousands of physical nodes. In case of a successful audit (which is performed only on a few nodes), the certification is issued to the **CSP**. However, there is still a high probability of the presence of non-compliant (or vulnerable/malicious) platforms where user VM can end-up. Therefore, we propose that the compliance certification should be assigned at the cloud platform level instead of **CSP** level. We achieve this requirement by using an automated cloud platform certification approach proposed in our \mathcal{ASArP} solution (presented in Chapter 11). The platform certification process can either be started by the cloud platform (e.g. after it is initially setup by the **CSP**), or by the **TTO** for continuous monitoring (see § 4.2.4) as required by the CSA STAR Continuous mandate (described in §3.2.3).

TTO uses the \mathcal{ASArP} solution to audit the platform compliance against the required baseline; for example, to check the compliance of the \mathcal{TP} for PCI-DSS standard, \mathcal{ASArP} is used to check its profile against the cloud platform baseline defined by the payment card industry. On a successful

Figure 4.1: Cloud Trust Management Model



evaluation (audit), the audited cloud platform gets a trust credential **Trust-Token** which is a short lived platform certificate (e.g. with 7 day validity if weekly audits are needed and scheduled). The **Trust-Token** is a modified version of the trust credential proposed in Chapter 10 where we replace the TAL value with an identifier (*RefProfile*) which identifies the compliance check requested, performed, and passed (e.g. PCI DSS, FISMA, HIPAA). After a successful audit and certification of the platform, a remote cloud user can verify its compliance – without requesting time consuming audit again – by only requesting the **Trust-Token** and using the PK_{BIND} key in it (see § 4.2.5 for a usage scenario of the **Trust-Token**, and Chapter 10 for its properties). The revised **Trust-Token** is expressed as follows:

$$TT = \text{Sig}_{TTO}[TTID, PlatformID, RefProfile, PK_{BIND}, Validity]$$

4.2.4 Continuous Platform Audit & Certification

The current state-of-the-art lacks solutions for continuous Platform Audit & Certification mechanisms which is a major problem in times when new vulnerabilities and exposures are identified and reported everyday which are sometimes security critical. By allowing automated, frequent, and platform level certifications as proposed in the \mathcal{ASArP} solution (in chapter 11), the **user** gets the assurance that the **CSP** always maintains, and **TTO** frequently verifies the platform security profile as stated in the initial SLAs.

4.2.5 Trusted VM Life Cycle

The processes presented in the previous sections set the foundations for a trustworthy cloud. The proposed solutions allow **user** to verify the statements/claims, in the presented SLAs, regarding the integrity of the cloud platforms. As a result, the **user** can specify and verify that her VM is hosted only on a cloud platform which fulfills her security requirements or legal obligations (e.g. compliance to PCI DSS, HIPAA, etc.). In the user VM launch process, **CSP** schedules the user VM on a cloud platform and sends the platform certificate to the user which provides assurance that the platform is evaluated, audited and certified by the **TTO**. The **user** uses the PK_{BIND} key in the **Trust-Token** to encrypt³ her VM image if it is custom image (use case discussed in chapter 8), or her token if generic VM image is used for the launch (use case discussed in chapter 9). Similarly, **Trust-Token** is used whenever user VM is migrated to another platform using the migration protocol described in chapter 10. As a result, the user gets assurance that her VM will be hosted only on correct platforms during its life time.

³anything encrypted with PK_{BIND} is sealed to that platform in a defined state

Chapter 5

Conclusions and Future Work

5.1 Conclusion

This thesis focuses on the reasons for existing trust deficit amongst the cloud stakeholders which is addressed by providing transparency mechanisms to the users (or their trusted assessors). We consider a public IaaS cloud model and propose mechanisms for ensuring that user VM is launched and remains hosted on trustworthy cloud platforms. Additionally, we propose solution to provide assurance that user data is only available (in plaintext) in a geolocation approved by the user. In our solutions, we use *remote attestation* and *sealing* capabilities provided by trusted computing. Our solutions significantly rely on trusted computing solutions which – along with its strengths – has few but important shortcomings; such as, lack of support for the management of software configurations, lack of platform assessment solutions based upon the Integrity Report, etc. In order to make our solutions more practical, we also address these issues in trusted computing domain and propose a novel solution which integrates security automation techniques provided by SCAP with trusted computing (a TCG-SCAP synergy).

The solutions we provide in this thesis are also validated (for implementation and performance feasibility) by using prototype implementations. Our solutions promise better transparency and assurance services

Proposed Solution	Problem Addressed	Goals Achieved
Security Considerations for Virtual Platform Provisioning (Chapter 7)	Find research problems in the Area	G1 – Identified (some) security issues that inhibit wide-scale cloud adoption
Secure Launch Mechanism for User-packaged VM (Chapter 8)	P1 – The integrity of the host cloud platform is not known	G2 – New Protocol that adds Transparency & Assurance Services G3 – Prototype validates implementation feasibility
Secure Launch Mechanism for Generic VM (Chapter 9)	P1 – The integrity of the used VM-image and the host cloud platform is not known	G2 – New protocol to add Transparency & Assurance services G3 – Prototype validates implementation feasibility
Secure VM Migration (Chapter 10)	P2 – User VM can be migrated to an untrusted platform	G2 – New protocol that adds assurance service
Enforcement and Assurance mechanisms for Data Placement in Correct Geolocation (Chapter 12)	P3 – User cannot enforce the geographical location his data	G1 – New security issue identified G2 – New protocol to add Assurance service G3 – Prototype validates implementation and performance feasibility
Solution for Continuous Platform Audits (Chapter 11)	P4 – The current audits are selective and not frequent Additional – Existing remote attestation solutions do not allow dynamic security evaluation of the remote platforms	G1 – New security issue related to shortcomings of manual audits G2 – New protocols which add Transparency and Assurance services G3 – Prototype validates implementation and performance feasibility

Table 5.1: Summary of Our Proposed Solutions, Problems they Solve, and Research Goals Achieved

for public clouds and fulfill the goals set for this thesis. In Table 5.1, we revisit these goals and provide a summary of our proposed solutions for different research problems. Finally, it is noteworthy that our proposed solutions are inline with the initiatives – taken by the government (US Government, EU cloud strategy) and industry consortium (CSA) such as FedRAMP and CSA STAR Continuous – for trustworthy clouds such as FedRAMP and CSA STAR Continuous.

5.2 Future Work

We see many possible directions for future work in both areas - Trustworthy Clouds and Trusted Computing – which are discussed in the following sections.

5.2.1 Sealing Data to a Platform State

The solutions presented in this thesis mainly use trusted computing as the fundamental technology to guarantee the correctness of the remote platforms. While we propose a real-time and dynamic remote platform assessment framework as opposed to the traditional rigid snapshot-based evaluation, there are still many other research challenges in the trusted computing domain. For example, the sealing of data to a specific *platform* can easily be done by only using the platform boot aggregate (PCR 0-7). However, sealing to the *current platform state* is not straight forward because in that case sealing PCR (e.g. PCR 10 representing the platform state) will always have different value in every system reboot due to the different order of loading software modules. Consequently, if we seal data to a specific state, the data will not be available after the reboot even if the platform configuration is not changed. There are different techniques to solve this problem such as by minimizing the contents of the TCB and using the measurements up to a certain level (which yields much more static result across reboots); or by changing the measurement policy for example from `measure_all` to `measure_executables_only` to keep the measurements manageable. However, these techniques have associated compromises (e.g. does not provide runtime protection, a vulnerable or malicious library module remains undetected) which we

plan to undertake in the future work. The main focus for our future solution will be to seal/unseal data to a platform state which includes every loaded item (executable, library, scripts, etc.).

5.2.2 Implementation of a Standards-based Trustworthy Clouds

As we have presented in § 3.2.4, our proposals are inline with main industry initiatives in this direction (FedRAMP, CSA STAR Continuous) and provide first implementations towards the envisaged goals; we plan to extend our solution with a more realistic prototype. This will be done by integrating and testing our solutions (e.g. *ASArP*) according to CSA STAR Continuous framework by deploying it in an OpenStack cloud implementation. Our solution will allow a user to request the Elements of Transparency (platform software stack, configuration compliance, geolocation) from the CSP using Cloud Audit interface which will eventually increase user trust in the cloud service.

Chapter 6

Overview of Papers

This thesis is a collection of six papers included in the next part (Chapter 7 to 12). The first paper (in Chapter 7) provides a basic set of security problems for provisioning resources to host user VMs (a cloud use case). The security issues identified in the first paper – VM launch and VM migration – are considered in papers included in Chapter 8 to 10. The paper included in Chapter 11 proposes a detailed solution to be used for the security evaluation and assessment of remote cloud platforms; after successful evaluation a platform certificate is issued which can be used in our proposed secure VM launch and migration protocols. Finally, we consider the security of data geolocation in Chapter 12. The following sections provide a brief overview of the included papers where the main contribution of the paper and also the author’s (of this thesis) contribution is identified.

6.1 Security Considerations for Virtual Platform Provisioning

Mudassar Aslam, Christian Gehrman. Security Considerations for Virtual Platform Provisioning. *In 10th European Conference on Information Warfare and Security ECIW-2011, 7-8 July 2011, Tallin, Estonia.*

Summary

The concept of virtualization is not new but leveraging virtualization in different modes and at different layers has revolutionized its usage scenarios. Virtualization can be applied at application layer to create sandbox environment, operating system layer to virtualize shared system resources (e.g. memory, CPU), at platform level or in any other useful possible hybrid scheme. When virtualization is applied at platform level, the resulting virtualized platform can run multiple virtual machines as if they were physically separated real machines. Provisioning virtualized platforms in this way is often also referred to as Infrastructure-as-a-Service or Platform-as-a-Service when full hosting and application support is also offered. Different business models, like data-centers or telecommunication providers and operators, can get business benefits by using platform virtualization due to the possibility of increased resource utilization and reduced upfront infrastructure setup expenditures. This opportunity comes together with new security issues. An organization that runs services in form of virtual machine images on an offered platform needs security guarantees. In short, it wants evidence that the platforms it utilizes are trustworthy and that sensitive information is protected. Even if this sounds natural and straight forward, few attempts have been made to analyze in details what these expectations means from a security technology perspective in a realistic deployment scenario. In this paper we present a telecommunication virtualized platform provisioning scenario with two major stakeholders, the operator who utilizes virtualized telecommunication platform resources and the service provider, who offers such resources to operators. We make threats analysis for this scenario and derive major security requirements from the different stakeholders' perspectives. Through investigating a particular virtual machine provisioning use case, we take the first steps towards a better understanding of the major security obstacles with respect to platform service offerings. The last couple of years we have seen increased activities around security for clouds regarding different usage and business models. We contribute to this important area through a thorough security analysis of a concrete deployment scenario. Finally, we use the security requirements derived through the analysis to make a comparison with contemporary related research and to identify future research challenges in the area.

Contribution:

This paper provides a security analysis for a distributed system where virtual resources are provisioned to the remote users. The paper considers a more specific use case of IaaS cloud model and identifies various threats that undermine user trust in the cloud environments. The paper lists major security requirements of the stakeholders and also recommends a set of security mechanisms which are useful for establishing trust among cloud stakeholders

My Contribution:

I did the major work presented in this paper. The idea to write these research findings in a paper was mine. The paper was completely written by me except some parts in the Introduction section which were contributed by my co-supervisor, Christian Gehrman. Other than that, Christian also provided useful input to define the area and improve paper quality.

6.2 Securely Launching Virtual Machines on Trustworthy Platforms in a Public Cloud

Mudassar Aslam, Christian Gehrman, Lars Rasmusson, Mats Björkman. Securely Launching Virtual Machines on Trustworthy Platforms in a Public Cloud. *In 2nd International Conference on Cloud Computing and Services Science (CLOSER)*, 18-21 April 2012, Porto, Portugal.

Summary

In this paper we consider the Infrastructure-as-a-Service (IaaS) cloud model which allows cloud users to run their own virtual machines (VMs) on available cloud computing resources. IaaS gives enterprises the possibility to outsource their process workloads with minimal effort and expense. However, one major problem with existing approaches of cloud leasing, is that the users can only get contractual guarantees regarding the integrity of the offered platforms. The fact that the IaaS user himself or herself cannot verify the providerpromised cloud platform integrity, is

a security risk which threatens to prevent the IaaS business in general. In this paper we address this issue and propose a novel secure VM launch protocol using Trusted Computing techniques. This protocol allows the cloud IaaS users to securely bind the VM to a trusted computer configuration such that the clear text VM only will run on a platform that has been booted into a trustworthy state. This capability builds user confidence and can serve as an important enabler for creating trust in public clouds. We evaluate the feasibility of our proposed protocol via a full scale system implementation and perform a system security analysis.

Contribution

The main contribution of this paper is a secure VM launch protocol for a *user-bundled VM*. The paper focuses on the main research goal by enabling verifiable trust between the cloud user and provider. The proposed protocol in the paper leverages hardware security features provided by trusted computing to provide security assurances of the host platform to a remote party (i.e. cloud user). The research presented in this paper also includes a running prototype of the proposed VM launch protocol which validates its implementation feasibility.

My Contribution

The paper was mainly motivated and written by me with some contributions from my co-supervisor, Christian Gehrman. He also contributed to revise the launch protocol with his useful suggestions to make it comprehensive and secure. The research required to setup a trusted computing platform as well as all the implementation effort required to test a running prototype was done by me.

6.3 Trusted Launch of Virtual Machine Instances in Public IaaS Environments

Nicolae Paladi, Christian Gehrman, Mudassar Aslam, Fredric Morenius. Trusted Launch of Virtual Machine Instances in Public IaaS Environments. *In 15th Annual International Conference on Information Security and Cryptology*, 28-30 Nov 2012, Seoul, Korea.

Summary

Cloud computing and Infrastructure-as-a-Service (IaaS) are emerging and promising technologies, however their adoption is hampered by data security concerns. At the same time, Trusted Computing (TC) is experiencing an increasing interest as a security mechanism for IaaS. In this paper we present a protocol to ensure the launch of a virtual machine (VM) instance on a trusted remote compute host. Relying on Trusted Platform Module operations such as binding and sealing to provide integrity guarantees for clients that require a trusted VM launch, we have designed a trusted launch protocol for VM instances in public IaaS environments. We also present a proof-of-concept implementation of the protocol based on OpenStack, an open-source IaaS platform. The results provide a basis for the use of TC mechanisms within IaaS platforms and pave the way for a wider applicability of TC to IaaS security.

Contribution

The paper also focuses on secure VM launch protocol but considers the use of a *generic VM image*; which means that in addition to guarantee the VM launch on a trustworthy cloud platform, the user also needs a mechanism to ensure that the trusted VM image is used to instantiate the user VM. The solution proposed in this paper uses a trusted third party to ensure the launch of user VM, using a correct image (which can be verified by user after launch), on a trustworthy host. The protocol is also implemented in a OpenStack IaaS environment.

My Contribution

My main contribution in this paper was in the form of implementation principles needed for the selection of trustworthy host platforms. I implemented a library which provides functions for the trusted third party to attest the target platform and seal the user VM to it.

6.4 Security and Trust Preserving VM Migrations in Public Clouds

Mudassar Aslam, Christian Gehrman, Mats Björkman. Security and Trust Preserving VM Migrations in Public Clouds. *In 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 25-27 June 2012, Liverpool, UK.

Summary

In this paper we consider the security and trust implications of virtual machine (VM) migration from one cloud platform to the other in an Infrastructure-as-a-Service (IaaS) cloud service model. We show how to extend and complement previous Trusted Computing techniques for secure VM launch to also cover the VM migration case. In particular, we propose a Trust-Token based VM migration protocol which guarantees that the user VM can only be migrated to a trustworthy cloud platform. Different from previous schemes, our solution is not dependent on an active (on-line) trusted third party. We show how our proposed mechanisms fulfil major security and trust requirements for secure VM migration in cloud environments.

Contribution

The idea of labeling security profiles to the cloud platforms was not new. However, there was no practical mechanism of securely implementing such profiles. We introduced a trust credential called `Trust-Token` which is used to define and assign Trust Assurance Levels to the cloud platforms. Based on this new credential we proposed a secure VM migration scheme which addresses one of the major security concerns of the cloud users.

My Contribution

The requirement of secure VM migration was introduced by my co-supervisor, Christian Gehrman. I did all literature review, requirements formulation and suggested the concepts introduced for the solution. The migration protocol was proposed by me with some corrective feedback from Christian. The idea to write this paper was mine and I wrote the complete paper which was refined by the useful feedback of my both supervisors, Christian and Mats.

6.5 ASArP: Automated Security Assessment & Audit of Remote Platforms

Mudassar Aslam, Christian Gehrman, Mats Björkman. ASArP: Automated Security Assessment & Audit of Remote Platforms - using TCG-SCAP synergies. *In Journal of Information Security and Applications (JISA)* (to appear).

Summary

Many enterprise solutions today are built upon complex distributed systems which are accessible to the users globally. Due to this global access, the security of the host platforms becomes critical. The platform administrators use security automation techniques such as those provided by Security Content Automation Protocol (SCAP) standards to protect the systems from the vulnerabilities that are reported daily; furthermore, they are responsible for keeping their systems compliant to the relevant security recommendations (governmental or industry). Additionally, third party audit and certification process is used to increase user trust in the enterprise solutions. However, traditional audit and certification mechanisms are not *continuous*, that is, not frequent to deal with the daily reported vulnerabilities, and for that matter even auditors expect the platform administrators to keep the systems updated. As a result, the end user is also bound to trust the platform administrators about the latest state of the platform. In this paper we develop an automated security audit and certification system ($ASArP$) which can be used by the platform users or by the third party auditors. We use security automation techniques for continuous monitoring of the platform security posture and make the results trustworthy by using trusted computing (TCG) techniques. The prototype development of the $ASArP$ validates the implementation feasibility; it also provides performance benchmarks which show that the $ASArP$ based audit and certification can be done much more frequently (e.g. daily or weekly). The feasibility of $ASArP$ based continuous audits is significantly better than the traditional platform audits which are dependent on the physical presence of the auditors thus making the frequent audits much more expensive and operationally infeasible.

Contribution

This paper proposes first solutions (as best of our knowledge) to integrate TCG and SCAP frameworks in a way that the shortcomings of both these solutions are addressed. While the TCG mechanisms are used to make SCAP results trustworthy, the TCG remote attestation is enhanced with dynamic and real-time assessment and rating of the remote platform using SCAP. These techniques can be used for remote platform audit and certification purposes. We also provide a prototype implementation to validate implementation feasibility.

My Contribution

The research problem was identified by me and the paper writing was completely done by me with useful feedbacks from my supervisors in the author list. The paper first presented in the conference [12] got BEST PAPER AWARD which was later extended with implementation results. The writing of the extended paper as well as the prototype implementation was totally done by me.

6.6 Trusted Geolocation Aware Data Placement in Public Infrastructure Clouds

Nicolae Paladi, Mudassar Aslam, Christian Gehrman. Trusted Geolocation Aware Data Placement in Public Infrastructure Clouds. *In 13th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2014.

Summary

Reliance on third-party providers for cloud storage and cloud computing decouples data management from both data ownership and responsibility for correct data usage. A data owner loses control over the geographical placement of data once it is transferred to a cloud provider and earlier agreements become the only available tool to manage data placement. In some cases, transfer of sensitive data to other countries is illegal and while agreements can be a basis for compensation, they can only help once the damage is already done.

Geo-location of data in cloud environments matters for several reasons: tax levels may differ based on where a transaction is conducted (rather than where the entity is registered); compliance rules or privacy laws may require that certain categories of data are not stored or processed in a different jurisdiction; finally, a large research organization might conduct research which is considered illegal in some countries (e.g. stem cell research).

A practical solution for protection of data in cloud environments must consider its impact on functionality – which is a major driving force for adoption of cloud computing – such that distributed data processing capabilities are least affected. We propose a solution that uses geolocation data and trusted computing capabilities in order to allow data to be transferred in clear text and processed only in geographical locations approved by the data owner. The solution does not affect data processing capabilities of distributed object storages. We discuss an implementation example based on Swift, a known distributed object storage.

Contribution

This paper describes a protocol to securely store location information on cloud host platforms and later use this information to ensure that data is only available in plain text on platforms that are located in a certain geographical location. We provide a detailed implementation description of the above protocol, based on a popular cloud operating system and a known distributed object store.

My Contribution

The idea of writing paper on assurance of data geolocation was introduced by Nicolae. I provided technical details of the storing of platform geo-location in the TPM and mechanisms to use it later to ensure that the data is only available in correct jurisdiction. I also contributed in implementing and writing these suggestions provide by me.

Bibliography

- [1] Peter Mell and Timothy Grance. The NIST Definition of Cloud Computing. <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>, September 2011. [Online; accessed 30-Jan-2014].
- [2] Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, and Andrew Warfield. Xen and the art of virtualization. *SIGOPS Oper. Syst. Rev.*, 37:164–177, October 2003. <http://doi.acm.org/10.1145/1165389.945462>.
- [3] Qi Zhang, Lu Cheng, and Raouf Boutaba. Cloud computing: state-of-the-art and research challenges. *Journal of Internet Services and Applications*, 1(1):7–18, 2010.
- [4] Ronald B. Knode. Digital trust in the cloud - liquid security in cloudy places. http://assets1.csc.com/lef/downloads/Digital_Trust_in_the_Cloud.pdf, August 2009.
- [5] CircleID. Survey: Cloud Computing 'No Hype', But Fear of Security and Control Slowing Adoption. http://www.circleid.com/posts/20090226_cloud_computing_hype_security/, 2009.
- [6] Security's Cloud Revolution Is Upon Us - Understanding Information Security Amid Major Cloud Disruption. <http://www.forrester.com/Security+Cloud+Revolution+Is+Upon+Us/fulltext/-/E-RES99302>, 2013. [Online; accessed 18-Feb-2014].
- [7] Gordana Crnkovic. Constructive research and info-computational knowledge generation. In Lorenzo Magnani, Walter Carnielli, and

- Claudio Pizzi, editors, *Model-Based Reasoning in Science and Technology*, volume 314 of *Studies in Computational Intelligence*, pages 359–380. Springer Berlin Heidelberg, 2010. 10.1007/978-3-642-15223-8_20.
- [8] Mudassar Aslam and Christian Gehrman. Security Considerations for Virtual Platform Provisioning. In *ECIW '11: Proceedings of the 10th European Conference on Information Warfare and Security*, pages 283–290, UK, July 2011. The Institute of Cybernetics at the Tallinn University of Technology, Academic Publishing Limited.
- [9] Mudassar Aslam, Christian Gehrman, Lars Rasmusson, and Mats Björkman. Securely launching virtual machines on trustworthy platforms in a public cloud. In *CLOSER 2012 - Proceedings of the 2nd International Conference on Cloud Computing and Services Science, Porto, Portugal, 18-21 April, 2012*, pages 511–521. SciTePress, 2012.
- [10] Nicolae Paladi, Christian Gehrman, Mudassar Aslam, and Fredric Morenius. Trusted launch of virtual machine instances in public iaas environments. In Taekyoung Kwon, Mun-Kyu Lee, and Daesung Kwon, editors, *Information Security and Cryptology - ICISC 2012*, volume 7839 of *Lecture Notes in Computer Science*, pages 309–323. Springer Berlin Heidelberg, 2013.
- [11] Mudassar Aslam, Christian Gehrman, and Mats Björkman. Security and trust preserving vm migrations in public clouds. In *Trust, Security and Privacy in Computing and Communications (Trust-Com), 2012 IEEE 11th International Conference on*, pages 869–876, jun. 2012.
- [12] Mudassar Aslam, Christian Gehrman, and Mats Björkman. Continuous Security Evaluation and Auditing of Remote Platforms by Combining Trusted Computing and Security Automation Techniques. In *Proceedings of the 6th International Conference on Security of Information and Networks, SIN '13*, pages 136–143, New York, NY, USA, 2013. ACM.
- [13] Mudassar Aslam, Christian Gehrman, and Mats Björkman. ASArP: Automated Security Assessment & Audit of Remote Platforms - using TCG-SCAP synergies. *Journal of Information Security and Applications (to appear)*, XX(X):XXX – XXX, 2014.

- [14] Toby Velte, Anthony Velte, and Robert Elsenpeter. *Cloud Computing, A Practical Approach*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 2010.
- [15] Security guidance for critical areas of focus in cloud computing. <https://cloudsecurityalliance.org/guidance/csaguide.v3.0.pdf>, 2011.
- [16] Dan Hubbard and Michael Sutton. Top Threats to Cloud Computing. Technical Report Version 1.0, Cloud Security Alliance, March 2010.
- [17] TPM Specification, TPM Main Part-I Design Principles. http://www.trustedcomputinggroup.org/resources/tpm_main_specification, March 2011. [Online; accessed 25-Sep-2012].
- [18] TPM Specification, TPM Main Part-III Design Principles. <http://www.trustedcomputinggroup.org/resources>, July 2007. [Online; accessed 25-Sep-2012].
- [19] TCG Specification Architecture Overview. <http://www.trustedcomputinggroup.org/resources>, August 2007.
- [20] TPM 2.0 Library Specification – Part 1: Architecture (Committee Draft - Level 00 Revision 01.07). http://www.trustedcomputinggroup.org/resources/tpm_library_specification, March 2014. [Online; accessed 05-May-2014].
- [21] TPM 2.0 Library Specification – Part 2: Structures (Committee Draft - Level 00 Revision 01.07). http://www.trustedcomputinggroup.org/resources/tpm_library_specification, March 2014. [Online; accessed 05-May-2014].
- [22] TPM 2.0 Library Specification – Part 3: Commands (Committee Draft - Level 00 Revision 01.07). http://www.trustedcomputinggroup.org/resources/tpm_library_specification, March 2014. [Online; accessed 05-May-2014].
- [23] TPM 2.0 Library Specification – Part 4: Supporting Routines (Committee Draft - Level 00 Revision 01.07). http://www.trustedcomputinggroup.org/resources/tpm_library_specification, March 2014. [Online; accessed 05-May-2014].

- [24] TCG PC Client Platform TPM Profile (PTP) Specification (Committee Draft - Level 00 Revision 00.35). http://www.trustedcomputinggroup.org/resources/pc_client_platform_tpm_profile_ptp_specification, March 2014. [Online; accessed 05-May-2014].
- [25] David Waltermire, Stephen Quinn, Karen Scarfone, and Adam Halbardier. Security Content Automation Protocol (SCAP), NIST Special Publication 800-126, Version 1.2. <http://csrc.nist.gov/publications/nistpubs/800-126-rev2/SP800-126r2.pdf>, September 2011. [Online; accessed 12-Mar-2013].
- [26] G. Witte, M. Cook, M. Kerr, and S. Shaffer. *Security Automation Essentials - Streamlined Enterprise Security Management and Monitoring with SCAP*. McGraw-Hill Osborne Media, 2012.
- [27] Specification for the Extensible Configuration Checklist Description Format (XCCDF), Version 1.2. http://csrc.nist.gov/publications/nistir/ir7275-rev4/nistir-7275r4_updated-march-2012_clean.pdf, March 2012. [Online; accessed 25-Oct-2013].
- [28] Jonathan Baker, Matthew Hansbury, and Daniel Haynes. The OVAL Language Specification version 5.10.1. https://oval.mitre.org/language/version5.10.1/OVAL_Language_Specification_01-20-2012.pdf, January 2012. [Online; accessed 20-Apr-2013].
- [29] Towards a Common Enumeration of Vulnerabilities. <http://cve.mitre.org/docs/docs-2000/cerias.html>, January 1999. [Online; accessed 12-Mar-2013].
- [30] David Mann. An Introduction to the Common Configuration Enumeration (CCE). http://cce.mitre.org/documents/Introduction_to_CCE_White_Paper_July_2008.pdf, July 2008. [Online; accessed 12-Mar-2013].
- [31] Brant A. Cheikes, David Waltermire, and Karen Scarfone. Common Platform Enumeration: Naming Specification, Version 2.3. <http://csrc.nist.gov/publications/nistir/ir7695/>

- [NISTIR-7695-CPE-Naming.pdf](#), August 2011. [Online; accessed 12-Mar-2013].
- [32] The National Vulnerability Database. <http://nvd.nist.gov/>. [Online; accessed 30-Jan-2014].
- [33] Peter Mell, Karen Scarfone, and Sasha Romanosky. The Common Vulnerability Scoring System (CVSS) and Its Applicability to Federal Agency Systems. <http://csrc.nist.gov/publications/nistir/ir7435/NISTIR-7435.pdf>, August 2007. [Online; accessed 20-Mar-2013].
- [34] Karen Scarfone and Peter Mell. The Common Configuration Scoring System (CCSS): Metrics for Software Security Configuration Vulnerabilities. http://csrc.nist.gov/publications/nistir/ir7502/nistir-7502_CCSS.pdf, December 2010. [Online; accessed 20-Mar-2013].
- [35] Cloud Computing - Benefits, Risks and Recommendations for Information Security. European Union Agency for Network and Information Security (ENISA). <http://www.enisa.europa.eu/activities/risk-management/files/deliverables/cloud-computing-risk-assessment>, 2009. [Online; accessed 20-May-2014].
- [36] B.P. Rimal, Eunmi Choi, and I. Lumb. A taxonomy and survey of cloud computing systems. In *INC, IMS and IDC, 2009. NCM '09. Fifth International Joint Conference on*, pages 44–51, aug. 2009.
- [37] Chirag Modi, Dhiren Patel, Bhavesh Borisaniya, Avi Patel, and Muttukrishnan Rajarajan. A Survey on Security Issues and Solutions at Different Layers of Cloud Computing. *The Journal of Supercomputing*, 63(2):561–592, 2013.
- [38] Luis M. Vaquero, Luis Roderó-Merino, and Daniel Morán. Locking the sky: a survey on iaas cloud security. *Computing*, 91:93–118, January 2011.
- [39] S. Pearson and A. Benameur. Privacy, security and trust issues arising from cloud computing. In *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*, pages 693–702, Nov 2010.

- [40] Burton S. Kaliski, Jr. and Wayne Pauley. Toward risk assessment as a service in cloud environments. In *Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing*, HotCloud'10, pages 13–13, Berkeley, CA, USA, 2010. USENIX Association.
- [41] Frank Doelitzscher, Christoph Reich, Martin Knahl, and Nathan Clarke. Understanding Cloud Audits. In Siani Pearson and George Yee, editors, *Privacy and Security for Cloud Computing*, Computer Communications and Networks, pages 125–163. Springer London, 2013.
- [42] Nuno Santos, Krishna P. Gummadi, and Rodrigo Rodrigues. Towards trusted cloud computing. In *Proceedings of the 2009 conference on Hot topics in cloud computing*, HotCloud'09, Berkeley, CA, USA, 2009. USENIX Association.
- [43] Jeremy Rissi and Sean Sherman. *Cloud-Based IT Audit Process*, pages 15–32. John Wiley & Sons, Inc., 2011.
- [44] J.S. Park, E. Spetka, H. Rasheed, P. Ratazzi, and K.J. Han. Near-real-time cloud auditing for rapid response. In *Advanced Information Networking and Applications Workshops (WAINA), 2012 26th International Conference on*, pages 1252–1257, March 2012.
- [45] Marnix Dekker, Christoffer Karsberg, Mattina Lakka, and Dimitra Liveri. Auditing Security Measures - An Overview of Schemes. Technical report, European Union Agency for Network and Information Security (ENSIA), 2013.
- [46] Ryan K.L. Ko, BuSung Lee, and Siani Pearson. Towards achieving accountability, auditability and trust in cloud computing. In Ajith Abraham, JaimeLloret Mauri, JohnF. Buford, Junichi Suzuki, and SabuM. Thampi, editors, *Advances in Computing and Communications*, volume 193 of *Communications in Computer and Information Science*, pages 432–444. Springer Berlin Heidelberg, 2011.
- [47] Bernd Grobauer and Thomas Schreck. Towards incident handling in the cloud: Challenges and approaches. In *Proceedings of the 2010 ACM Workshop on Cloud Computing Security Workshop, CCSW '10*, pages 77–86, New York, NY, USA, 2010. ACM.

- [48] Rüdiger Glott, Elmar Husmann, Ahmad-Reza Sadeghi, and Matthias Schunter. Trustworthy clouds underpinning the future internet. In John Domingue, Alex Galis, Anastasius Gavras, Theodore Zahariadis, Dave Lambert, Frances Cleary, Petros Daras, Srdjan Krco, Henning Müller, Man-Sze Li, Hans Schaffers, Volkmar Lotz, Federico Alvarez, Burkhard Stiller, Stamatis Karnouskos, Susanna Avessta, and Michael Nilsson, editors, *The Future Internet*, volume 6656 of *Lecture Notes in Computer Science*, pages 209–221. Springer Berlin Heidelberg, 2011.
- [49] K.W. Ullah, A.S. Ahmed, and J. Ylitalo. Towards building an automated security compliance tool for the cloud. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2013 12th IEEE International Conference on*, pages 1587–1593, July 2013.
- [50] Nick Papanikolaou, Siani Pearson, Marco Casassa Mont, and Ryan KL Ko. Automating compliance for cloud computing services. In *CLOSER 2012 - Proceedings of the 2nd International Conference on Cloud Computing and Services Science, Porto, Portugal, 18-21 April, 2012*, pages 631–637. SciTePress, 2012.
- [51] Guide to Understanding FedRAMP. http://www.gsa.gov/graphics/staffoffices/Guide_to_Understanding_FedRAMP_060412.pdf, June 2012.
- [52] Recommended Security Controls for Federal Information Systems and Organizations, NIST Special Publication 800-53, Revision 3. http://csrc.nist.gov/publications/nistpubs/800-53-Rev3/sp800-53-rev3-final_updated-errata_05-01-2010.pdf, January 2010. [Online; accessed 30-Jan-2014].
- [53] Guide for Assessing the Security Controls in Federal Information Systems and Organizations, NIST Special Publication 800-53A. <http://csrc.nist.gov/publications/nistpubs/800-53A-rev1/sp800-53A-rev1-final.pdf>, June 2010. [Online; accessed 30-Jan-2014].
- [54] Fred H Cate. EU Data Protection Directive, Information Privacy, and the Public Interest. *Iowa L. Rev.*, 80:431, 1994.
- [55] Unleashing the Potential of Cloud Computing in Europe - Communication from the Commission to the European Parliament,

the Council, the European Economic and Social Committee and the Committee of the Regions. <http://ec.europa.eu/digital-agenda/en/european-cloud-computing-strategy>, 2012. [Online; accessed 20-May-2014].

- [56] Ron Knode and Doug Egan. Digital Trust in the Cloud - A Precis for Cloud Trust Protocol (CTP). http://assets1.csc.com/cloud/downloads/wp_cloudtrustprotocolprecis_073010.pdf, 2012.