Magdeburger Journal zur Sicherheitsforschung

Gegründet 2011 | ISSN: 2192-4260
Herausgegeben von Stefan Schumacher und Jörg Sambleben
Erschienen im Magdeburger Institut für Sicherheitsforschung

This article appears in the special edition »In Depth Security – Proceedings of the DeepSec Conferences«.
Edited by Stefan Schumacher and René Pfeiffer

Trusting Your Cloud Provider

Protecting Private Virtual Machines

Armin Simma

This article proposes an integrated solution that allows cloud customers to increase their trust into the cloud provider including cloud insiders (e.g. administrators). It is based on Mandatory Access Control and Trusted Computing technologies, namely Measured Boot, Attestation and Sealing. It gives customers strong guarantees about the provider's host system and binds encrypted virtual machines to the previously attested host.

Citation: Simma, A. (2015). Trusting Your Cloud Provider: Protecting Private Virtual Machines. *Magdeburger Journal zur Sicherheitsforschung*, *1*, 530–539. Retrieved June 17, 2015, from http://www.sicherheitsforschung-magdeburg.de/publikationen.html

Version 2015/06/16 20:01

1 Introduction

One of the top inhibitors for moving (virtual machines) to the cloud is security. Cloud customers do not fully trust cloud providers. The problem with sending virtual machines to the cloud is that »traditional« encryption is no solution because encrypted code cannot be executed. Taking a closer look at the numerous surveys about cloud adoption (and at inhibitors for NOT moving to the cloud) it can be seen that insider attacks are ranked in the top critical attacks. The insider in our scenario is the administrator of the provider's system or a user with high privileges.

A solution to this problem is based on (1) Trusted Computing (TC) technologies and (2) Mandatory Access Control (MAC). MAC is used to prevent the administrator - who must be able to access the host system for his tasks - from accessing virtual machines running on top of the cloud provider's system. Our solution is able to log all activities of users. Users (including the administrators) are not able to manipulate this log.

The former technology (TC) is used as a mechanism for giving the cloud customer a proof that the system hosting his virtual machines (= the cloud providers infrastructure) was not manipulated. The proof is hardware-based: it prevents several kinds of attacks e.g. rootkits or other BIOS-manipulating attacks. The proof is based on measuring all systems (system parts) that were executed since startup of the physical machine. Each part is measured before execution. This »measurement chain« is called Trusted Boot. A standardized tamper-resistant hardware called the Trusted Platform Module (TPM) plus a standardized protocol allows for the proof to the customer. The proof is called attestation.

A second technology used for securing the cloud is Trusted Computing's sealing mechanism. Sealing is an extension of asymmetric encryption: the decryption is done within the hardware (TPM) but only if the current measurement values are equal to predefined reference values. The reference values are defined by the cloud customer and specify a known good system plus system configuration. These technologies (Trusted Boot, Attestation, Sealing) allow the cloud customer to be sure that a specific (trustworthy) system is running on the provider's site.

In the rest of the article we will refer to this multi-technology solution as TRUMAC2, which stands for Trusted MAC-, Measured Boot- and Attestation-based Cloud.

1.1 Inhibitors for moving to the cloud

Cloud computing has rapidly gained acceptance in the last decade. Nevertheless many organizations and companies are still reluctant to move their data and services to the cloud. One of the top inhibitors for moving to the cloud is security. There are several surveys and studies confirming this: [2] states that »cloud security remains a major concern«. Other studies acknowledging this are: [3] [4]

In a typical (and simple) cloud scenario two basic entities can be identified: the cloud provider and the cloud customer (or subscriber). The customer does not fully trust the provider[1]. The (low-level) trust typically stems from the reputation of the cloud provider plus legal contracts like service level agreements (SLA). The customer has no way to verify the IT infrastructure (hardware and software) of the provider. If any verification is performed, an external auditor conducts it. The audit process, which results in a (compliance) certification, is not performed at the same time as the usage of the cloud by the customer. In this article a concept is presented that enables the cloud customer to get a »real-time« verification of the cloud provider's IT system without the need of an auditor visiting the site of the cloud provider.

1.2 Insider Attacks

Extending the simplified two-entities cloud scenario from above a third entity is introduced: the insider within the cloud provider's IT system which is shown in Figure 1. The provider and the insider are distinguished because the provider is trusted to some extent (by reputation, contracts, certifications etc.) but the trustworthiness level of the insider from the perspective of the customer is lower. Typically, in cloud scenarios, the insider is not known at all. Dinoor writes in [5] that clouds are »services, in which the consuming organization doesn't know the ›where‹ much less the ›who‹ of how the service is administered«.

The high risk of insider attacks is confirmed by several studies that enumerate insider attacks often performed by users with high privileges like administrators. [6][7][8]

The Computer Emergency Response Team (CERT) located at Carnegie Mellon University (CMU) shows a real case similar to a cloud computing scenario where a rogue administrator of a business partner was able to steal sensitive information [9]. The same web article from CMU CERT classifies cloud administrators: hosting company, virtual image, system and application administrators. This article deals with the first three classes. Application administrators are out of scope since we consider only Infrastructure as a service (IaaS) clouds and we do not consider the application level in this paper.

A category of attack, which is not enumerated in the CMU CERT article, is an attacker performing a privilege escalation attack [10] and thus gaining privileges. TRUMAC2 protects against privilege escalation, which is an attack vector with high frequency. »If your organization wants to gain an edge in stopping advanced attacks – start by locking down

[1] This article does not address the opposite direction - the provider being the trustor and the customer being the trustee or, in other words, the provider trusting the customer.

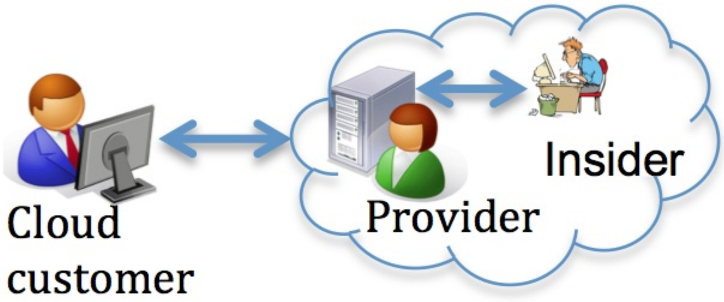

Figure 1: Three entities in a cloud scenario

the privileged pathway all attackers take.« [11]

A survey conducted recently shows that organizations are very concerned about insider attacks. 36% of the 707 respondents answered that they think to be »extremely vulnerable or vulnerable« to an insider attack whereas 27% answered to be »not at all or not very vulnerable«. [12]

This survey is not about insiders within cloud scenarios specifically but it shows that cloud computing makes the detection of such attacks more difficult because it »distributes sensitive data beyond internal IT control«.

This article proposes a methodology to give back control over sensitive data to the data owner who in the case being considered is the cloud customer.

Duncan enumerates specific subjects that are considered as insiders. The subjects of concern to this paper are the »insiders at the cloud provider side of the relationship« and the »external cloud service providers if resources are outsourced to other providers« [13]

If an insider is motivated to intentionally and adversely impact the goals of an organization the term malicious insider (MI) is used. The issue covered in this paper is the privileges given to insiders. If an insider is or was given specific privileges for accessing resources of the system and she is becoming malicious the consequences to the organization can be disastrous.

Administrators are powerful users in a modern IT (operating) system since they typically have full privileges to the operating system (OS) i.e. they have access to all (OS) commands and processes. Virtualization hypervisors allow administrators of the underlying host system or of the hypervisor - if a type 1 (baremetal) hypervisor is used - to access the guest images. In this paper a technology to prevent administrators of the host system/ hypervisor from accessing guest images is presented.

A presentation given by Gartner analyst Neil McDonald at »Security and Risk management Summit 2014« emphasized the importance of auditing cloud administrator activities. McDonald gave as a current practical example Amazon's AWS cloud. He said that the current auditing possibilities offered by Amazon to their cloud customers - called CloudTrail - are not enough for the customers. Amazon should provide »a view related to the activities of Amazon's systems administrators to know what they are doing«. The main concern of customers, he said, is how easy it is for cloud administrators to create snapshots of customers VMs.

»If data is so sensitive that snapshotting is keeping you up at night, don't put it in the cloud,« said MacDonald. [14]

1.3 Trust and the cloud

Trust is a big issue in cloud computing. In a talk at defcon 2009 H. Meer pointed at this problem: He said that the typical answer to the question about why we (the customer) should trust the cloud (provider) is a simple one: »Hey, you have to trust somebody«. [15] In other words: typical IT systems are implicitly trusted but this trust is often not based on facts (controls, procedures, SLAs) but it is a »blind« trust because of the reputation of the provider or developer of the system. Meer compares this to the trust in operating system (OS) vendors: consumers trust them implicitly e.g. that the OS vendors do not put a backdoor into their OS. Comparing this OS example with cloud infrastructures the difference is: the OS can be reverse engineered, which is hard to do but possible. H. Meer said, »reverse engineering keeps [big software companies] honest«. But how can the customer »reverse engineer« the cloud provider's infrastructure?

By using attestation technology this »blind« trust in cloud (providers) is over: The customer has the possibility to base his trust on facts and not only on the brand awareness level of some cloud provider.

H. Meer also talked about transparency - or strictly speaking about: missing transparency: He named Google App Engine as an example. In an interview given by Python originator and Google SW developer Guido van Rossum to cloudsecurity.com van Rossum did show complete non-transparency by not telling any internals about their cloud infrastructure. [16]

Andrew Baumann et al. list three entities that have to be trusted by the cloud customer: 1) the software at the provider's site which includes hypervisor, host

OS and firmware; 2) people working at the provider's site, most notably the administrator and 3) intelligence agencies and law enforcement bodies. [17]

The solution proposed in this paper helps to increase trust in 1) the software by attestation of the software including firmware. Concerning entity 2) the solution ensures confidentiality and integrity of guest images of the cloud customer by sealing the virtual images plus enforcing mandatory access control within the providers system: It prevents provider's staff including privileged users to access the customer's data and code. Assuming that the TPM is trusted - and not manipulated by any entity of category 3) - the proposed solution is also a remedy against replication of data by intelligence agencies.

Stephen Weis confirms this fact that Measured Boot and Attestation can defend »attacks« by intelligence agencies. [18]

2 Organizational controls to increase trust in the cloud (provider's system)

2.1 Privilege Identity Management

A well-established security principle is the principle of least-privilege. In cloud environments this principle is important. R. Glott et al. explain what it means to cloud systems: The different users including administrators should be granted only the privileges that are necessary for performing their defined tasks. R. Glott et al. give three examples for user accounts that should be separated in their roles: Infrastructure administrators, security administrators and customer employees. Each of the users has specific privileges. The account should not be granted more privileges than necessary. [19]

Privilege Identity Management deals with the organizational controls to manage privileged accounts. Enterprises or organizations need to control access to their resources. The users and processes have individual responsibilities and tasks within their work. To fulfill their task they need specific privileges. Some resources are shared between users i.e. access to the resource must be granted to these users at differing levels. An important step that is sometimes forgotten in the operational process of identity management is the revocation of user accounts if users change departments or leave the company. [20]

2.2 Auditing the infrastructure

When resources are outsourced Privilege Identity Management gets more complex: Shlomi Dinoor states it this way: « As corporations outsource to managed service, hosting and cloud providers, they increasingly cede direct control to someone else's privileged users [...]« [5]

Therefore Privilege Identity Management has to be extended with further activities or steps if data or processes are sent to the cloud. Two out of the 10 steps enumerated in Dinoors paper are listed here, because TRUMAC2 helps in performing these steps:

- Check that the provider has processes and procedures that fit into the customer's processes.
- The check should be accompanied with thorough audits. These audits can be performed by an external auditor or by the customer itself.

At present in practice a detailed audit of cloud »service's compliance are labor-intense, inconsistent, non-scalable, or just plain impractical to implement«. [21, p. 20] One of the reasons for this impracticability is that cloud providers do not allow performing audits on their infrastructure and that there is no transparency of the cloud infrastructure. Therefore in many cases third-party audits are used. On the other hand, such third-party audits are not enough for critical and/or sensitive data. [21]

TRUMAC2 supports cloud customers in performing a direct audit by allowing a »real-time« attestation of the provider's infrastructure.

A company, which has to fulfill compliance/regulatory requirements, needs the possibility to perform audits at each level including the infrastructure. If applications are running on the cloud, the cloud provider's system must be regularly audited. The problem with most cloud providers is: Although some cloud providers promote certifications and audits performed by external third-party auditors no provider allows audits defined and specified by a customer to be conducted at their infrastructure.

Attestation allows the customer to get guarantees about the provider's infrastructure. The disadvantage of attestation is the fact that a malicious user should not know internal details of the provider's system (e.g. host OS version) because known vulnerabilities of a specific system can be exploited if the system is known. Sadeghi and Stüble introduced property-based attestation, which allows attesting properties and not binary code.[22] Further details and application within cloud scenarios can be found in [23] [24] [25] .

3 Base technologies for TRUMAC2: Measured Boot, Attestation, MAC and Sealing.

IaaS clouds are based on virtualization technology. Therefore a technology that has to be taken into account when protecting sensitive data within cloud is virtualization. Eric Siebert shows in a web article how easy it is to steal virtual machines. [26] If the virtual machine contains sensitive data this scenario must be completely avoided. Siebert proposes different countermeasures to this problem: One solution is encryption of the sensitive data, another is logging and auditing of all activities and commands executed by administrators. TRUMAC2 considers and incorporates both of these measures: tamper-resistant auditing through measured boot, remote attestation and MAC; encryption through sealing. These underlying technologies are described in the following sections.

3.1 Trusted Computing: Measured Boot and Integrity Measurement

[27] describes a process called Trusted Boot (also Measured Boot), which is based on Integrity Measurement. The principle is that each part of the system is measured before it is executed. Measuring is performed by hashing the binary code. The measured results are stored in platform configuration registers (PCR) within the TPM. When PCR values are written to, the old content is preserved by concatenating the old PCR value with the new data and hashing the concatenation, i.e. PCR_new = SHA1 (PCR_old || data).

Figure 2 shows the integrity measurement process: It starts with the core root of trust for measurement (CRTM), which is part of the hardware, TPM and therefore an implicitly trusted part. The CRTM measures the BIOS (1), stores the value in a PCR of the TPM (2), after which execution is transferred to the BIOS. This so-called measurement chain continues by measuring the bootloader (3), saving the hash to the TPM (4), executing the bootloader (5) and then consecutively continues the measuring / saving/ executing - process with each component in the system. [28]

3.2 Trusted Computing: (Remote) Attestation

After measurement, it must be possible to securely get the PCR values from the TPM. [27] calls this process Integrity Reporting. Various cryptographic keys are embedded within TPM chips. These keys allow for integrity reporting and remote attestation. Remote attestation is based on digitally signing the PCR values. By using remote attestation a remote entity gets a proof that the received PCR values are from a tamperproof genuine TPM, i.e. a guarantee about the integrity of the value.

Figure 3 shows the simplified process of integrity reporting/attestation: The challenger is the entity that requests the PCR values. It creates a nonce using a random number generator (RNG). This nonce is sent to the TPM. The TPM concatenates the nonce and the value of the requested PCR and hashes the result. This hash is signed using a signature key, which is generated from the Storage Root Key. The signature is sent back to the challenger. The challenger creates a hash from the concatenated nonce and reference PCR value. If the reference hash and the received hash are the same, the integrity of the PCR value is verified. [28]

3.3 Mandatory Access Control

Mandatory access control (MAC) is an access control model that does not allow the end user to change access control policies, as opposed to discretionary access control (DAC). Within DAC the owner of a resource is allowed to change the policy. MAC policies are defined by the system and cannot be bypassed. As a consequence MAC is a strong access control model because »any user, even with administrator privileges, is constrained by the defined policy.«[29] A similar statement can be found in [30]: root users cannot compromise the policy and thus the security of the system within a MAC-controlled system. An implementation that demonstrates the robustness of MAC is [31]. In their system an attacker would have to successfully bypass the MAC mechanism to become root but Briffaut et al. write »during one year and a half of deployment, we never observed such an attack.« Another evidence that SELinux makes (virtualization) systems more secure was given at Black Hat 2011 by the presentation of Nelson Elhage [32]. Elhage showed that exploit CVE-2011-1751 allowed an attacker to break from a VM to the host system but he stated that the attack he showed is not possible if sandboxed using SELinux.

Concerning the solution described in this paper it is important to note that changing the policy requires a reboot of the system.

Concrete implementations of MAC within operating systems like SELinux allow fine-grained policies. The problem with implementing MAC policies is the complexity of policies.

An integration of MAC (SELinux) into virtualization and cloud environments is svirt. [33]

In our implementation of the prototype we did not confine root but we did log each activity including the activities of root. To change the SELinux policy a reboot of the OS is necessary. A reboot of a physical machine - in our scenario at the cloud provider site - will easily be detected. After each reboot a new attestation should be performed. This restarted attestation would reveal that the SELinux policy was changed and therefore the system is not trusted any more.

A technology to prevent privilege escalation attacks is described in [34]. This technology could be incorporated in our system and would provide a further level of defense. In high-assurance infrastructures the root account - which is able to define access control policy - should be protected by either multi-factor authentication and/or four eyes principle.

The technical details of the MAC policy of TRUMAC2 can be found in the master thesis of Philipp Rusch. [35]

A system that is similar to TRUMAC2 with respect to the access control technology is described in [36]. Shamon is based on IMA, XEN, remote attestation and MAC - the same technologies we used for TRUMAC2. The difference is: Shamon is not built specifically for cloud scenarios. The goal of shamon is not to confine root user but to control inter-VM communic-

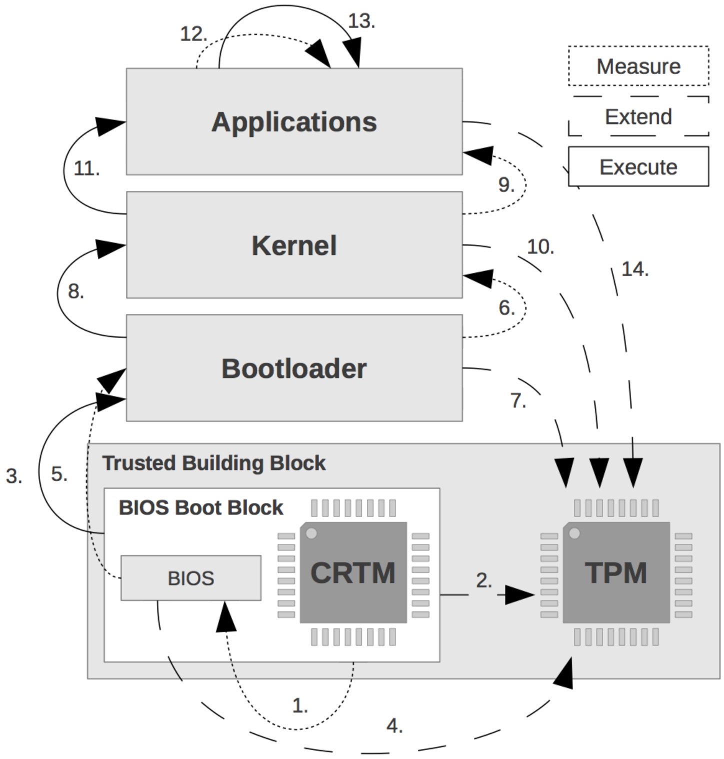

Figure 2: The measurement flow [28]

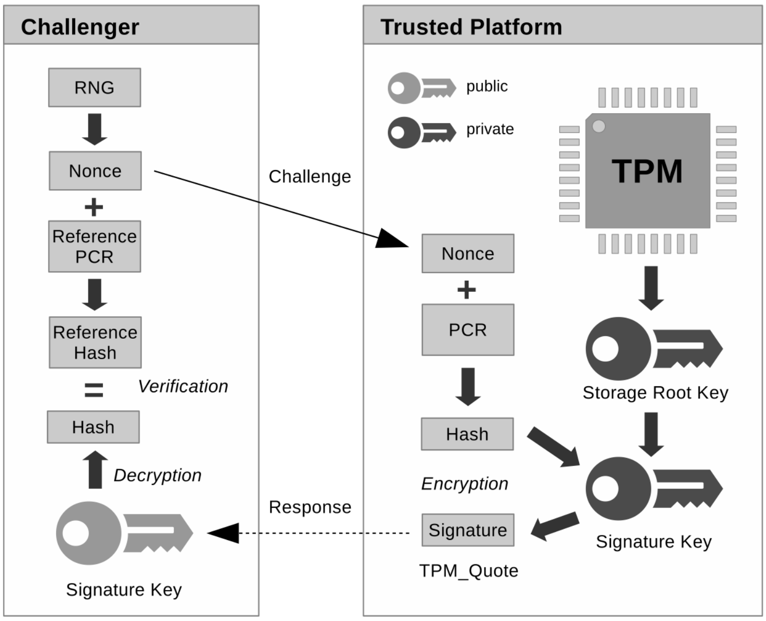

Figure 3: Remote Attestation [28]

ation within distributed processing systems like computing grids. Shamon also aims at keeping the trusted code base minimal.

3.4 Sealing

If the provider implements the technologies described before (TC and MAC) there is still the problem of the transfer of the virtual image: the customer has to send the virtual machine to the provider, which is done in all but a handful of cases via network/internet. For confidentiality of the image cryptography could be applied: Customers encrypt the VM with a key, the provider decrypts the VM with the same symmetric key - if symmetric encryption is used - or with the private key - in case of asymmetric encryption. The problem here is: the customer has to trust the provider that he keeps the key secret. This traces back to the initial problem: the customer has to trust the provider including its staff (e.g. network or system administrator).

A solution to this problem is to apply another technology of TC: sealing. Sealing is based on, first, a private key stored within the TPM and, second, on performing decryption solely in the case that PCR contents have specific values. The private key never leaves the TPM; decryption of data encrypted with the corresponding public key is done - without any exception - within the TPM. By sealing the decryption process to pre-set PCR contents the cloud customer knows that it's virtual machine is only decrypted on a trusted system.

Sealing is a countermeasure against the time-of-check-time-of-use attack [37] as well as against privileged users that are able to record network traffic from/into the cloud provider's system.

4 The organizational trust ecosystem for TRUMAC2

To apply the technologies from the preceding subsections (trusted boot, integrity measurement, reporting, attestation and MAC) within a cloud scenario a trusted third party (TTP) is necessary. This TTP has several tasks within the whole ecosystem: (1) It provides the certificates which are used for the digital signatures used during attestation; (2) It uses - or provides - a reference system out of which hash values are to be created. These reference hashes are compared with the PCR values received during attestation.

In practice there is a third task, which can be performed by the same TTP or by another entity. We call this entity trusted third tester (TTT).

Figure 4 describes the ecosystem: The host OS and/or VMM is developed in (1). TTT tests the host OS and hypervisor or ideally formally verifies the hypervisor (2). The latter case requires a small hypervisor. TTT publishes reference integrity values (3). The hypervisor is running on the hardware of the cloud provider, which embeds a TPM (4). Special Software (Cloud Attestor and Advisor, CAA) is running at the customer site. This software performs the attestation process in (5) and (6) and compares the received PCR values with the signed integrity values. If the hashes are equal the customer has a proof about the system running on the provider's site. This proof plus thorough tests and audits in (2) is the base of the trust in the cloud provider('s system).

5 Related Technologies

5.1 Software Guard Extensions (SGX)

Intel recently developed a technology called software guard extensions (SGX) that allows creating so called enclaves. [38] Enclaves are isolated areas that are protected from code outside the enclave - even from privileged code if external to the enclave. Enclaves can be seen as reverse sandboxes: it is not possible to access the enclave from outside. Andrew Baumann et al. have built a system based on SGX that has the same goal as TRUMAC2: building a trusted cloud. In their system, which they call »Haven«, code and data is protected from »outside«. »Outside« includes the platform on which the code runs. [17] Such a platform could be the virtualization host of the cloud provider. Haven is a very promising alternative technology to build a secure and trustworthy cloud infrastructure.

5.2 Mt. Wilson and Trusted Execution Technology (TXT)

In [21] a technology that is similar to TRUMAC2 is described. The codename of the component that is similar to TRUMAC2 is Mt.Wilson. The book goes into great detail of the technologies described in this article (Measured Boot, Integrity Measurement, Attestation, Trusted Computing). If you need more (detailed) information - more than we can give in this short article - the book is definitely worth reading.

The difference between TRUMAC2 and Mt. Wilson is, first, Mt. Wilson does not use MAC specifically and, second, only parts of the Mt. Wilson software is licensed as open source. TRUMAC2 on the other hand is completely open source.

5.3 Homomorphic Encryption

Classical encryption of virtual machines sent to the cloud is only possible if the provider knows the decryption key because virtual machines have to be executed on the provider's hardware.

The goal of homomorphic encryption is to execute encrypted code or, in other words, performing a (binary) operation on encrypted data where only the owner of the secret key can decrypt the result of the operation. The concept is known since the 1970ies but the first

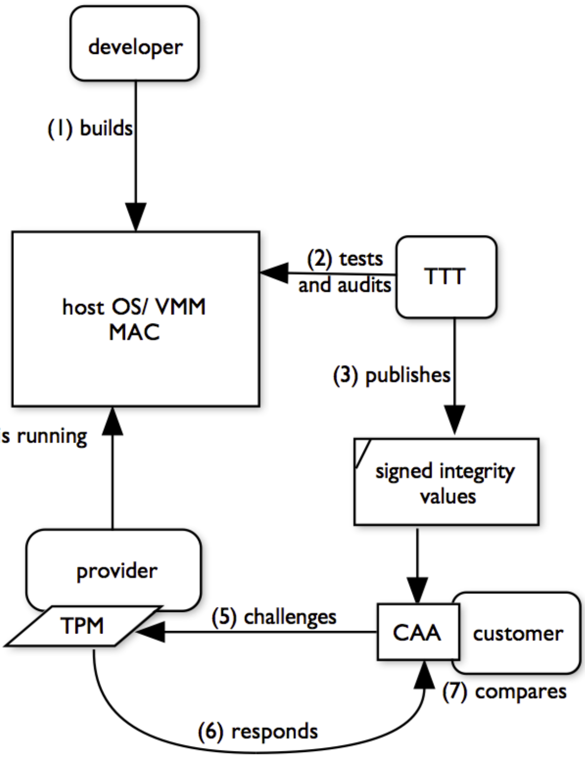

Figure 4: The trust ecosystem [28]

efficient homomorphic encryption scheme was published by Gentry [39]

Shoup and Halevi have implemented an open source library for HE [40] but they write in the accompanying Readme.md that it »is mostly meant for researchers working on HE and its uses«.

6 Conclusion

Since most of the burden to implement the described controls is put on the cloud provider the question arises about the benefit to the provider. The benefit for the cloud customers is clear: they can attest the provider's infrastructure and use the attestation for compliance or other required documentation/ certification. The benefit for the provider is to gain more competitiveness. As Dinoor writes in [5]: »By not effectively demonstrating strong privilege control policies and processes [...] service providers could potentially leave a lot of business on the table.«

In recent years and especially in the last year enterprises have realized the importance of enabling customers to trust the cloud. Many enterprises have committed to building trusted cloud solutions.

When we performed a literature and web research within the area of trusted cloud (Trusted Computing technologies like attestation applied to the cloud) in 2012 we could not find any cloud provider or cloud solution with TC technologies incorporated in practice. At that time it was only discussed in and dealt within academic research.

Currently (October 2014) things have totally changed: practical solutions incorporating TC technologies (e.g. attestation, Intel TXT) for making clouds more trustworthy have sprung up in the last year. Examples for commercial solutions are [41] [42] [43] [44] [45] [46] A solution using open source software is described in [47]. Several projects that have a similar or the same goal as TRUMAC2 have been supported by EU research funds: tclouds[2], a project with 14 partners, has been successfully completed; the goal of SECCRIT[3] is to build a cloud that can be used within critical infrastructures.

All these activities give hope that the cloud will finally be based on a secure and trusted infrastructure, that cloud customers can send sensitive data to the cloud without any concerns about security and that cloud solutions can be used for application within critical infrastructures.

7 About the Author

Armin Simma is Hochschullehrer (german for: professor) at the Vorarlberg University of Applied Sciences, Austria. After graduating in Computer Science at the University of Linz, Austria he worked for two years at CERN, the European Organization for Nuclear Research in Switzerland. Since 2001 he is teaching and doing research in the areas of IT security, virtualization, cloud technology, operating systems and computer networking. He can be reached via email (armin.simma@fhv.at) or researchgate.net

[2] www.tclouds-project.eu

[3] www.seccrit.eu

8 References

[1] Mikael Eriksson, Makan Pourzandi, and Ben Smeets, »Trusted computing for infrastructure,« Ericsson, Vol. 91, Oct. 2014.

[2] B. Rahul, Maureen, and S. Scott, »Alert Logic Cloud Security Report,« Alert Logic, Houston, TX, 2014.

[3] »Cloud Adoption Report,« Bitglass.com, Campbell, CA, Apr. 2014.

[4] »Future of Cloud Computing Survey,« northbridge.com, 2014.

[5] S. Dinoor, »Feature: Privileged Identity Management: Securing the Enterprise,« Netw Secur, vol. 2010, no. 12, pp. 4–6, Dec. 2010.

[6] Heidi Shey and et al., »Understand The State Of Data Security And Privacy: 2013 To 2014,« Forrester, Oct. 2013.

[7] K. Mickelberg and N. Pollard, »US cybercrime: Rising risks, reduced readiness. Key findings from the 2014 US State of Cybercrime Survey,« PwC, CSO Magazine, CERT SEI/CMU, Jun. 2014.

[8] »Insider Threat Survey Report,« spectorsoft.com, Vero Beach, FL, 2014.

[9] B. Claycomb and A. Nicoll, »Insider Threats Related to Cloud Computing,« Aug-2012. .

[10] V. Igure and R. Williams, »Taxonomies of attacks and vulnerabilities in computer systems,« IEEE Commun. Surv. Tutor., vol. 10, no. 1, pp. 6–19, 2008.

[11] »2014 Global Advanced Threat Landscape,« CyberArk, Jul. 2014.

[12] J. Oltsik, »Vormetric / ESG Insider Threats Survey,« Vormetric, The Enterprise Strategy Group, Milford, MA, Sep. 2013.

[13] A. Duncan, S. Creese, and M. Goldsmith, »An overview of insider attacks in cloud computing,« Concurr. Comput. Pract. Exp., Mar. 2014.

[14] E. Messmer, »Gartner: Best practices for Amazon AWS security,« networkworld.com, Jun. 2014.

[15] H. Meer, »Clobbering the Cloud,« presented at the defcon, Las Vegas, NV, 2009.

[16] C. Balding, »Cloudsecurity.org Interviews Guido van Rossum: Google App Engine, Python and Security,« Jul-2008.

[17] A. Baumann, M. Peinado, and G. Hunt, »Shielding Applications from an Untrusted Cloud with Haven,« in 11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14), Broomfield, CO, 2014, pp. 267–283.

[18] S. Weis, »Protecting Data In-Use from Firmware and Physical Attacks,« presented at the Black Hat, Las Vegas, NV, 2014.

[19] R. Glott, E. Husmann, A.-R. Sadeghi, and M. Schunter, »Trustworthy Clouds Underpinning the Future Internet,« in The Future Internet, vol. 6656, J. Domingue, A. Galis, A. Gavras, T. Zahariadis, D. Lambert, F. Cleary, P. Daras, S. Krco, H. Müller, M.-S. Li, H. Schaffers, V. Lotz, F. Alvarez, B. Stiller, S. Karnouskos, S. Avessta, and M. Nilsson, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 209–221.

[20] E. Cole, »Three scary, but true, security tales,« Oct. 2014.

[21] R. Yeluri and E. Castro-Leon, Building the Infrastructure for Cloud Security: A Solutions View. Apress, 2014.

[22] A.-R. Sadeghi and C. Stüble, »Property-based attestation for computing platforms: caring about properties, not mechanisms,« in Proceedings of the 2004 workshop on New security paradigms, 2005, pp. 67–77.

[23] L. Chen, R. Landfermann, H. Löhr, M. Rohe, A.-R. Sadeghi, and C. Stüble, »A protocol for property-based attestation,« in Proceedings of the first ACM workshop on Scalable trusted computing, 2006, pp. 7–16.

[24] S. Xin, Y. Zhao, and Y. Li, »Property-Based Remote Attestation Oriented to Cloud Computing,« 2011, pp. 1028–1032.

[25] A. Nagarajan, V. Varadharajan, M. Hitchens, and E. Gallery, »Property Based Attestation and Trusted Computing: Analysis and Challenges,« 2009, pp. 278–285.

[26] E. Siebert, »How to steal a virtual machine and its data in 3 easy steps,« Jan. 2010.

[27] »TCG specification architecture overview, Rev.1.4,« trusted computing group, 2007.

[28] A. Simma and P. Rusch, »Retaining Control Over Private Virtual Machines Hosted by a Cloud Provider Using Mandatory Access Control, Trusted Boot and Attestation,« in Proceedings of the 13th European Conference on Cyber Warfare and Security, Piraeus, Greece, 2014, pp. 172–180.

[29] M. Blanc and J.-F. Lalande, »Improving Mandatory Access Control for HPC clusters,« Future Gener. Comput. Syst., vol. 29, no. 3, pp. 876 – 885, 2013.

[30] M. Blanc, A. Bousquet, J. Briffaut, L. Clevy, D. Gros, A. Lefray, J. Rouzaud-Cornabas, C. Toinard, and B. Venelle, »Mandatory Access Protection Within Cloud Systems,« in Security, Privacy and Trust in Cloud Systems, S. Nepal and M. Pathan, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 145–173.

[31] J. Briffaut, J.-F. Lalande, and C. Toinard, »Security and Results of a Large-Scale High-Interaction Honeypot,« JCP, vol. 4, no. 5, pp. 395–404, 2009.

[32] N. Elhage, »Virtunoid: Breaking out of KVM,« presented at the Black Hat, 2011.

[33] J. Morris, »sVirt: Hardening Linux Virtualization with Mandatory Access Control,« presented at the Linux.conf.au, Hobart, Australia, 2009.

[34] A. Chatterjee and A. Mishra, »Securing the Root Through SELinux,« in Intelligent Computing, Networking, and Informatics, vol. 243, D. P. Mohapatra and S. Patnaik, Eds. Springer India, 2014, pp. 653–

659.

[35] P. Rusch, »Trusted Boot und Mandatory Access Control: Vertrauenswürdige Ver- und Bearbeitung von sensiblen und privaten Prozessen und Daten in Fremdsystemen, wie z.B. Cloud-Umgebungen,« Master's Thesis, University of Applied Sciences Vorarlberg, Dornbirn, Austria, 2014.

[36] J. M. McCune, T. Jaeger, S. Berger, R. Caceres, and R. Sailer, »Shamon: A System for Distributed Mandatory Access Control,« in Computer Security Applications Conference, 2006. ACSAC '06. 22nd Annual, 2006, pp. 23–32.

[37] E. Shi, A. Perrig, and L. Van Doorn, »BIND: A Fine-Grained Attestation Service for Secure Distributed Systems,« 2005, pp. 154–168.

[38] I. Anati and et. al, »Innovative Technology for CPU Based Attestation and Sealing,« Intel, Aug. 2013.

[39] C. Gentry, »Fully Homomorphic Encryption Using Ideal Lattices,« in Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing, New York, NY, USA, 2009, pp. 169–178.

[40] Shai Halevi and Victor Shoup, »Design and Implementation of a Homomorphic-Encryption Library,« Nov. 2013.

[41] L. Yan and S. Ye, »End-to-End Trusted Cloud For China,« presented at the Intel Developer Forum 2014, 2014.

[42] »Huawei Unveils Servers at IDF 2014: Joint Innovation for a Win-Win Future,« Huawei, Apr. 2014.

[43] »Trusted Workload Migration with EMC, RSA, Intel, and HyTrust,« EMC, RSA, Intel, HyTrust, 2013.

[44] »IBM and Intel Bring New Security Features to the Cloud,« Softlayer; IBM, Sep. 2014.

[45] J. Greene, »Sometimes Trust is Not Enough: Intel TXT Advances to the Next Stage at IDF,« 15-Sep-2014.

[46] M. Trouard-Riolle, »Citrix XenServer Powered Trusted VMs in OpenStack Clouds,« 08-Sep-2014. .

[47] Christian Huebner, »Trusted Cloud computing with Intel TXT: The challenge,« Apr-2014.