LIGHTWEIGHT DISTRIBUTED ATTESTATION FOR THE CLOUD

Martin Pirker, Johannes Winter and Ronald Toegl
Institute for Applied Information Processing and Communications (IAIK),
Graz University of Technology, Inffeldgasse 16a, 8010 Graz, Austria

Keywords: Cloud Node Security, Trusted Computing, Trusted Execution Technology.

Abstract: Moving local services into a network of Cloud nodes raises security concerns, as this affects control over data and code execution. We leverage the Trusted Platform Module and Trusted Execution Technology of modern platforms to detect malicious Cloud nodes running untrusted software configurations. To achieve this, we propose a node-to-Cloud join protocol that enforces remote attestation.

1 INTRODUCTION

Running conventional, local IT services implicitly offers full control over the hardware as well as the software setup. However, the promises of more efficiently utilized IT resources and reduced costs cause more and more services to be migrated to *Cloud Computing* providers. This leaves one with nothing but the service level contractually agreed with the Cloud provider. Eventually, Cloud datacenter security may fail and expose clients' data or code. This may happen e.g. through human error, mailicious intent, or in lawful compelled-assistance scenarios. Such leakage of customer sensitive data may prove fatal for a business, or a least cause intervention by data protection authorities.

In this paper, we address the loss of control on remote data and code execution in the Cloud beyond conventional measures such as automated 24x7 network monitoring, intrusion detection technologies, defined life-cycles for storage media, physical access restrictions with multi-factor identification, or armed guards. To raise the challenge for attackers, our design integrates Trusted Computing technologies into the Cloud formation phase, which enables us to remotely assess and report the security state of the connecting and running Cloud nodes. This enables a Cloud made of nodes with known-good configurations only, while keeping the protocol overhead limited.

While there is no absolute security, we believe our proposal offers an interesting trade-off between physical control and security properties for local versus distributed data processing.

Outline. The remainder of the paper is structured into the following major sections. Section 1 starts with a brief motivation of the main topic discussed in this paper. Section 2 offers a brief background summary of the capabilities of Trusted Computing technologies. We then continue in Section 3 to present our architecture, along with the core Cloud join protocol. We discuss additional security limitations and trade-offs in Section 4. In Section 5 we present links to related work. Finally, Section 6 concludes the paper.

2 SECURITY ENHANCED MASS-MARKET COMPUTING PLATFORMS

Over the last years, mass-market consumer computer platforms and devices were enhanced with dedicated functions to support advanced security concepts. In the following we give a short introduction on the features available. We identify a set of features which can be put to good use in the design of global trusted infrastructures, where every *computing node* in the Cloud is based on a modern security enhanced hardware platform.

The ever evolving industry standard PC platform introduced hardware features in the last years which allow to realise enhanced security for specific scenarios. The concept of Trusted Computing as promoted by the Trusted Computing Group (TCG) extends the industry standard PC architecture with a specialised hardware component, the Trusted Platform Module (TPM) (Trusted Computing Group, 2007b).

Pirker M., Winter J. and Toegl R..
LIGHTWEIGHT DISTRIBUTED ATTESTATION FOR THE CLOUD.
DOI: 10.5220/0003929105800585
In *Proceedings of the 2nd International Conference on Cloud Computing and Services Science* (CLOSER-2012), pages 580-585
ISBN: 978-989-8565-05-1

A TPM features cryptographic primitives similar to a smartcard, but is physically bound to its host platform. The tamper-resilient chip provides functions for public-key cryptography, key generation, cryptographic hashing, random-number generation, and others. With this hardware crypto support, and being a chip operating independently from other devices, the TPM can provide certain trusted functions.

An important concept of Trusted Computing is the measurement, logging and reporting of the platform state. Upon platform hardware reset a special set of platform configuration registers (PCRs) in the TPM are reset to a well defined start value. PCRs cannot be directly written to. Rather, a PCR with index i, $i \geq 0$ in state t is extended with input x by setting $PCR_i^{t+1} = \text{SHA-1}(PCR_i^t || x)$. This enables the construction of a chain-of-trust. From the BIOS onwards, every block of code is measured into a PCR before execution control is passed to it. Thus, the current values in the set of PCRs represent a trace of what happened since system reboot, up to the current state of the system.

The TPM can *bind* data to a platform by encrypting it with a *non-migratable* key, which never leaves the TPM protection. An extension to this is *sealing*, where a key may only be used with a specific, *trusted* PCR configuration. Thus, decryption of sealed data can be restricted to an expected state – running software and configuration – of the computer.

The current state may also be TPM signed with the TPM *Quote* operation and reported in a so-called *remote attestation* protocol (Coker et al., 2008; Sadeghi and Stüble, 2004; Trusted Computing Group, 2007a). To protect the platform owner's privacy, a pseudonym identity must be used: an *Attestation Identity Key (AIK)*. The authenticity of an AIK can be certified either by an on-line trusted third party, called PrivacyCA or by applying the group-signature-based DAA scheme (Brickell et al., 2004). Then, a remote verifier may analyze the Quote result and decide whether to trust the given configuration or not.

TPMs also provide a limited amount of non-volatile memory (NV-RAM) to store user- or owner-supplied information. One specific piece in NV is the TPM *Endorsement Key* (EK). It is a unique asymmetric RSA keypair of which the private part never leaves the TPM in clear. An accompanying certificate – typically signed by the manufacturer – documents the fact that the key belongs to a real hardware TPM on a trusted computing platform. It can also serve as a unique identification of a platform.

Recent hardware platforms from Intel, with Intel Trusted Execution Technology (TXT)[1], extend the basic TCG model of a *static* chain-of-trust from hardware reboot and trust rooted in early BIOS. They provide the option of a *dynamic* switch to a well-defined, measured system state (Grawrock, 2009), meaning at any point of execution after platform reboot. This is called a *dynamic root of trust for measurements (DRTM)*. Consequently, this capability significantly cuts down the complexity of the chain-of-trust measurements to assess the platform state by excluding the early, messy bootup operations.

[1]We restrict our discussion to Intel's Trusted Execution Technology (TXT) as this is currently the dominant technology provider – comparable features are also available on e.g. AMD platforms.

3 SCENARIO AND ARCHITECTURE

We now outline the scenario we base our secure Cloud formation approach on. The proposed architecture takes advantage of the platform security technologies reviewed in the previous section. First we identify the core assumptions and properties we want to achieve, then we show how they can be implemented with the use of Trusted Computing features, data structures and protocols.

3.1 Cloud Node Properties

An architecture of networked nodes necessitates trade-offs between multiple degrees of freedom for the design. One cannot have absolute control while at the same time assume having no ongoing maintenance work. Also, uniform deployment often is contrary to the customizability of software configurations. Our architecture strives for the following properties:

- **Distributed.** We assume that the individual nodes in the Cloud are distributed both geographically and organizationally. Nodes could be placed in different countries and continents, and could be owned and operated by diverse sets of operators.
- **Attested.** Due to wide node distribution it is consequently difficult to enforce conventional security oversight on the nodes. Instead of absolute physical and organizational control we use the Trusted Computing technique of *remote attestation* to enforce an assessment process. The result of this protocol is the decision whether a remote node is in a trusted state and therefore allowed to become part of the Cloud, or not.

- **Lightweight.** In order to enable a diverse set of distributed stakeholders to be able to participate in the Cloud, the installation of a Cloud node and joining the Cloud network should be a simple setup and maintenance task that does not add significant organizational overhead.
- **Heterogeneous.** The larger the Cloud network, the more resources may be shared among participants. Consequently, this allows to process larger tasks and improves the economics of scale which reduces costs. Thus, as many different nodes as possible should be able to join the Cloud. Consequently, the Cloud node software should build on a base or primitives which are easily portable to different platforms.

3.2 Cloud Control

A Cloud infrastructure connects many computing nodes. Still, for being part of one specific Cloud (network) there must be a central Cloud management service – we call this *Cloud control*.

In our architecture the role of the Cloud provider is to be this centralized management. The provider's always-online, professionally run 24x7 datacenter manages client profiles and runs accounting tasks. It provides an information service on available Cloud nodes and service capacities. However, to profit from the economics of scale, the Cloud provider only contributes a few "last resort" processing nodes to the Cloud. The vast majority of nodes is expected to be run distributed, at remote sites, and their operators may not be under direct provider supervision.

3.3 Computing Nodes

To provide a certain level of security assurance, a minimum barrier to overcome is that Cloud node hardware platforms must offer the required Trusted Computing security features. These are implemented in a specific Cloud node *software image* which can be downloaded from central Cloud control and then booted at a node. If a connection to Cloud control can be established and only if the new node's security is successfully attested, this node becomes available to the Cloud.

We further expect every node to host a TPM as specified by the TCG (see Section 2). The Cloud node software image is booted by a trusted boot process, meaning a mechanism exists which enforces measurement of system state into the TPM PCRs, starting from a well-defined initial platform state. The combination of well-known software images and the deterministic TXT boot process allows the determination of trusted platform configurations.

Local Storage. From a security point of view unencrypted temporary storage on node local mass-media (e.g. harddisc) may not be acceptable. Consequently, any node-local temporary data storage should be fully encrypted, e.g. with a transparently encrypted file system under a symmetric key. Thus, to re-access the node storage after service interruption (e.g. reboot) the key for the local temporary storage must only be made available to the identical Cloud node software image running at the same Cloud node. This challenge can be solved by encrypting the storage key with an asymmetric TPM key *sealed* to a specific Cloud node software image.

For a full automatically (re-)bootable Cloud node we identify 3 data items to enable persistent local Cloud node user data storage:

- TPM Ownership password
  For creation of the AIK keypair in the Cloud join protocol (see Section 3.4) the TPM ownership password must be available. Consequently, for a new software image version to be booted at a Cloud node the TPM owner must enter the TPM ownership password at least once. It is a sensible policy that a platform owner consents to what software is to be run on his machine. The ownership password can then be hashed into the TCG-compliant format and TPM-sealed to the specific software version booted, so further reboots and Cloud joins can be automated.
- Storage – bulk encryption AES key
  Upon first successful completion of the Cloud join protocol the local encrypted storage should be created and initialised, and the encryption key sealed to the current state. If the identical Cloud node software image is rebooted at the identical platform, in step 3 of the join process (see Section 3.4) the node is able to reopen the local storage, if available. This can yield a proof of previous state of work performed to Cloud control and significantly speed up reintegration into the Cloud.
- Storage – image file or partition
  The encrypted storage itself.

We observe that the primary data files of a Cloud node consist of a read-only, attestable Cloud software image which is booted and measured with Trusted Computing, and 3 pieces of data which together enable to keep persistent data over automated node reboots. We suggest to integrate them into one image (file or partition): at the beginning of the storage area a few bytes are reserved for the sealed ownership password hash and the sealed bulk encryption key, which enable the

Cloud join process and access to the rest of the persistent mass-storage area.

3.4 Core Operations

Based on the described node architecture we can now shift our focus to the creation of a Cloud of nodes, under the supervision of a Cloud service provider. Our approach focuses on the security of the Cloud formation process of distributed remote nodes joining the Cloud network via central Cloud control. In the following we give a description of the basic operations in our architecture.

Our join protocol uses an extension of a modified AIK certification exchange. For space reasons we cannot explain every wrinkle of the TCG-specified AIK exchange and refer to (Pirker et al., 2009) for an extensive presentation. The advantage of including this mechanism into the Cloud joining phase is that Cloud control stays very well in control of which hardware platforms are added to the Cloud.

3.4.1 Cloud Node Joining to Cloud Control

Assume a new Cloud node wants to join the Cloud. The Cloud software image is booted on the node. The image which was booted is recorded into the TPM PCRs. Further, the image contains the unique asymmetric RSA public key CC_{pub} and the network address of Cloud control. Immediately after boot the node automatically wants to join the Cloud, a network connection is established.

Node		Cloud Control
(1) EK_{cert},AIK_{pub}, *label*; encrypted with CC_{pub}	$\rightarrow$	decrypt request with CC_{priv}
		verify EK_{cert}
(2) TPM_ActivateIdentity	$\leftarrow$	nonce+etc., encrypted with EK_{pub}
run TPM_Quote		
(3) quote result, platform info	$\rightarrow$	verify quoted state
(4)	$\leftarrow$	"ok"/"denied"

Figure 1: Join Protocol (simplified transport en-/ decryption notation for clarity).

The *join protocol* is depicted in Figure 1. 4 messages are required to be exchanged between the joining node and central Cloud control:

1. The first objective is to establish a secure connection to Cloud control. The first data blob sent to CC is symmetrically encrypted with a fresh symmetric key K, while K itself is encrypted asymmetrically with CC_{pub}. The data blob contains the EK certificate (EK_{cert}) of the TPM, along with AIK_{pub} of a newly created AIK keypair in the TPM. The standard PCA *label* field is used to give a helpful indication what platform the node is running.

   Cloud control receives the blob, decrypts the payload with its secret CC_{priv}. The TCG validation process of determining a valid hardware TPM, as represented by included EK_{cert}, is run.

2. On successful validation, Cloud control generates a random fresh nonce, and other supplemental data for querying of the remote node platform state, depending on the reported platform included in *label*. The return blob is again symmetric/asymmetrically encrypted similar to step 1, with the asymmetric encryption key being used the EK_{pub} contained in the certificate presented by the node.

   The node uses the standard TPM_ActiveIdentity command for decrypting the package. With the received nonce the current system state is quoted using the AIK previously generated. The symmetric key created by Cloud control is used for encrypting all further message exchanges with Cloud control.

3. The signed system state obtained from the TPM, along with additional node or platform specifications (e.g. available storage or processing power) is sent back.

   Cloud control is now able to check the included TPM signed platform state. The quote must be signed with the AIK presented in step 1, the nonce must be equal to the fresh random one sent back in step 2. The reported platform state of the node must be well-known to Cloud control and a trusted Cloud node software image must have been booted.

4. Cloud control welcomes or denies the node in the Cloud.

The first two steps follow the standard TCG design for AIK certification and use Trusted Computing primitives to establish a secure point to point connection from Cloud control to the hardware TPM without man-in-the-middle. For compliance with the TCG standards, we choose 2048-RSA as asymmetric, 128-bit AES as symmetric and SHA-1 as hash cryptographic primitives.

The node has to have faith that he possesses the correct Cloud control public key upon first connection[2] Cloud control verifies that the remote platform hosts a real hardware TPM with the non-migratable EK and a non-migratable AIK was created in it.

The next two steps convince through the Trusted Computing quote operation that an up-to-date PCR state is signed by the AIK. Thus, to Cloud control that the node is in an eligible state and can subsequently join the Cloud.

After this initial join protocol the desired Cloud software platform is run and jobs are assigned to this processing node.

3.4.2 Node Update / Cloud Rejoin

The basic case of a new node joining the Cloud network presented in the previous section is sufficient to construct a Cloud network. From a practical point of view there are two more basic operations to consider.

In the case of an *update* of the node software image Cloud control may refuse the old version in step 4 of the join protocol. Consequently, the Cloud node operator must obtain an updated software image and then try again.

If a Cloud node is rebooted and wants to *rejoin* the Cloud with the identical Cloud node software image, this impacts the question of persistent node data. For instance, it may be very inefficient to resychronize gigabytes of working data with the Cloud over the Internet.

In order for the Trusted Computing attestation process to produce the same measurement result every time, one can only measure read-only data in always strictly the same input order. Naturally, any data processed and code executed may affect a platforms's security state. Consequently, only after successful execution of the Cloud join protocol any user related code and data may be processed or executed. The security of data storage containing user data persisting over reboots must be ensured.

4 TRUSTED PLATFORM SECURITY LIMITATIONS

On the PC platform the current TPM v1.2 has been the Trusted Computing basic building block for almost a decade. Its design does not accomodate for all of the demands of modern platforms today, e.g. with virtualization scenarios (Berger et al., 2006). Also, the current generation of TPM chips and Trusted Execution Technology have been demonstrated to be compromisable (Tarnovsky, 2010), (Wojtczuk and Rutkowska, 2009), (Wojtczuk et al., 2009). However, we expect many of these attacks to be prevented by updated TPM revisions which have been announced for the near future.

The singular proof that a trusted platform hosts a real hardware TPM is embodied by the EK certificate for the TPM. To our knowledge to date just one TPM chip supplier, namely Infineon, includes an EK certificate with their chip.

With physical access one can also compromise the chain-of-trust reported by a state-of-the-art Trusted Execution technology implementation, as demonstrated by (Winter and Dietrich, 2011). Under the assumption that the physical platform is safe, this leaves software attacks as the primary attack surface. Runtime operating system bugs may always exist.

5 RELATED WORK

The Trusted Computing TPM is now becoming mainstream enough for systems integration research and actual prototypes. For the Cloud the TPM promises the possibility to strongly identify a single platform in the Cloud, to measure and report the exact software configuration and to protect the integrity of data and code stored in the Cloud. Especially for IaaS, this has been studied in a number of projects and publications.

In previous work, (Podesser and Toegl, 2011) studied the seamless integration of remote attestation in a SaaS Cloud for Java applications.

(Santos et al., 2009) propose a basic security architecture involving trusted virtualization and present a few security protocols. No practical implementation was reported.

(Krautheim et al., 2010) propose the Trusted Virtual Environment Module, a software appliance that serves as virtual security module for IaaS Cloud applications on virtualization platforms. As a cryptographic module the proposal shows a potential way to allow platform owner and Cloud user to share responsibility and control over data in the Cloud.

(Brown and Chase, 2011) propose to use Remote Attestation so that users can gain insights and trust into SaaS service applications by leveraging trust in a neutral third party. They assume the Cloud platform and provider to be trustworthy, without actually relying on hardware security mechanisms.

The IBM Trusted Virtual Data center (TVDc) (Berger et al., 2008) is designed to offer security guarantees in hosted data centers. It provides containment and trust guarantees based on virtualization. Isolation and TPM-based integrity are managed. It builds upon

[2] Actually, if the CC key in the node image was manipulated CC will detect the potential man-in-the-middle attack in step 3 later.

a Hypervisor derived from Xen and performs TPM-based measurements of software.

The UK myTrustedCloud (Wallom et al., 2011) project studies the integration of an IaaS Cloud platform with KVM-based virtualization and hypervisor trust mechanisms built upon IBM IMA. Different levels of attestation are provided for the different layers in the software architecture.

6 CONCLUSIONS AND OUTLOOK

Our work shows how to join Cloud nodes that are in a specific, trusted state, into a Cloud computing network.

In our approach we assumed *distributed* Cloud nodes, which raises the challenge to physically manipulate them. Based on Trusted Computing technologies we presented a protocol which ensures that nodes joining the Cloud can only do this if they can *attest* that they are in a good state. Our approach is *lightweight* as it does not come with Trusted Computing complexities, a simple TXT enabled boot process is sufficient.

In future work we will examine the modifications required to the respective platform operating system to provide a stable measurement chain and explore which platforms are suited for our architecture.

ACKNOWLEDGEMENTS

We thank the anonymous reviewers for their feedback. This work has been supported by the European Commission through project FP7-ICT-SEPIA, grant agreement number 257433, and project FP7-SEC-SECRICOM, grant agreement number 218123.

REFERENCES

Berger, S., Cáceres, R., Goldman, K. A., Perez, R., Sailer, R., and van Doorn, L. (2006). vTPM: virtualizing the trusted platform module. In *USENIX-SS'06: Proceedings of the 15th conference on USENIX Security Symposium*, pages 305–320.

Berger, S., Cáceres, R., Pendarakis, D., Sailer, R., Valdez, E., Perez, R., Schildhauer, W., and Srinivasan, D. (2008). Tvdc: managing security in the trusted virtual datacenter. *SIGOPS Oper. Syst. Rev.*, 42:40–47.

Brickell, E., Camenisch, J., and Chen, L. (2004). Direct anonymous attestation. In *Proceedings of the 11th ACM conference on Computer and communications security*, pages 132–145, Washington DC, USA. ACM.

Brown, A. and Chase, J. S. (2011). Trusted platform-as-a-service: a foundation for trustworthy cloud-hosted applications. In *Proceedings of the 3rd ACM workshop on Cloud computing security workshop*, CCSW '11, pages 15–20, New York, NY, USA. ACM.

Coker, G., Guttman, J., Loscocco, P., Sheehy, J., and Sniffen, B. (2008). Attestation: Evidence and trust. *Information and Communications Security*, pages 1–18.

Grawrock, D. (2009). *Dynamics of a Trusted Platform: A Building Block Approach*. Richard Bowles, Intel Press, Intel Corporation, 2111 NE 25th Avenue, JF3-330, Hillsboro, OR 97124-5961.

Krautheim, F. J., Phatak, D. S., and Sherman, A. T. (2010). Introducing the trusted virtual environment module: a new mechanism for rooting trust in cloud computing. In *Proceedings of the 3rd international conference on Trust and trustworthy computing*, TRUST'10, pages 211–227, Berlin, Heidelberg. Springer-Verlag.

Pirker, M., Toegl, R., Hein, D., and Danner, P. (2009). A PrivacyCA for Anonymity and Trust. In Chen, L., Mitchell, C. J., and Andrew, M., editors, *Trust '09: Proceedings of the 2nd International Conference on Trusted Computing*, volume 5471 of *LNCS*. Springer Berlin / Heidelberg.

Podesser, S. and Toegl, R. (2011). A software architecture for introducing trust in java-based clouds. In Park, J. J., Lopez, J., Yeo, S.-S., Shon, T., and Taniar, D., editors, *Secure and Trust Computing, Data Management and Applications*, volume 186 of *Communications in Computer and Information Science*, pages 45–53. Springer Berlin Heidelberg.

Sadeghi, A.-R. and Stüble, C. (2004). Property-based attestation for computing platforms: caring about properties, not mechanisms. In *NSPW*, pages 67–77.

Santos, N., Gummadi, K. P., and Rodrigues, R. (2009). Towards trusted cloud computing. In *Proceedings of the 2009 conference on Hot topics in cloud computing*, HotCloud'09, Berkeley, CA, USA. USENIX Association.

Tarnovsky, C. (2010). Hacking the Smartcard Chip. In *Blackhat DC*.

Trusted Computing Group (2007a). TCG infrastructure specifications.

Trusted Computing Group (2007b). TCG TPM Specification Version 1.2 Revision 103.

Wallom, D., Turilli, M., Taylor, G., Hargreaves, N., Martin, A., Raun, A., and McMoran, A. (2011). mytrustedcloud: Trusted cloud infrastructure for security-critical computation and data managment. In *Proeedings of Cloudcom 2011*. in print.

Winter, J. and Dietrich, K. (2011). A Hijacker's Guide to the LPC Bus. In *EuroPKI 2011 proceedings*. in print.

Wojtczuk, R. and Rutkowska, J. (2009). Attacking Intel Trusted Execution Technology. Technical report, Invisible Things Lab.

Wojtczuk, R., Rutkowska, J., and Tereshkin, A. (2009). Another Way to Circumvent Intel Trusted Execution Technology. Technical report, Invisible Things Lab.