Privacy Concerns of TPM 2.0 *

Ijlal Loutfi, Audun Jøsang
University of Oslo, Norway
ijlall@ifi.uio.no, josang@ifi.uio.no

Abstract:
The goal of trusted computing is to provide solutions that allow users to bootstrap trust into their machines based on hardware. The flagship technology for trusted computing is the Trusted Platform Module (TPM) which is specified by the Trusted Computing Group (TCG). TPM hardware chips are currently embedded in 1 billion devices. One of the services that sets TPMs apart from other hardware-enabled security technologies is integrity attestation. However, integrity attestation has been criticized for allowing third party entities that use remote attestation to breach end user's privacy. The recent TPM 2.0 specification has as one of its goals to alleviate this issue. In this paper, we showcase how the new definition of privacy and its corresponding solutions have weaknesses. Solutions which are aimed at protecting end users from third-party privacy attacks have the paradoxical side-effect of exposing end users to potential tracking by manufacturers and other law enforcement entities. We propose two solutions to mitigate this issue.

Keywords: Trusted Computing, TPM 2.0, Remote Attestation, Endorsement Key, Tracking, Privacy.

1. Introduction

In the 1990s, it became increasingly obvious to people in the computer industry that the Internet was going to revolutionize the way personal computers were used, and that commerce was going to move toward this environment. This immediately led to a realization that there was a need for increased security in personal computers (Challener, 2015). This realization has been strengthened by the ubiquitous use of computing devices, as well as by the sensitivity of the business processes that are moving online, such as health, education, and research. However, this goal was challenged by the early hardware and security design principles, which did not take into account security requirements. Hence, a new hardware-based standardized security solution became imperative. The goal of such a solution would be to become an anchor on which new architectures can be built (Challener, 2015).

In 2003, a proposed security solution came in the form of a chip called the Trusted Computing Platform, TPM. Its specifications were released and are maintained by the Trusted Computing Group (TCG). Now, the TPM is present in almost all of computing devices worldwide, and its most recent specification, TPM 2.0, was released in 2014 (TCG-Architecture, 2014). TPM services include all of the security mechanisms traditionally available in other hardware-enabled security solutions, such as providing secure storage for cryptographic material and acting as a crypto processor. In addition, the TPM differentiates itself by its remote integrity attestation capability. Remote attestation allows a platform to cryptographically attest its state to a third party. The latter can then decide on whether it wants to pursue a transaction with the TPM-equipped platform, based on its evaluation of the provided attestation state (TCG-Part1, 2014).

While remote attestation offers, theoretically, significant security advantages, its implementation and practical application has not taken off as expected (Lyle, 2009). The main criticism against remote attestation has focused on 2 aspects: the impracticality of the TPM infrastructure management, and the breach of end users' privacy. The TPM 2.0 specifications released in April 2015 came to alleviate the management and privacy concerns. While we believe that the new specification has succussed in mitigating the management issues, it has increased the risk of privacy breaches. This is due to the way they approach and define privacy. Indeed, the used threat model does not include TPM manufacturers and law enforcement entities as potential threats to end user privacy, which leaves the door open to tracking applications to be built on top of the TPM infrastructure. In a post-Snowden world, assuming the existence of such applications is not a far-fetched thought. The focus of this paper will be on analysing the TPM specifications, and showcasing how tracking can be achieved in such an environment.

We start this paper with a description of the trusted computing paradigm and its flagship technology, TPM. In Section 3 we take a deep dive into the newly released TPM 2.0 specifications. Section 4 presents an overview

* 15th European Conference on Cyber Warfare and Security (ECCWS-2016), Munich, July 2016

of the available literature that discusses the privacy issues of TPM. In Section 5, we discuss how the privacy definition introduced in TPM 2.0 is incomplete, and how it can enable tracking of end users. In Section 6, we propose mitigation solutions. We close with a discussion section.

2. Trusted Computing and the Trusted Platform Module

The concept of the Trusted Computing was developed by an industry consortium, referred to as the Trusted Computing Group. It aims at protecting computing infrastructure and billions of end points, based on a hardware root of trust (TCG-Architecture, 2014). The principal mechanism for achieving this goal is to verify and enforce known, and thereby trusted, configurations of computing platforms. The verification of platform configuration rests on establishing a complete chain of trust, i.e. a verified list of all hardware and software that has been installed on a platform (Lyle, 2009)(Llopart, 2013). This chain of software can then be compared to a list of known 'trusted' software modules, in contrast to traditional approaches such as virus scanners that try to recognize and eliminate instances of bad software in isolation from the rest of software modules on a computing platform (Dietrich, 2012).

The TPM (Trusted Platform Module) is of one of the main building blocks of this paradigm. TPM is defined by the TCG as a computer chip micro-controller that is attached to the motherboard. The technologies proposed by the TCG are centred on the TPM. In a basic server implementation, the TPM is a chip connected to the CPU (Lyle, 2009). The main services offered by TPMs are: Secure Storage; Integrity Measurement; and Remote Integrity Attestation. The latter is the focus of the paper, and its details are explained in Section 3 (TCG-P2. 2014).

3. Integrity Reporting and Attestation

Remote integrity attestation works on the principle that a platform can be trusted if all the software and hardware it has run (its `configuration') can be identified and verified by a relying party (Lyle, 2009)(Martin, 2008). Section 3.1 explains how the TPM enables the collection of measurements about the designated software and hardware, and Section 3.2 focuses on how these measurements can be communicated to a third party through the process of remote integrity attestation.

3.1 Integrity Measurement

The TPM can securely store artefacts (encryption keys, passwords, certificates). It can also store platform integrity measurements that are used to verify the platform integrity. In the TPM jargon, this is expressed by saying that it helps to ensure that the platform remains trustworthy. This is possible thanks to the 16 PCRs (Platform Configuration Registers) it provides. A PCR is a 256/512 bit wide register that can hold a hash digest. Each digest corresponds to a measurement of a piece of software or hardware present on the platform (TCG-P2. 2014). It is not possible to write directly to a PCR. The only allowed PCR operation is **extend(x)**: This operation calculates the new value of a PCR as a hash digest of the concatenation of the old value and the new value **x** (TCG-P3. 2014) which then becomes a hash chain. By definition, the extend operation is non-commutative, meaning that the tracking of the order of events is guaranteed. Also, because the size of the hash digest is fixed, a PCR has the capacity to store an arbitrary number of measurements.

3.2 Remote Integrity Attestation

Attestation is a more advanced use case for PCRs. In a non-TPM platform, remote software can not usually determine a platform's software state. If the state is reported through strictly software means, compromised software can simply lie to the remote party. A TPM attestation can theoretically offer cryptographic proof of software configuration state. This state is communicated through a set of digitally signed PCR measurements. Indeed, the attestation is a TPM quote: a number of PCRs are hashed, and that hash is signed by a TPM key. If the remote party can validate that the signing key came from an authentic TPM, it can be assured that the PCR digest report is authentic and has not been altered (Challener, 2015).

Figure 1 illustrates the principle of remote attestation based on the TPM. The entities in the left-hand side reside in the client platform, and the challenger on the right-hand side represents the remote party.

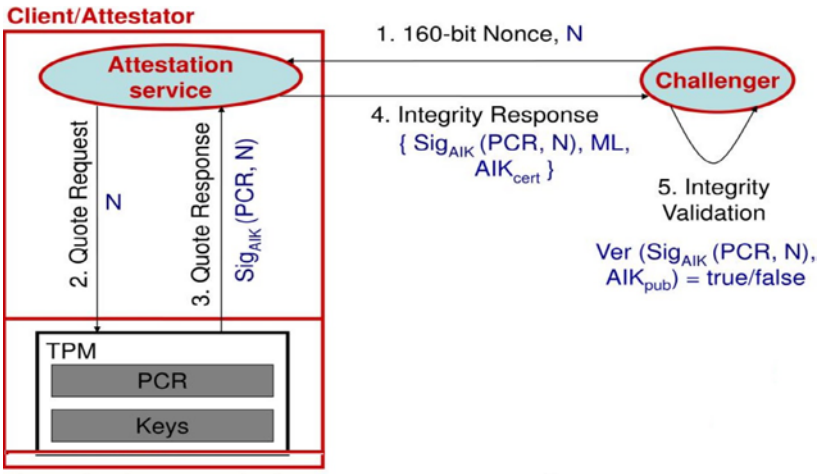

Figure 1: Remote Attestation Message Exchange (Lavina, 2010)

3.2 Issues around Remote integrity Attestation

In this section, we present a concise summary of the TPM privacy issues that our survey of the literature has revealed.

First of all, enabling integrity measurement and reporting would allow a remote party to have information about what software is being run on the user's platform. This can be problematic in an open ecosystem where a remote party can decide to take different actions based on what software is run by the platform it is interacting with, possibly based on unfair discriminatory criteria (Sadegui, 2004) (Kühn, 2007).

Furthermore, if a remote party has malicious intentions, it can identify the specific software modules running on the platform, and target its attack based on the known vulnerabilities of those software modules**.**

We concluded that most of the privacy issues discussed boil down to two main questions:

- How can TPM provide a means to prove that a key was created and was protected by a TPM without the recipient of that proof knowing which TPM was the creator and protector of the key? (Challener, 2015)
- How can we ensure different recipients of remote attestation are not going to link the identities of the users, and augment their knowledge about him and his platform, to more than they were supposed to have?

TCG has tried to address these concerns first in TPM 1.2 and then in TPM 2.0. The first draft specification of TPM 1.2 proposed the use of separate privacy CA. However, privacy groups complained about the difficulty of implementing and operating privacy CAs (Challener, 2015). In TPM 1.2, this led to the inclusion of new commands for a second method of anonymizing keys to help address this concern — direct anonymous attestation (DAA) — which is based on group signatures and provides a relatively complicated method for proving that a key was created by a TPM without providing information as to which specific TPM created it. The advantage of this protocol is that it lets the AIK (Attestation Identity Key) creator choose a variable amount of knowledge they want the privacy CA to have, ranging from perfect anonymity (when a certificate is created, the privacy CA is given proof that an AIK belongs to a TPM, but not which one in particular), to perfect knowledge (the privacy CA knows which EK (Endorsement Key) is associated with an AIK when it provides a pseudonymous certificate for the AIK). The difference between the two methods is apparent when TPM hardware is broken into and inspected, whereby a particular EK's private key is leaked, and potentially published in the Internet. At this point, a privacy CA can revoke the certificate if it knows that it created the privacy certificate associated with that particular EK, but cannot revoke the certificate if it does not know the association between certificate and EK. Once more, DAA failed to be adopted in the field (Challener, 2015).

In TPM 2.0, the TCG responded to the above mentioned privacy issues and the criticism against the security solutions of TPM 1.2, by creating 3 separate key hierarchies. This seems to have tamed down those voices of

criticism. We believe that this architecture will alleviate the management burden. However, it will exacerbate the privacy concerns.

The privacy concerns of TPM 2.0 are due to ***the way privacy is defined by TCG***. In the specifications of trust requirements for TPM 2.0, TCG excludes the manufacturers of TPM chips and computing platforms from the set of potential privacy threats. This assumption is unrealistic for corporate and private users of computing platforms, especially in the post-Snowdon world we live in, where we know that (secret) state sponsored tracking and mass surveillance is a reality. The next section extends on the justification for our claim.

4. Incomplete Privacy Trust Model

The previous sections presented the privacy issues raised by the community, as well as the TCG's response in terms of the TPM 2.0 specifications. We believe the trust model upon which privacy is defined for TPM 2.0 is incomplete. We will examine this claim by examining how TCG defines privacy.

> *The inability of remote parties receiving TPM digital signatures to correlate them —to cryptographically prove that they came from the same TPM. A user can use different signing keys for different applications to make correlation difficult. The attacker's task is to trace these multiple keys back to a single user* (Challener, 2015)

This definition models remote transaction parties as the sole potential threat to end users' privacy. It remains silent about potential threats from TPM manufacturers and law enforcement entities. Hence, by omission, the latter players are assumed to be entities that the end user assumes will not try to breach his/her privacy. That this is an unrealistic assumption is intuitively clear when considering the Snowden revelations (Greenwald *et al.*, 2013) (Menn, 2013). That is, a law enforcement entity can and does subpoena different commercial manufacturers, and legally coerces them to hand in private end users' data.

As an analogy, consider the manufacturer of self-encrypting drives (SEDs), i.e. data storage devices that automatically encrypt and decrypt data that is being written to, and read from the device, under a key defined by the owner. Assume now that the same threat model as for TPM 2.0 is being used, which would imply that the SED manufacturer can read data on all SEDs it has produces and sold, because it has a way to recover encryption keys defined by owners. This type of SEDs would certainly not succeed in the market if the key escrow capability were known. No users in their right mind would buy such SEDs.

For TPM 2.0, the market situation is different, because users do not actually buy TPM chips. Instead, TPM chips come bundled with the computer platform that users buy, typically without their knowledge.

The TCG aims at making the TPM a cornerstone of the TCB (Trusted Computing Base) of computing platforms, where privacy is assumed to be one of its main functions. From that perspective it is reasonable to expect that the TPM genuinely supports the privacy of end users in a transparent fashion. It is therefore surprising to read in the TPM 2.0 specifications that end-user privacy has been partially traded off to give TPM manufacturers the power to identify and trace end-user computing platforms.

4.1 TPM Privacy Compromising Design Decisions

Because of this incomplete definition of the privacy trust model, the TPM 2.0 specifications define insufficient and inadequate mechanisms to protect end users from law enforcement entities and the TPM manufacturers breaching their privacy. Two of these problematic design decisions are presented in the following sections:

4.1.1 New Hierarchies: The Always-On Platform Hierarchy

A hierarchy is a collection of entities that are related and managed as a group. Those entities include permanent objects (the hierarchy handles), primary objects at the root of a tree, and other objects such as keys in the tree.

TPM 1.2 had only one hierarchy, represented by the owner authorization and storage root key (SRK). There can be only one SRK, always a storage key, which is the lone parent at the top of this single hierarchy. The SRK is generated randomly and can not be reproduced once it has been erased. It can not be swapped out of the TPM. Child keys can only be created and wrapped with (encrypted by) the SRK, and these child keys may in

turn be storage keys with children of their own. However, the key hierarchy is under the control of the one owner authorization; so, ultimately, TPM 1.2 has only one administrator.

The TPM 1.2 hierarchy architecture led to considerable challenges to how TPM platforms would be managed. This is mainly because of the overlapping authorization domains for the TPM firmware, TPM owner (e.g.: IT administrator who own the platform in which the TPM is embedded), and TPM end user. In order to overcome the management challenges, TPM 2.0 introduced a new architecture, which we believe has increased the risk of tracking TPM end users (Challener, 2015). TPM 2.0 defines four different key hierarchies:

- ***Standard storage hierarchy:*** Replicates the TPM 1.2 family SRK for the most part
- **Platform hierarchy:** Used by the BIOS and System Management Mode (SMM), *not* by the end user
- **Endorsement hierarchy or privacy hierarchy**: Prevents someone from using the TPM for attestation without the approval of the device's owner
- ***Null hierarchy:*** Uses the TPM as a cryptographic coprocessor

The platform hierarchy, which is the focus of this section, is intended to be under the control of the platform manufacturer, represented by the early boot code shipped with the platform. The platform hierarchy is new for TPM 2.0. In TPM 1.2, the platform firmware could not be assured that the TPM was enabled. Thus, platform firmware developers could not include tasks that relied on the TPM (TCG-Architecture, 2014).

As the most privileged hierarchy, the Platform Hierarchy *is enabled* at reboot. The intent is that the platform firmware will generate a strong platform authorization value (and optionally install its policy). Unlike the other hierarchies, which may have a human enter an authorization value, the platform authorization is entered by the platform firmware. Therefore, there is no reason to have the authorization persist (and to find a secure place to store it) rather than regenerate it each time (Challener, 2015).

The problematic part of the platform hierarchy in our opinion is that:

- It has its own enable flag
- The platform firmware decides when to enable or disable the hierarchy. While the TCG intent of this design decision is for the TPM to always be enabled and available for use by the platform firmware and the operating system, this decision clearly takes consent away from end users.
- The platform hierarchy also does not need to use the same cryptographic material that is used in the rest of the TPM, making it ever more obscure.

4.1.2 Re-Certification:

Every hierarchy has a list of keys and object that belong to its control domain. The keys and data under any given hierarchy are encrypted using the primary key of that same hierarchy. In TPM 1.2, the Endorsement Key sitting at the top of the TPM hierarchy can never be erased once it has been embedded in the TPM chip by its manufacturer. This was a clear breach of end user's privacy, as all subsequent keys generated by the TPM could be traced back to a single EK, and hence, identifies the platform. The proposed solutions (DAA and privacy CA) are not real solutions for this privacy issue, as they would still allow the TPM manufacturers and low enforcement entities to track end users. Furthermore, these solutions were never really implemented in TPM 1.2, leaving end users exposed also to third party attestators privacy attacks.

In TPM 2.0, each one of the 4 new hierarchies could have a different seed from which all subsequent hierarchy keys are generate. Furthermore, these seeds could be deleted even after the TPM has been shipped to end users. At first, this design decision sounds like an appropriate solution to guard against privacy threats from TPM manufacturers and law enforcement agencies. However, it is very surprising that TPM 2.0 has introduced an extra mechanism, called *re-certification*, that always maintains a cryptographically traceable link from the platform Hierarchy controlled by the TPM manufacturer, to the other hierarchies, even when the root keys of the other hierarchies are deleted and re-generated by the user. This mechanism gives TPM manufacturers the power to trace TPM-equipped platforms no matter what. Ironically, in the jargon of the TPM 2.0 specifications, this is described as *"to maintain the chain of trust"*.

Indeed, in the platform hierarchy, the primary seed (EPS) can be replaced by a new value. However,

"the TPM specification allows cross certification of keys between the Platform hierarchy and the Endorsement hierarchy under control of the platform firmware. Cross certification allows a chain of trust to be maintained as the seeds are changed (Challener, 2015)*. "*

Figure 1 illustrates the ownership of key hierarchies before and after re-generation by the end user. Also shown is how the TPM manufacturer can keep track of key hierarchies it does not own.

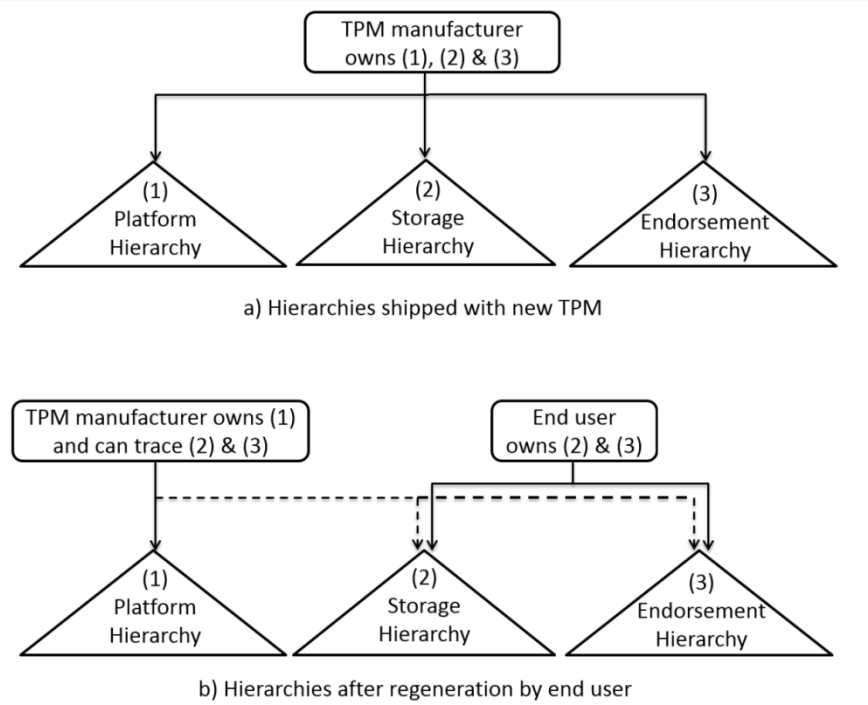

Figure 2: Re-certification of key hierarchies in TPM 2.0

5. Proposed Solutions:

Any reasonable solution to the above discussed privacy issues would of course require a radical change of the design of TPM, as it would question the fundamental definition of privacy that the specifications are built upon. While we do wish this to be the case, we do acknowledge the impracticality, or rather the difficulty of pushing for such a change. Hence, in this paper we are advocating for two main measures in order to mitigate the risks of mass tracking.

5.1 Elimination of Re-Certification:

If TPM is to truly succeed and thrive in an enterprise environment, and/or be adopted in a serious manner for large scale national deployment; and if it is to become an attractive architecture for developers to build their security upon, the TPM should allow for a way for its keys and cryptographic data to be completely outside of the PKI of the TPM manufacturers. While this obviously would require more infrastructure management from the IT organization that would choose such a design, it would mean that they can be assured that their data will be completely under their sovereignty, and that their privacy-sensitive data has no way of being compromised from manufacturers. This requirement is ever more important in today's geopolitical situation, in which state spying on each other is common currency. In this post-Snowden era, one can not assume that a company in a foreign nation state will not hand in private data about end users, should the authorities of that nation state request it. This proposed solution, if adopted, would be in line with the TPM decision to allow each manufacturer to embed cryptographic algorithms that they trust. These two design requirements together are what could make TPM truly a device that not only defends against malware, but also against mass tracking by potential adversaries. TPM would in this case be the end user's true advocate.

5.2. Disclosure

In case of TPM 2.0 where the TPM has a platform hierarchy over which the owner has no control, and which can be enabled whether the owner of the platform desires it or not, we believe that the updates and any other possible cryptographic communication that goes into and out of it should be made visible to end users, whenever they are interested. If they are not able to disable it completely, it would be more ethical for them to be aware of what is happening in their own platform. Hence, we suggest that a monitoring application should be provided to end users in order to notify them about passive actions that do not require their active participation, and yet are happening. This could allow consent from end users in order to perform TPM platform updates for instance. Such an application is technically possible, so it is really a matter of willingness whether we would see it happening or not.

6. Conclusions

In this paper, we have discussed the shortcomings of the privacy definition provided by the Trusted Computing Group, and around which TPM 2.0 specifications have been designed. We conclude that TPM 2.0 does indeed strengthen the protection of end users from third party applications that use the remote attestation service. However, it does not protect end users from tracing and tracking by platform and TPM manufacturers, nor from law enforcement authorities that can subpoena them to hand in user sensitive data. In this post-Snowden era, it is not far-fetched at all to talk about mass surveillance at a state scale. This problem becomes ever more serious when considered in the context of states interfering with the sensitive personal data of foreign end users, just because the platform and TPM manufactures happen to be in their legal jurisdiction. This is no light matter to play around with, given the type and amount of sensitive information remote attestation by the TPM can reveal about its platform owner. We have proposed two solutions aimed at mitigating the risk of tracing and tracking. We believe that these issues should be highlighted, not only for ethical reasons, but also because it will give the TPM a real chance in being adopted in the enterprise market. If so, it would be a powerful tool to add to our security defence arsenal in a cyber-landscape of increasing threats.

References

Dietrich, Kurt *et al*. (2007). A Practical Approach for Establishing Trust Relationships between Remote Platforms using Trusted Computing. In *Trustworthy Global Computing*, LNCS 4912, pp 156-168. Springer 2007.

Greenwald, Glenn and MacAskill, Ewen (2013). *NSA Prism program taps in to user data of Apple, Google and others*, The Guardian, 7 June 2013, http://www.theguardian.com/world/2013/jun/06/us-tech-giants-nsa-data

Jiewen, Yao and Zimmer, Vincent (2014). *A Tour Beyond BIOS with the UEFI TPM2 Support in EDKII*. Intel White paper, 2014. https://firmware.intel.com/sites/default/files/resources/A_Tour_Beyond_BIOS_Implementing_TPM2_Support_in_EDKII.pdf

Kühn, Ulrich and Selhorst, Marcel and Stüble, Christian (2007). Realizing property based attestation and sealing with commonly available hard- and software. Proceedings of the 2007 ACM workshop on Scalable trusted computing (STC'07), pp.50-57. ACM New York, 2007.

Lavina, Jain and Vyas, Jayesh (2010). *Security Analysis of Remote Attestation.* CS259 Project Report. Stanford University, 2010.

Lyle, John and Martin, Andrew (2010). Trusted computing and provenance: Better together. In Proceedings of the *2nd Conference on Theory and Practice of Provenance (TAPP'10)*, pages 1. 2010.

Lyle, John and Martin, Andrew (2009). On the Feasibility of Remote Attestation for Web Services. *International Conference on Computational Science and Engineering (CSE '09)*. IEEE 2009.

Martin, Andrew (2008). *The ten page introduction to trusted computing*. Technical report CS-RR-08-11, University of Oxford, 2008.

Menn, Joseph (2013). *Exclusive: Secret contract tied NSA and security industry pioneer*, Reuters, 20 December 2013, http://www.reuters.com/article/2013/12/20/us-usa-security-rsa-idUSBRE9BJ1C220131220

Pelegri-Llopart, Eduardo and Yoshida, Yutaka and Moussine-Pouchkine , Alexis (2007). *The GlassFish Community Delivering a Java EE Application Server*, Sun Microsystems, 2007. https://glassfish.dev.java.net/faq/v2/GlassFishOverview.pdf.

Schneier, Bruce (2015). *Everyone wants to have security, but not from them*. https://www.schneier.com/blog/archives/2015/02/everyone_wants_.html.

Sadeghi, Ahmed-Reza and Stuble, Christian (2004). Property-based attestation for computing platforms: caring about properties, not mechanisms. *Proceedings of the 2004 Workshop on New Security Paradigms (NSPW '04)*. 2004.

TCG. TPM Main Part 1, Design Principles (2014). Technical report, Trusted Computing Group. 2014.

TCG. TPM Main Part 2, TPM Structures (2014). Technical report, Trusted Computing Group. 2014.

TCG. Trusted Platform Module Summary. Technical report, Trusted Computing Group. 2014.

TCG. TPM specification: Architecture overview. Technical report, Trusted Computing Group. 2014.

TCG. Attestation Identity Key (AIK) Certificate Enrolment Specification: Frequently asked questions. 2011.

TCG. Endorsement Key (EK) and Platform Certificate Enrolment Specification. 2013.

Will, Arthur and Challener, David and Goldman, Kenneth (1015). *A Practical Guide to TPM 2.0*. Apress Open, Springer New York, 2015