Improving End-user Security and Trustworthiness of TCG-Platforms

Klaus Kursawe,
kursawe@acm.org

Christian Stüble
Saarland University, Germany
stueble@acm.org

September 29, 2003

Abstract

Over the last two years, the computing industry has started various initiatives to increase computer security be means of new hardware. The most notable effort is the Trusted Computing Group (TCG, former TCPA), which unites most important companies to develop a standard for security hardware.

While efforts towards more security are generally appreciated, the new technology has raised fears that the user will lose control over his platform, and that its extensive use would harm competition in several areas of IT technology.

In this paper, technical modifications to the TCPA Specification (based on version 1.1b) are suggested. If implemented, these suggestions would eliminate the potential to abuse the technology against the users. To the best of our knowledge, none of the intended use cases of the TCG would be affected. The proposals will be discussed with TCG Members on the GI annual conference 2003 in Frankfurt and then officially presented to the TCG.

Motivation

The Trusted Computing Group (TCG, former TCPA) claims to provide security and trustworthiness of computers for both vendors and end-users. Nevertheless, the TCPA specification version 1.1b [9, 8] lacks some important properties that may help to improve the trust in the TCPA specification and to improve security of end-user TCPA-platforms. In the following, we distinguish three different roles that may be interested in the integrity metrics of trusted platforms:

- Obviously, the *user* of a computing platform is interested in the integrity metrics to protect its privacy and to prevent misuse of private information, e.g., a signature key.

- The *owner* of a computing platform, e.g., a company, may want to protect the integrity of their computing systems to be able to enforce their policy. For instance, companies often want to prevent that users change/replace the installed software. For personal systems, owners and users can be the same.

- *Providers* of digital goods may be interested in the integrity metrics of computing platforms, since only a few platform configurations guarantee that their digital goods are protected. Additionally, also providers of digital services may be interested in the integrity metrics, since they may adherence for some properties under special circumstances.

In the following sections, we discuss different technical modifications to the TCPA specification version 1.1b.

1 Publish Use Cases and Application Scenarios

The TCG claims to improve security and trustworthiness of IT-systems with the TCPA specification. However, this statement is quite fuzzy, and little details about the intended use cases have been published. Therefore, technical discussions about the current and future TCPA specifications are very difficult to make, since one does not know whether suggested modifications violate the requirements that are currently fulfilled by the specification.

Thus, the suggestion is that the TCG publishes the assumptions, the trust model and the use cases that were and are used by the TCG to develop the TCPA specification.

Making this information public would, for example, stop the unpleasant public discussion whether TCPA enables Digital Rights Management (DRM) or not. Since there exists not the "one and only DRM application", there are many positive and negative consequences which are all more or less related to the general DRM. Many people are sceptic about TCPA, since they may only know the bad ones.

Having a list of use cases, it is on the one hand possible to have public discussions whether these scenarios will be accepted or not. On the other hand, it enables the security community to suggest improvements about the TCPA specification which allows the good use cases but prevents the bad ones.

2 Secure Booting and Platform Authentication

A secure bootstrap architecture is an important security requirement to protect the integrity of the trusted computing base (TCB) of security-critical systems. Unfortunately, no solution based on game-theoretic protocols can solve the boot problem in general, since it remains impossible for users and owners to detect that the platform has been replaced: an attacker can always perform a so-called mafia fraud by relaying authentication information (e.g., the whole user input and output) between the honest and the malicious platform [1].

Obviously, the primary security objective of a secure platform is to authenticate the user to protect integrity and confidentiality of information. Since users enter critical information (e.g., authentication data like passphrases) into the platform, it is also necessary that a secure bootstrap architecture authenticates the platform to the user *before* security-relevant data is entered. Therefore, it is, especially in the context of end-user devices, important that users can recognize the current platform configuration without another trusted device, e.g., a smartcard.

The current TCPA specification comes with two different mechanisms to provide a secure bootstrap architecture:

Attestation Attestation allows external IT-systems (e.g., another host or a security token) to verify the integrity metrics of a TCPA platform and thus fulfills – partly – the platform authentication requirement. But since the designated verification mechanism is based on cryptographic schemes, users without such a trusted device are unable to verify the platform configuration. Because the trusted device has to provide a secure user interface, conventional smartcards are inappropriate. Thus, the secure boot problem is moved to another device.

Sealing By binding security-critical information to a specific platform configuration using sealing, it can be prevented that security policies are bypassed by insecure operating systems. Thus, the primary security objective is fulfilled. The major drawback of this solution is that local users without another trusted device may not be able to notice that the platform configuration has changed.

Therefore, it is necessary to be able to protect the integrity of the first piece of unprotected code that is executed. In the following, we suggest two possible solutions:

- Extend the CRTM and the BIOS in such a way that only trusted configurations are loaded. As an example, the CRTM/BIOS could compare the current configuration with a list of

trusted configurations, selected by the platform owner. If the configuration does not match, the CRTM/BIOS can halt the boot process. This is similar to how the bootstrap architectures proposed by Arbaugh (see, e.g., [3, 4, 6]) work.

- To prevent that the CRTM can actively affect the boot process, an alternative solution would be to "show" the platform owner which configuration will be loaded. For instance, the owner could give trusted configurations (PCR values) names. On startup, the CRTM/BIOS can display the name of the configuration (or "unknown" if the owner has not named it) and wait until the user acknowledges the current configuration by, e.g., a boot passphrase.

As opposed to the functions provided by the current TCPA specification, the proposed solution protects against all kinds of software attacks that modify software components of the platform without requiring the assistance of a secondary trusted computing device.

3 Substitution of the PC registers

While the integrity measurement mechanism based on PCR's allows many meaningful applications, e.g., based on sealing or attestation, this feature has the potential to significantly increase monopolies in the field of operating systems and applications (see, e.g., [5]):

- Attestation allows content and service providers to force users to use a specific software configuration, e.g., a specific operating system. This is already the case today (e.g., many banks provide banking software for only one operating system, a lot of websites support only one web-browser, and a lot of hardware devices, like music players, expect a specific operating system). Nevertheless, at least in principal users of alternative operating systems can use alternative, mostly open-source, software. If TCPA is used to enforce the software configuration, this becomes impossible.

- Sealing can be misused by software vendors in such a way that concurrent products cannot be compatible any more, as they simply cannot access the necessary keys. Applications can force users to use proprietary document formats a concurrent product cannot read. As a consequence, to be able to use existing data, users cannot switch to an alternative software product. If operating systems seal their filesystem, even virtual machines (e.g., VMware) will not be able to mount them any more.

One Solution to prevent these disadvantages by allowing the platform owner to substitute the status of the PC registers with virtual values is to allow users to – under controlled circumstances – overwrite the current platform configuration. To still benefit from the PC registers, this should be done by a substitution rather than overwriting - if the correct (owner defined) PCR value is measured, the TPM substitutes it with another one. As this substitution is potentially dangerous, the conditions under which substitution is allowed have to be carefully controlled. Under no circumstances should it be possible to talk an unexperienced user into enabling this option or to enable it without definite consent of the owner.

Of course, the correct PCR values are important in many scenarios, for instance if a bank adherences for a provided banking software. Therefore, the TPM additionally should add the correct PCR values, encrypted with the key of the TPM vendor. In case of a conflict, e.g., if a user takes the bank to court, the vendor (or the court) can decrypt these values to verify whether the user used the correct banking software or another one.

Since only the platform owner is allowed to decide which configuration should be mapped to another virtual one, the suggested modification has no consequences for business DRM cases, e.g., document management within a company. Since the real PCR values are added, also the attestation function can still be used. But the modification does prevent DRM cases between unknown platforms, e.g., between content providers and end-users.

4 Overwriting the Endorsement Key

The endorsement key (EK) is used by TCPA to uniquely identify a trusted platform module (TPM). According to the current specification, the endorsement key is generated and certified by the vendor of the TPM. On previous discussions many people displayed fear that the creation process or the certification process could intentionally or unintentionally be compromised. As a consequence, several misuses with faked attestation identity keys (AIK) would become possible.

Although a global EK is required to authenticate unknown platforms to each other (e.g., in e-commerce scenarios), there are a lot of scenarios imaginable which only require a platform authentication in a local environment, e.g., a local network. For instance, authentication between platforms of a company does not require a global PKI. Instead, it would suffice to use local certificates, e.g., generated by the platform Owner or the User.

Therefore, a functionality that allows users to generate a new EK that overwrites the old one should be added. This way, possible misuses of a compromised EK generation process or a compromised EK certification process can be limited. Additionally, DRM scenarios are effectively prevented after the creation of a new EK, since providers of digital content cannot securely verify the platform configuration any more.

To allow the owner to integrate the platform into a global PKI later, the TCPA specification should provide an appropriate mechanism. Two possible approaches are:

- Create a chain of certificates that allows external systems (or, at least, the vendor of the TPM), to verify that the new EK was created by a valid TPM. As this implementation would still leave a unique identifier in the TPM, a function could be added to show the chain of certificates to a trusted party which then certifies the new endorsement key. After this certificate is issued, the chain can be deleted.

- Store the original EK within the TPM and use it as default after the execution of the "take ownership" command.

Since functions to generate cryptographic keys are already part of the TCPA specification, the overhead of this modification should be rather small.

5 Access to the Storage Root Key

The current TCPA specification ensures that only the TPM has access to the storage root key (SRK). While this behavior is essential for many scenarios, e.g., to protect the user of a system from the owner in a business environment, it also has some important disadvantages that are still under public discussion [2, 7]. First, this property allows vendors to develop software on the top of a TCPA platform that acts contrary to the interest of the user or owner of the platform, e.g., a system-wide censorship of documents. Second, it supports monopolies, since it prevents software re-engineering methods (OpenOffice may not be able to read Word-documents based on TCPA). Third, all data that was encrypted using a non-migratable key can only be recovered with cooperation of the platform manufacturer using the maintenance [6].

In our opinion, the TCPA specification could be improved in such a way that platform owners are able to decide whether their platform can be used for criticized scenarios, or not. Ideally, platform owners are able to decide it once when take-ownership is performed. Although this requirement can be fulfilled by the operating system that controls the TPM, TCPA should also provide a hardware mechanism that ensures such a behavior, since users can currently not necessarily trust their operating system and its vendor.

By allowing the platform owner to access (know) the SRK, the enforcement of security policies that violate security policies of owners can by prevented at the hardware level, since owners which know the SRK are able to decrypt sealed data independent of the TPM.

Obviously, a SRK that is known to the platform owner violates a couple of meaningful DRM- and authentication use cases. A solution is to allow owners only to access the SRK optionally:

owners could be enabled to overwrite[1] the SRK using a secure function, e.g., a BIOS function that can only be invoked by a local authenticated owner. To allow providers to distinguish between SRKs that are known to their owners and those which are not, the CRTM should consider the fact that the SRK is known to the user when it does the integrity check, e.g., by involving this information into the derived hash value.

6 Migrating the SRK

If platform owners want to change to another TCPA platform, e.g., to a newer one, all data including the Storage Root Key (SRK) has to be moved to the new platform. While this is no problem with migratable keys, non-migratable keys, e.g., the SRK, can only be moved by using the (optional) maintenance function defined in the TCPA specification v1.1b. The disadvantage of this function, which can also be used to create a backup of the SRK, is that it requires to involve the vendor of the TPM to migrate the SRK to another platform. While this procedure is copious and expensive, it is questionable what happens if platform owners want to migrate an SRK of a TPM of vendor A to a TPM of vendor B. In the current TCPA implementations by IBM and HP, this option has been left out completely – there is no way to migrate the SRK at all.

To be able to securely migrate the SRK without involving the TPM vendor, the suggestion is to improve maintenance TPM mode: when the platform owners invoke the migration mode, they have to define the public part of the EK of the target TPM and a certificate indicating that the given key is a valid TPM EK. The source TPM then encrypts the SRK by this key and outputs it. After outputting the encrypted key, the platform owner can only repeat the output (to prevent loss of data if the encrypted key gets loss) or continue the migration by deleting the local SRK. The target TPM decrypts the encrypted EK, and uses it as its own SRK.

To prevent loss of data because of hardware problems of the target TPM, the existing maintenance function can be used as a backup: the source TPM encrypts the SRK not only under the EK of the target TPM, but also under the public key of the vendor of the source TPM.

7 Removal of Cryptographic Keys

Currently, TCPA does not allow to effectively remove sealed data, e.g., cryptographic keys. A secure removal of keys is for instance important whenever digital content has to be migrated to another platform. A possible scenario is a user, which resells digital content it owns. Currently, the user could make a copy of the persistent storage (e.g., the harddisk), sell the content (which removes the key used to encrypt the content) and reset the state of its platform by using the copy of the persistent storage. This scenario is different from the SRK migration problem discussed in Section 6, since the SRK is stored within the TPM and can therefore securely be deleted.

The following solutions to this problem are imaginable, if the TPM is extended by a protected counter:

- The counter is used to stamp a key revocation list.
- The TPM is extended by another PC register that acts as the counter (or the CRTM reads the current counter status and writes it into an existing PCR). If the counter is increased, sealed data that was bound to this register becomes invalid.

8 Take Ownership without Legacy Operating System

The take-ownership operation of the TPM is very security-critical, since it builds the basic and only trust relationship between platform owner and TPM. The current TCPA specification does currently not define which component performs the take-ownership operation. Currently available

[1] Platform owners must not be allowed to access a SRK that was preliminary used for DRM use cases

TCG-Platforms, e.g., those provided by IBM or HP, perform these operations by the operating system or application. Commonly used operating systems are not secure, nor are their vendors trustworthy enough to perform such a highly critical function. Therefore, we suggest to provide a trusted BIOS function (or, e.g., a separated trusted software component) that provides the take-ownership operation.

9 Deactivation of the TPM

Although the TPM can be deactivated according to the current TCPA specification, many people criticize that vendors of software or hardware could try to convince unexperienced users to activate the TPM, e.g., to be able to enforce their policies.

To prevent this scenario, vendors should build TCPA platforms in such a way that software products cannot distinguish between a platform without a a TPM and a platform with a deactivated TPM. We see two possible solutions:

- The TPM is a hardware module that can be unplugged by the platform owner. This approach is already proposed in the TCG PC-Specific Specification, but there are no announcements of a corresponding product. To prevent that adversaries attack the interface between platform and TPM, additional protection, e.g., encryption between TPM and CRTM, is necessary.
- Vendors always provide hardware platform with TCPA support and without TCPA support (and design the TCPA hardware appropriate) making it undecidable to software components to decide whether a TPM is not available or deactivated. To implement this, either the definition of a deactivated TPM has to be changed, or a new mode of deactivation has to be introduced.

References

[1] A. Alkassar and C. Stüble. Towards secure IFF — preventing mafia fraud attacks. In *Proceedings of IEEE Military Conference (MILCOM)*, 2002.

[2] R. J. Anderson. The TCPA/Palladium FAQ. `http://www.cl.cam.ac.uk/~rja14/tcpa-faq.html`, 2002.

[3] W. A. Arbaugh, D. J. Farber, and J. M. Smith. A reliable bootstrap architecture. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pages 65–71, Oakland, CA, May 1997. IEEE Computer Society, Technical Committee on Security and Privacy, IEEE Computer Society Press.

[4] N. Itoi, W. A. Arbaugh, S. J. Pollack, and D. M. Reeves. Personal secure booting. In V. Varadharajan and Y. Mu, editors, *Information Security and Privacy — 6th Australasian Conference, ACISP 2001*, volume 2119 of *Lecture Notes in Computer Science*, pages 130–144, Sydney, Australia, July 2001. Springer-Verlag, Berlin Germany.

[5] C. Koenig. Trusted Computing im Fadenkreuz des EG-Wettbewerbsrechts. Zentrum für Europäische Integrationsforschung (ZEI), `http://www.zei.de/download/Konferenzseite/koenig_20030509.pdf`, 2003.

[6] S. Pearson. *Trusted Computing Platforms — TCPA technology in context.* Hewlett-Packard Company, Prestice Hall PTR, 2003.

[7] B. Schneier. Palladium and the TCPA. `http://www.counterpane.com/crypto-gram-0208.html#1`.

[8] Trusted Computing Platform Alliance (TCPA). TCPA PC specific implementation specification, Sept. 2001. Version 1.00.

[9] Trusted Computing Platform Alliance (TCPA). Main specification, Feb. 2002. Version 1.1b.