Clarifying Misinformation on TCPA

David Safford, IBM Research, October, 2002

Introduction

In a recent papers, Ross Anderson [8], Bill Arbaugh [10], and Lucky Green [12] criticize TCPA [1] and Palladium [3] claiming that they are a disaster for the consumer, serving only to enforce Digital Rights Management (DRM). These papers have incited widespread consumer concern, even leading to an anti-TCPA website [9] calling for letter writing campaigns to TCPA member companies. Unfortunately these papers misrepresent TCPA in that they:

- improperly lump together TCPA, Palladium, and DRM
- present speculation about TCPA as fact
- are full of technical misunderstandings of the TCPA Specification

This report analyzes these anti-TCPA papers, pointing out these errors in detail, and showing that their conclusions about TCPA are simply wrong. This paper defends only TCPA. Palladium and DRM have to defend themselves. Note: I have a system with a TCPA chip running both Windows and Linux, and have verified all of the following comments on actual hardware. All views are my own, not necessarily those of IBM.

What is TCPA?

First of all, the papers improperly lump together TCPA, Palladium, and DRM, considering them as one thing. So let's clarify what we mean by "TCPA". The Trusted Computing Platform Alliance was formed to establish an industry standard for a trusted computing subsystem to be added to PC's. IBM has been shipping a predecessor, called the "Embedded Security Subsystem" (ESS) chip for over two years in NetVista desktop and ThinkPad notebook computers. The ESS chip was basically a public key smartcard chip placed directly on the motherboard's SMB bus. The concept was to make public key hardware tokens available at very low cost, by embedding them, and eliminating the need for separate smart cards and readers. Other companies were looking at similar solutions, and it became clear that there needed to be a single common standard. The TCPA organization has published open, freely downloadable specifications for all of TCPA[1]. The TCPA main specification defines a chip that meets the security requirements of all the member companies. In addition, other TCPA specifications cover PC specific interface and software details.

The TCPA chip itself has three main groups of functions:

- public key functions
- trusted boot functions
- initialization and management functions

The public key functions are very similar to IBM's original ESS chip design, (which already has GPL'ed driver code in active use by several projects.) They provide for on-chip key pair generation, along with public key signature, verification, encryption and decryption. The "trusted" boot functions provide the ability to store in Platform Configuration Registers (PCR), hashes of configuration information throughout the boot sequence. Once booted, data (such as symmetric keys for encrypted files) can be "sealed" under a PCR. The sealed data can only be unsealed if the PCR has the same value as at the time of sealing. Thus, if an attempt is made to boot an alternative system, or a virus has back-doored the operating system, the PCR value will not match, and the unseal will fail, thus protecting the data. The initialization and management functions allow the owner to turn functionality on and off, reset the chip, and take ownership. This group of functions is somewhat complex, to provide strong separation of what can be done at BIOS (boot) time, and what can be done at normal run-time,

so that sensitive operations (like reading the endorsement key) can't be performed by malicious applications trying to compromise one's privacy.

What is Palladium?

Palladium is a Microsoft led project to add "trusted" computing to Windows, through a combination of hardware and software. The hardware consists of a chip similar to the TCPA chip (they call it the SCP), along with processor modifications to add a ring -1 protection level, chipset modifications to help isolate a trusted memory space, and to provide trusted path from the keyboard and trusted display. In software, Palladium uses hypervisor techniques to add a trusted software 'TOR", in a protected space separate from the normal operating system. At this time, it is not clear whether or not Palladium will support the use the TCPA chip in addition to their SCP chip. Microsoft has stated that the Palladium hardware will have an open specification, and that Linux could be written for it. However, Microsoft has a large number of patents on the use of these hardware modifications. So far, Microsoft has not guaranteed even "reasonable and non-discriminatory" licensing of these patents, so Microsoft could easily block Linux from using these features.

The bottom line is that TCPA and Palladium are two ***different*** projects. The TCPA hardware provides only a subset of the full Palladium functionality, which includes significant additional hardware and software elements. Only TCPA already has a freely downloadable detailed specification, and a tested port of all driver and library level software to Linux.

What is DRM?

Digital Rights Management is the attempt to control the viewing and copying of digital content, such as music and movies. Existing DRM systems, such as Microsoft Media Player, with its "Windows Media Audio' (.wma) files, run at the application level. Ross Anderson and others speculate that TCPA and Palladium will be used to place stronger DRM in the "trusted" operating system. My personal opinion (not speaking for IBM) is that DRM is stupid, because it can never be effective[6,7], and it takes away existing rights of the consumer. But this is not the place for that debate. To condemn TCPA for the ability to run a bad application is absurd. This argument is exactly like the arguments of governments in their attempts to ban encryption, under the rationale that encryption can be used by terrorists to hide their messages. In the case of encryption, people realized that encryption is simply a tool for protecting data, and it can be used in good cases or bad. The same is true of the trusted computing offered by TCPA. Trusted computing can make any application more secure - good applications or bad. I have no problem with people arguing against DRM; I agree completely. But to argue that trusted computing is bad because it can support DRM is fallacious - it completely ignores the security TCPA can add to good applications, such as the security of my personal authentication keys, or my personal encrypted files. See the companion paper, which goes into more detail on the good things that can be done with TCPA.

Specific Technical Comments:

Ross Anderson's TCPA FAQ [8]

Incorrectly lumps together TCPA, Palladium, and DRM:

> *Ross says that "Palladium ... will build on the TCPA hardware", and that "The obvious application is digital rights management (DRM)".* Most of the rest of his FAQ comments on the problems with DRM, as possibly done in Palladium. First, this is not an issue with TCPA, but with Palladium. Secondly, the obvious application of TCPA is to enable individuals to secure their private keys, and to secure their encrypted data against viruses or other attacks that compromise the operating system, not DRM. It is wrong to lump TCPA together with Palladium and DRM, and not to distinguish arguments between them. Argue all you want against DRM, but don't blame TCPA for things done in Palladium - they are two different systems.

Is full of technical errors: Some of the more extreme errors include:

> *"When you boot up your PC, Fritz [the TCPA chip] takes charge. He checks that the boot ROM is as expected, executes it, measures the state of the machine; then checks*

the first part of the operating system, loads and executes it, checks the state of the machine; and so on." This is completely false. The TCPA chip doesn't execute anything. It accepts request data, and replies with response data. In the IBM version, TCPA sits on the LPC bus, using I/O mapped registers. The TCPA chip does not and **cannot** control execution!

"The early versions will be vulnerable to anyone with the tools and patience to crack the hardware (e.g., get clear data on the bus between the CPU and the Fritz chip). However, from phase 2, the Fritz chip will disappear inside the main processor - let's call it the 'Hexium' - and things will get a lot harder. Really serious, well funded opponents will still be able to crack it. However, it's likely to go on getting more difficult and expensive." Two mistakes here: first, reading the bus to the TCPA chip cannot and will not reveal a private key. Private keys are generated on the chip, and never leave the chip unencrypted. But more importantly, TCPA was designed to protect the user's data from external attack, not from attack by the owner. Defending against owner attack is a ***much*** harder problem in hardware tamper resistance. TCPA chips have not been designed to resist local hardware attack, such as power analysis, RF analysis, or timing analysis. This is one of the examples that show that TCPA was not intended for DRM, which requires much higher levels of tamper resistance, since you don't trust the owner. Speculating that TCPA might add greater tamper resistance in the future is another example of pure speculation.

"You might prefer not to have to worry about viruses, but neither TCPA nor Palladium will fix that: viruses exploit the way software applications (such as Microsoft Office and Outlook) use scripting." While TCPA cannot prevent stupidity in software applications, it definitely can control the resulting damage. In particular, no virus can steal a TCPA protected private key. How can it, if the private key is generated in the chip, stored on the chip, and never leaves the chip? In addition, viruses that try to back-door or trojan the system to gain access to your sensitive data can be detected and blocked by TCPA, by its refusing to unseal sensitive keys in the compromised environment. The whole point of TCPA is to put security critical information into hardware, beyond the reach of malicious or broken software.

" Seen in these terms, TCPA and Palladium do not so much provide security for the user as for the PC vendor, the software supplier, and the content industry. They do not add value for the user, but destroy it." Personally, I find the ability to protect my private keys, and to protect my encrypted data very important and very valuable.

Presents speculation as fact:

"Fritz checks that the hardware components are on the TCPA approved list, that the software components have been signed, and that none of them has a serial number that has been revoked. If there are significant changes to the PC's configuration, the machine must go online to be re-certified." None of this exists anywhere in the TCPA specifications, or shipping product. While these things could theoretically be done on any operating system, to present them as existing fact, rather than the pure speculation that they are is irresponsible.

"There is one respect, though, in which you can't turn Fritz [TCPA] off. You can't make him ignore pirated software. Even if he's been informed that the PC is booting in untrusted mode, he still checks that the operating system isn't on the serial number revocation list." Here is more pure speculation presented as fact. There is no "serial number revocation list". Could someone create such a list? Yes, this could be done, with or without TCPA. There is no end to the infinite number of stupid things that could be done on a Turing complete system. TCPA simply does not do this.

"TCPA will undermine the General Public License (GPL)... To get a certificate from the TCPA consortium, the sponsor will then have to submit the pruned code to an evaluation lab, together with a mass of documentation showing why various known

> *attacks on the code don't work. (The evaluation is at level E3 - expensive enough to keep out the free software community, yet lax enough for most commercial software vendors to have a chance to get their lousy code through.) Although the modified program will be covered by the GPL, and the source code will be free to everyone, it will not make full use of the TCPA features unless you have a certificate for it that is specific to the Fritz chip on your own machine."* More wild speculation, presented as fact. The fact is that we are working on releasing TCPA code for Linux under the GPL. There is no such thing as TCPA certification of code; this is pure invention.

Lucky Green's Defcon Presentation [12]

> *"TCPA's Business Objectives:" are to "Prevent use of unlicensed software:", and "Digital Rights Management"*. The terms copy protection and DRM do not appear anywhere on www.trustedpc.org. They were not the main business objectives, and the resultant chip is not particularly suited to DRM, being poorly defended against owner tampering. The main goals are to secure the user's private keys and encrypted data against external software attack.
>
> *"[TCPA] chip: tamper resistant, surface mounted"*. Well, no. The TCPA chip is not particularly tamper resistant against owner attack. In the Common Criteria Evaluation target for TCPA, hardware tamper resistance is specifically not included as a goal. The IBM version ships as an LPC daughterboard, and is not specially protected against local hardware attacks. This was never a requirement, as the goal was to protect user data against external software attack.
>
> *"TCP OS Boot Process" diagram contains the constructs "Approved Hardware List", "Serial Number Revocation List", and "OS binary decrypt"* all presented as existing TCPA functions. None of them exist. This is, again, pure speculation presented as fact.
>
> *"PCI cards are TCPA-approved"*. Nope. Pure speculation presented as fact.
>
> *"Palladium is a TCPA Operating System"*. TCPA is operating system independent, and is already running in standard Windows and Linux systems.
>
> *"GPL... source alone is worthless without a TPM -specific certificate."* . The TCPA chip performs ***all*** functions without the use of an external certificate. While it is possible to write a DRM application that requires external certificates, there is no TCPA certificate authority, such applications do not exist, and this is all just pure invention again.

Bill Arbaugh's Comment [10]

Bill Arbaugh's comments are the most reasonable, and do offer some helpful suggestions. However, they also present some critical misunderstandings of TCPA.

> *"Both [integrity protection and trusted storage] use trusted root certificates as this basis [of their security guarantees.]"* This is a misunderstanding of the TCPA specification. There is no requirement for certificates at all, to use ***any*** TCPA chip function. There doesn't even exist such a root authority for TCPA in general, or for IBM's currently shipping chips. You can generate private keys, use them to sign, and decrypt, and seal/unseal data under PCR's, all without any certificates. The only time a certificate is needed is if you want to be able to prove to a third party that you have an approved TCPA chip. Most applications do not have this need, and this certification is not currently supported with IBM's chips. If you want to do an application that needs such a certificate, the TCPA has an endorsement key that can be used to get a suitable certificate. The only way this can work is if someone, like the manufacturer, has recorded a given TCPA chip's public endorsement key, and can use this knowledge to certify identity keys from the given TCPA chip. This is not required, and software access to the endorsement key can be disabled. There is certainly a privacy aspect of access to the endorsement key, as it uniquely identifies the platform, and the TCPA specification goes to great lengths to allow for anonymous certification. The best defense for privacy conscious users is simply to turn off the endorsement key.

Bill makes 5 specific suggestions:

1. *"Allow owners to load their own trusted root certificates."* The TCPA chip does not have or load any certificates. The only private key that cannot be cleared and arbitrarily loaded by the owner is the endorsement public key pair, which possibly is created on the chip at manufacture time, and the public part recorded by the manufacturer. It does not make any sense for the user to delete or replace the endorsement key, as only the original endorsement key recorded by the manufacturer can be used for this endorsement. If you want endorsement, you have to have that key. If you don't want endorsement, you can disable all access to the key. All other keys can be arbitrarily loaded and deleted. Note that IBM does not currently, and never has, recorded any endorsement keys, anyway, because no customers have asked for it.

2. *"Allow the TPM to be completely disabled."* This is certainly possible with the existing IBM version. It is an LPC bus daughterboard, which can simply be unplugged, and doing so clears all keys. (The TCPA BIOS will complain, and the TCPA-less version needs to be loaded.) Disabling the TCPA in BIOS, is more convenient. It does leave some harmless commands active, but all sensitive commands are disabled. It is completely under the control of the system owner. Under Linux, you can choose whether or not to load the TCPA device driver. If you don't load the driver, no application can access the chip at all. This is completely under your control.

3. *"Allow for complete privacy."* Disabling the endorsement key provides complete privacy. Ensuring complete privacy while using any form of endorsement key is clearly very difficult. The operations around the Endorsement Key are actually meant to protect user privacy by enabling the generation of multiple abstracted identities. The specification went to great lengths to define a process whereby the Endorsement Key functionality is limited to the generation of these identities only. A privacy CA can be selected by the user as the only entity that can link the Endorsement key with a specific identity. A different privacy CA can be used for each identity if desired. The user has complete control over the choice of if and how to use the endorsement key.

4. *"Work with the open source community"* . Absolutely! We have all the basic TCPA code running on Linux, and are actively working to GPL the basic driver and interface library.

5. *"Hold a technical workshop"*. This is a really good idea, particularly after (4) is done. TCPA is a complex device, and the specification is hard to understand, particularly the initialization/management parts.

Summary

- TCPA is not Palladium
- TCPA is not DRM. DRM is just one possible application of a trust component. Criticize DRM all you want.
- TCPA **does not**:
    - control execution
    - block execution based on signatures, or revocation lists, or approved lists
- TCPA **does**
    - provide protection of a user's private keys and encrypted data
    - protect sensitive data from many software attacks, including viruses, worms and trojans.

References

TCPA

[1] TCPA website/specifications:

http://www.trustedcomputing.org
http://www.trustedcomputing.org/docs/main%20v1_1b.pdf
http://www.trustedcomputing.org/docs/TCPA_PCSpecificSpecification_v100.pdf

Palladium

[2] Seth Schoen's "Palladium Details":

http://www.activewin.com/articles/2002/pd.shtml

[3] Microsoft Palladium White Paper:

http://www.neowin.net/staff/users/Voodoo/Palladium_White_Paper_fina

[4] Paul England and Marcus Peinado, Microsoft, "Authenticated Operation of Open Computing Devices"

http://link.springer-ny.com/link/service/series/0558/papers/2384/23
Batten and Seberry (Eds.), ACISP 2002, LNCS 2384, Springer-Verlag, 2002

[5] Peter Biddle, comments on Cryptography list:

http://www.cl.cam.ac.uk/~rja14/biddle.txt

DRM

[6] Ed Felton's paper on SDMI challenge:

http://www.usenix.org/events/sec01/craver.pdf

[7] Fravia's Reverse Engineering site:

http://www.woodmann.com/fravia/

Anti TCPA

[8] Ross Anderson's "TCPA/Palladium Frequently Asked Questions":

http://www.cl.cam.ac.uk/users/rja14/tcpa-faq.html

[9] Anti TCPA Website:

http://antitcpa.alsherok.net/

[10] Bill Arbaugh's TCPA criticism:

http://www.cs.umd.edu/~waa/TCPA/TCPA-goodnbad.pdf

[11] John Gilmore's comments against TCPA

http://lists.w3.org/Archives/Public/www-drm/2001Jan/0009.html

[12] Lucky Green's Defcon presentation

http://www.cypherpunks.to/TCPA_DEFCON_10.pdf