EFF Comments on TCG Design, Implementation and Usage Principles 0.95

Seth Schoen
Electronic Frontier Foundation

October 1, 2004

> Each user of computers must decide what security means to him. [...] Since there are many different sets of needs, there can't be any absolute notion of a secure system.
>
> — Butler W. Lampson, "Requirements and Technology for Computer Security"

1 Introduction

The Electronic Frontier Foundation (EFF) is pleased to submit these comments on the Trusted Computing Group (TCG)'s "Design, Implementation, and Usage Principles" draft version 0.95.

TCG, together with its predecessor the Trusted Computing Platform Alliance (TCPA), has been the subject of intense criticism in the on-line community over projected applications of its trusted platform module (TPM). Most of these applications are speculative; some are possible within the TCG specifications, and others are not. Many depend on what platform or operating system vendors do. Some abusive applications might be subject to market discipline, while others could be tolerated by many consumers or be largely invisible most of the time. In some cases, consumers might have little recourse because of limited and imperfect competition.[1]

[1] This is especially true in the case of creative works, whose authors and those to whom they have transferred their rights enjoy a broad legal monopoly over the first sale of their respective works of authorship. As Karen Coyle wrote of an analogous case ("A Response to 'P3P and Privacy'"), "choices are and will be limited on the Web [because] information services tend to be unique. Because of the nature of intellectual property and copyright, there is generally only one outlet for an information resource. This is something that is often missed even by economists when they discuss the market model in an information environment. If I want to read the New York Times online but don't like their privacy practices, it doesn't do me any good to read another newspaper instead. My choice is simply to give up my personal data or to not get the product. [...] In the case of information resources that are only available electronically, I have no alternative format."

EFF previously commented on a related, although vendor-specific, policy and best practices document. We refer the reader to our comments on the LaGrande Technology Policy on Owner/User Choice and Control, which are attached to this document.[2] Although Intel's LaGrande Technology (LT) is just one application of a TPM, most of our concerns about LT are related to its use of TPM functionality and are applicable to the design of the TPM itself. Many of our concerns about the limitations of the LT Technology Policy are also relevant to the draft TCG Principles.

In these comments, we discuss the risks trusted computing creates, outline some ambiguities and limitations of the draft version 0.95 of the Principles, and provide an overview of technical approaches toward making trusted computing harder to abuse.

We believe that some, although not all, of the Internet community's criticisms of TCPA and TCG are well-taken, and that trusted computing as currently conceived by industry will create some important risks for the public if it is widely deployed and used. We expect that some parties may use trusted computing in the ways the Principles describe as "coercion." Some may also used trusted computing to diminish competition by creating new barriers to interoperability and new obstacles to reverse engineering and emulation.

TCG's Principles fail to clarify which uses of trusted computing are appropriate. They provide little guidance about which business rationales might justify controversial practices, and they mostly avoid confronting the inevitable prospect of conflicts of interest between parties. Since the Principles lack specificity about several important issues, and since organizations have a favorable view of their own motives, it is hard to imagine implementers readily agreeing that any particular decision has run afoul of the Principles.

The Principles cannot (and do not claim to) mitigate all the potential harms of trusted computing, so even as TCG strengthens and clarifies the Principles, it should also consider technical changes that will deter abuses. TCG's future technical work should reflect a commitment to design based on the Principles; they should inform the creation of the technology, not be an afterthought to it.

2 Current TPM features can readily be abused

Widespread deployment of TPMs creates at least two kinds of risks to the public, both of which are mentioned by the Principles: a shift in the historic balance of power within the personal computer platform, and a new set of threats to competition and interoperability. We will consider these dangers here in some detail.[3]

[2] For readers outside of TCG, our LaGrande Policy comments can be found at http://www.eff.org/Infrastructure/trusted_computing/eff_comments_lt_policy.pdf.

[3] We previously discussed some of these problems in "Trusted Computing: Promise and Risk."

2.1 Altering the balance of power

The Principles acknowledge that TPMs will change the world in important ways that the public may not find congenial.

> It might be argued that a claim that TCG deployment will not interfere with established practices of data ownership is (at best) disingenuous: by providing a significantly more effective mechanism for data owners to enforce their intellectual property rights, TCG technologies may be held to significantly alter the balance of power in practice.

Regardless of which claims would be disingenuous, TCG technologies *are* likely "to significantly alter the balance of power" between consumers and suppliers of goods and services. The balance of power in personal computing has traditionally included the fact that *everyone* was uncertain about which software was actually running on a particular computer. Some of that uncertainty is considered undesirable by almost everyone but industrial spies and the authors of computer viruses. However, that uncertainty also prevented software publishers from having particularly powerful tools to limit modification and reverse engineering. It prevented them from detecting whether an execution environment was real or virtual. These disabilities on the part of publishers – which they have regretted and spent vast sums trying to work around – have been welcomed by consumers, who have enjoyed a corresponding ability to exercise fine-grained control over software running on their PCs.[4]

The prospect of reliable proofs of code identity is an exciting one in many ways. It is also a threat to the consumer benefits that have resulted from uncertainty. It represents a powerful tool for publishers who want to detect, and thereby obtain leverage to punish, consumer behaviors that violate *publishers'* security policies. We emphasize below that there are no limits in principle to what those policies might require. The changes TPMs may produce in the balance of power are therefore not limited to the effectiveness of technical mechanisms to enforce intellectual property rights.[5]

Here are just a few contexts in which the cocktail of TPM features will shift the balance of power away from PC owners, adding a qualitatively new robustness to various anticonsumer measures:

[4] For example, "no software running under an emulator shall be able to distinguish the emulated environment from a physical PC" is a widespread security policy that consumers have long used to protect their investments in hardware and software against lock-in and planned obsolescence. Or again: "software should not be able to tell whether it is running under a debugger" has been a perfectly plausible requirement on the part of a PC owner for years.

[5] Publishers might, of course, try to describe their behavior as an attempt to enforce intellectual property rights, by taking an extremely expansive view of what kinds of activities those rights constrain. See, *e.g.*, David Nimmer, Elliot Brown, and Gary N. Frischling, "The Metamorphosis of Contract into Expand," 87 California Law Review 17 (1999), and Julie E. Cohen, "*Lochner* in Cyberspace: The New Economic Orthodoxy of 'Rights Management'," 97 Michigan Law Review 462 (1998). As the Hon. Alex Kozinski put it not long ago, "property owners are very grabby."

• They will strengthen digital rights management (DRM) by providing primitives particularly useful to DRM, including a readymade framework for document tethering and powerful new defenses to traditional attacks against DRM systems.[6]

• They will strengthen spyware, helping it enforce its policies, protecting it against anti-spyware scanners, firewalls, and proxies. They will help spyware developers hide the functionality of their spyware from anti-spyware researchers, and conceal what sorts of data a spyware program is transmitting or might transmit.

• They will strengthen product activation and tethering, and facilitating forced upgrades and downgrades. They will help prevent consumers from moving data and software from one place to another, and help those software developers who so choose to remotely deactivate applications forever. They will allow hardware failures, or simple hardware upgrades, to mean the risk of permanent loss of documents or software.

• They will help inhibit emulation, which has been endorsed repeatedly by courts as a valuable procompetitive technology. They will allow software written for one platform to die with that platform and deter with cryptography efforts to prolong that software's life and utility.

2.2 Anticompetitive effects of trusted computing

The Principles say that TPM applications "should certainly not introduce any new interoperability obstacles. [...] TCG has set the modest, but extremely important, goal of not making things worse." Mindful of the likelihood that these applications could indeed make the interoperability situation worse, the Principles call for avoiding "the introduction of artificial barriers to interoperability." But, just as we can expect no consensus about what constitutes a valid security rationale, we can expect no consensus about what is an "artificial" barrier. To some people, controlling or precluding the creation of third-party client software seems normal and rational, not artificial.[7]

[6] TCG has emphasized that the TCG architecture is not a DRM system and that DRM is merely a potential application (among others) of a TPM. This is true, but it does not diminish the observation that TPMs will strengthen DRM and that some people have looked forward to TPM deployment for precisely this reason. We regret that DRM was cited with approval as a TPM application in TCPA's whitepaper "Building a Foundation of Trust in the PC."

[7] The video game world, for instance, has a particularly dismal record on interoperability; U.S. caselaw on reverse engineering is replete with good law made when courts rejected video game firms' invitations to limit third-party interoperability. These firms did not think they were doing anything strange or artificial by trying to lock out third-parties; rather, they were doing what came most naturally to them – or, as they are fond of putting it, simply protecting their long-standing business models. In their view, the prospective competitors were intruders on their markets and into their products as surely as a script kiddie hijacking your browser would be.

Apple Computer recently expressed a similar attitude when it said that reverse engineers at RealNetworks had "adopted the tactics and ethics of a hacker to break into the iPod"; Prof. James Boyle, writing in the *Financial Times*, wondered

Competition in a modern high-technology market is a very complicated thing indeed.[8] We might imagine that it's possible to decide in some neutral or absolute sense whether something is anticompetitive, procompetitive, or neutral in its competitive effects. This is wishful thinking. First, it can be difficult to decide what is a relevant market, but this is a prerequisite to trying to say whether that market is competitive.[9] Consumer advocates often criticize the possession of *any* substantial market power by firms in *any* market, even if antitrust law does not consider that market power a cause for concern. Second, antitrust laws vary from place to place. A court can make a legally binding final determination about whether some behavior is illegally anticompetitive under the laws of a particular jurisdiction at a particular time, but that determination is hardly the last word on the competitive effects that an action will have in various markets over various periods of time.

Can there be a consensus about which barriers to entry are "artificial?" Every firm has an incentive to view and describe its own behavior as procompetitive, at least in the eyes of regulators and the public. A rival may have an equally strong incentive to see that same behavior as anticompetitive. The original firm may then point to some allegedly legitimate business reason for its behavior, but getting to the bottom of the question of whether that behavior was anticompetitive in its purposes or effects is still another matter. In any case, some firms are willing to concede that their actions create certain barriers to entry, but justify those barriers in terms of "protecting the incentives to investment" or "protecting the viability of new business models."

2.3 Code identity and the economics of interoperability

To see the relevance of code identity (such as the proofs of platform and code identity made possible by the TCG TPM's attestation features) to competition, it isn't necessary to assume the existence of some villain out to sustain a monopoly.[10] Code identity can have anticompetitive effects *inadvertently* by

> How exactly had Real "broken into" the iPod? It hadn't broken into my iPod, which is after all my iPod. If I want to use Real's service to download music to my own device, where's the breaking and entering? What Real had done was make the iPod "interoperable" with another format. If Boyle's word processing program can convert Microsoft Word files into Boyle's format, allowing Word users to switch programs, am I "breaking into Word"? Well, Microsoft might think so, but most of us do not. So leaving aside the legal claim for a moment, where is the ethical foul? Apple was saying (and apparently believed) that Real had broken into something different from my iPod or your iPod. They had broken into the *idea* of an iPod. (I imagine a small, Platonic white rectangle, presumably imbued with the spirit of Steve Jobs.)

[8] See, e.g., Carl Shapiro and Hal R. Varian, *Information Rules: A Strategic Guide to the Network Economy* (Cambridge, MA: Harvard Business School Press, 1998).

[9] One advocate might say that it's important to protect competition in the market for books or the market for printers, where another might say that it's important to maintain a competitive market for Lexmark printer cartridges.

[10] That would-be monopolist would probably appreciate the ability to verify code identity, since things that serve to differentiate products tend to increase market power and things that

altering the economics of interoperability. It can create barriers to entry and platform and application lock-in; it can significantly increase the cost of switching one's underlying hardware or software configuration; it can contribute to the marginalization of minority platforms. It can do all these things as a mere byproduct of its use for other purposes, even where no part of a security policy may contain any explicitly anticompetitive motives or purposes. One source of these effects is the cost of the research necessary to decide whether a given attested configuration is appropriate or inappropriate. That cost is a marginal cost for an attestation verifier when deciding to accept a new platform. Depending on the security properties of interest, it might cost millions of dollars to demonstrate to a verifier's satisfaction that a new platform or application has security properties equivalent to the old. Those research costs would have to be borne by someone.

When someone is developing a new application that will make use of attestation, there is an obvious economic incentive to begin by researching the security properties of existing platforms and their configurations (especially *expected* or at least *widely deployed* configurations). There is a much smaller incentive to use resources to research the security properties of minority or unexpected platforms. This has the potential to create a vicious cycle in which minority platforms are poorly supported by applications that use attestation, and therefore become even less popular.

Some people would argue that these effects are inherent in every situation in which users of one platform want to interoperate with users of another, or with an application that may not have anticipated their existence. But code identity changes the situation in at least two important ways, both of them unfavorable to minority platforms. First, code identity verification prevents minority platform users and developers from taking *unilateral* actions, such as reverse engineering, to achieve interoperability. Second, code identity verification reduces the usefulness of standards for promoting interoperability. Absent verification of code identity, after all, minority platforms interoperate because developers or users of those platforms expend resources to guarantee interoperability – through mechanisms like standards development processes, experimentation, user innovation, and reverse engineering. But those resources would be expended in vain in the presence of code ID because code ID can serve as an absolute barrier to interoperability, making interoperability subject only to the *challenger's* expenditure of resources and nobody else's.

For instance, in the past Samba developers expended considerable effort learning how the SMB protocol worked and creating an independent implementation that created significant value for its customers, much of it as a result of interoperability with the majority platform; the majority platform developer, and non-Samba users in general, were not required to do anything to promote interoperability, but interoperability was achieved. If code identity verification were present here, specific action by Windows developers or Windows server operators would be required, which would consume *their* resources; this would

make products interchangeable tend to decrease it.

means that code identity provides an economic deterrent to interoperability because the willingness of people on the majority platform side to take potentially expensive pro-interoperability actions on an ongoing basis is uncertain, whereas the propensity of people on the minority platform side to do so is always assured. In other words, code identity verification shifts some of the costs of achieving interoperability onto those who have the least incentive to bear them.

incumbent or majority platform developer is	**without code ID, or code ID unused**	**with code ID**
supportive towards interoperability	interoperability supported	interoperability likely supported (but barriers to entry may still be present)
apathetic towards interoperability	minority platform developer expends resources to achieve interoperability	interoperability may be precluded inadvertently
hostile towards interoperability	majority platform developer expends resources to frustrate interoperability minority platform developer expends resources to achieve interoperability some interoperability likely possible in long run	interoperability may be precluded deliberately

Figure 1: Likely effects on interoperability of the use of proofs of code identity.

The Principles and other best practices documents *can't alter the economics of this situation* or the incentives that follow from it; even someone with the best intentions may still ultimately be party to diminished competition and interoperation. Thus, as Figure 1 summarizes, code identity very likely will make the interoperability picture worse – whether incumbent platform developers are pro-interoperability or anti-interoperability.

Let's consider an example of a relatively benign use of code identity. Developers of multiplayer on-line video games want to deter cheating (violation of predetermined game rules) by determining code identity. In many video games, a client is trusted with information that the rules of the game indicate should not be disclosed to the player. What's more, in some video games, a client's messages are trusted to conform to the rules of the game, so that a client that sent inappropriate messages could cause a violation of the integrity of the game. Video game clients may not be trustworthy because users can cheat by modifying them. Successful verification of code identity can change that situation by assuring that clients that participate in a game are trustworthy.

This security goal – one that most video game players would readily endorse

– is in tension with the imperative of interoperability. A commercial game publisher may feel that it derives a commercial benefit from limiting interoperability with its games wholly apart from increasing the perceived difficulty of cheating.[11] Even a game developer who wholeheartedly endorses third-party development is in a quandary. For example, the creators of Netrek, a venerable open-source on-line multiplayer game, have long used a relatively primitive pure software technique for verifying the identity and integrity of clients.[12] Netrek developers strongly believe that third parties should be able to create their own compatible Netrek software that can be used in on-line games. They also believe that verifying the identity of client software benefits Netrek by making cheating more difficult. These beliefs are in tension because new clients must be carefully examined and approved, or must be produced only by trusted parties.[13]

3 Absence of specific guidance in the Principles

We have already mentioned that the Principles fail to resolve the question of which barriers to interoperability are "artificial." In a similar way, they fail to clarify which demands for attestation are inappropriate "coercion," and which kinds of policies represent appropriate "security goals." These ambiguities undermine the usefulness of the Principles for making real-world decisions. The uncertainties can be resolved, if at all, only by making a decision about whose notion of propriety should control.

3.1 The Principles on "coercion"

The Principles decry "[t]he use of coercion to effectively force the use of the TPM capabilities," but then create a hole large enough to fit an industry through: the use of coercion is acceptible where it is "completely appropriate." Apparently the propriety of "us[ing] market clout [...] to essentially *force* the use of TPM technology" depends on whether an industry believes that decision is "a necessary component of remaining in business." But most firms characterize *all* their significant decisions, especially those that arouse a controversy, in precisely that

[11] For instance, the publisher might believe that interoperable third-party software will tend to take away market share, or provide competition in case the publisher chooses to expand into new markets.

[12] See, e.g., http://www.netrek.org/downloads/clients/ ("Netrek was developed according to the Open Source model [...] To prevent cheating, all official client binaries authenticated themselves to the server via RSA. They are called 'blessed' clients.").

[13] The fact that Netrek is open source is not as helpful as we might expect to would-be developers of new clients, because Netrek servers are able to distinguish between officially-sanctioned and unofficial binary images, and permit only the former to interoperate. As a result, end users are unable to use the Netrek source code to create clients that can actually participate in network games with others. Thus, the verification of code identity can cause third-party interoperability with open source software to be limited in ways traditionally associated with proprietary software, even where diminishing interoperability is not an explicit security goal. It so happens that Netrek's software-based "blessed client" system is not a particularly robust security measure, but our point would apply with equal force if Netrek developers decided to rework Netrek as a TPM application.

way. Worse, leaving this hole risks creating a self-fulfilling prophecy: businesses that fear that competitors will derive some kind of market advantage by forcing customers to provide attestations may well conclude that business necessities demand that they follow suit. A defense of business necessity for an otherwise reprehensible practice seems to invite a race to the bottom.

The Principles offer a pecular example of a purported business necessity for coercion: "a bank," they suggest, "that did not insist on remote attestation might not stay solvent for long." This claim is particularly odd given that *banks do not insist on remote attestation now.* While banks have been subject to a variety of on-line fraud and identity theft attacks, they have experimented with a wide variety of approaches to mitigating these attacks. They do not seem to have concluded that authenticating customers' application software is a requirement for solvency. Even the recent rise of "phishing" has not led to any widely-publicized bank failures. We do not dispute that banks and their customers might derive security benefits by using TPM functions, but (especially where customers bear some of the risks of fraud) we have suggested that this is no excuse for forcing customers to use code identity features. The financial industry has often found that customers were eager to adopt security measures voluntarily, and we see no reason to believe that customers would not adopt TPM-based security features without coercion if they were implemented in a way that served customers' interests.[14]

The idea that coercion is unacceptible except where it turns out to be appropriate is reminiscent of the idea that limits on interoperability must be associated with a "security goal." Neither offers useful guidance.

3.2 What is a valid security rationale?

The Principles try to address the problem of TPM applications that thwart interoperability in this way:

> Applications using TCG-specified capabilities to determine system configuration should base their decision to interoperate on the conformance of the measured configuration only for the purpose of clearly-articulated security goals.

This suggests, for instance, that someone should not deploy an application that makes itself unavailable to Apple Macintosh users *solely because* they are Macintosh users, as opposed to on account of their inability to meet some sort of security requirement. Nor, we presume, should someone following the Principles

[14] We reiterate below that TPMs should be designed in a way that provides an *architectural* deterrent to their use in ways the Principles call "coercion." These improvements would form no barrier to the use of TPM security features by a bank. In any case, banks have always had an array of security measures available to them and the expertise to determine which are most appropriate and valuable. Reposing the whole future of the banking industry in the receipt of proofs of code identity seems like hyperbole.

deploy a public service that will forever accept communication only from a single pre-specified client application.[15]

However, this principle is undermined by its ambiguity. First, it does not specify *to whom* a security goal must be "clearly-articulated." One interpretation is that the security goal must be specific, and hence capable of a clear statement, but that it need not actually be communicated to anyone – that is, it should actually *exist*, but there is nobody who is necessarily entitled to hear it.[16]

Another interpretation is that the implementers of an application must have articulated the security goal upon which the "decision to interoperate" is based – but only within their own organization, and not to the general public.[17] Yet another possibility is that the security goal (but perhaps not the details of its implementation) must be communicated or available to particular users; and perhaps the Principles call for a truly public disclosure of both the security goal and the measures used in its implementation. But the Principles have not revealed whether one of these interpretations, or some other interpretation, is correct.

Second, it is not evident what kinds of things the Principles would view as "security goals." As we emphasize below, anything can be characterized as a security goal, and the phrase is not specific enough to have a definite meaning beyond that of something like "desired outcomes."

In other words, the Principles seem to call on application designers to have a specific reason to believe that the code they write will, to the extent it uses TPM features, apply those features to achieve the outcomes the designers intend. We trust that TCG intended to say something more specific than this!

3.3 Can there be consensus about security goals?

Commentators agree that security is observer-relative, not absolute. Sometimes this fact is obscured by the existence of an apparently widespread and stable consensus about whether certain security measures are desirable. But as security experts like Butler Lampson (*supra*) and Bruce Schneier observe, security goals are human goals; different people have different, and potentially conflicting,

[15] Analogously, the section of the Principles discussing key migration says that "keys should be marked non-migratable only where there is a clear security requirement." We take "clear security requirement" as expressing the same idea as "clearly-articulated security goals," and we consider it vague for the same reasons. Who decides whether the non-migratability of keys is a security benefit or, in the case of a disaster or a hardware failure, a serious threat to availability?

[16] In the law of searches and seizures, a similar rule applies to the decisions of law enforcement officers to make so-called Terry stops: the law enforcement officers must be *able* to articulate a reason for detaining a particular person, but they are not actually required to reveal that reason at the time the person is detained. "And in justifying the particular intrusion the police officer must be able to point to specific and articulable facts which, taken together with rational inferences from those facts, reasonably warrant that intrusion." Terry v. Ohio, 392 U.S. 1, 21 (1968).

[17] Presumably one engineer would turn to another and explain when and why the application they're developing will refuse to work with certain other applications, and the substance of that discussion would thereafter be viewed as a trade secret.

security goals and requirements.[18]

The security literature does not reveal any conceptual limits on the kinds of things that can potentially be made a part of a security policy.[19] A security policy need not coincide with any jurisdiction's law or any notion of morality; it need not depend on any cultural norms; it does not have to approved by a majority vote.

Consider the recent controversy over the Apple iPod and the RealNetworks Harmony technology. This controversy involves applications of cryptography and security-related technologies, but everyone involved – at least Apple Computer, RealNetworks, the recording industry, consumers, independent labels who don't use digital rights management technology at all, and independent software developers such as open source programmers – perceives a slightly different interest in the outcome. All these parties may have contrasting security requirements, based on which uses of the iPod (and of musical works) they consider appropriate or beneficial.

The Principles also mention the use of assistive technology by people with disabilities. In the past, some people have denied that the creation and integration of assistive technologies justifies acts of reverse engineering. This view is abhorrent to us, but we cannot deny that a policy with the effect of preventing the development of assistive technologies is still a *security* policy. From the point of view of a platform developer, people creating unauthorized means of facilitating accessibility may well be considered attackers, and keeping them out may well be described as a security goal.

It is thus essential for anyone assessing a security system to ask whose security policy is being enforced against what sorts of attackers. (This is another way of saying that the threat model needs to be clearly specified.) TCPA and TCG have so far not engaged their critics on this question. The presence of tamper-resistance requirements in the TPM specification raises the prospect that some people will support TPM adoption precisely because a TPM can help

[18] As Schneier writes in *Beyond Fear*, "[s]ecurity systems are never value-neutral; they move power in varying degrees to one set of players from another." He also emphasizes the observer-relativity of security. "Security requires the concept of an *attacker* who performs [...] unwarranted actions. It's important to understand that this term is meant to be nonpejorative, value-neutral. Attackers may be on your side. [...] The term *attacker* assumes nothing about the morality or legality of the attack." Schneier also mentions that security rationales are tied to someone's "agenda," which may be distinct from, and in conflict with, someone else's agenda.

[19] The *enforcement* of a policy is a separate question. Someone who tried to enforce a security policy that prohibited the British royal family from occupying Buckingham Palace might find it a significant challenge. Similarly, on a traditional PC running a contemporary operating system, someone trying to enforce a security policy forbidding the PC's owner to make a byte-for-byte copy of RAM into a file on a permanent storage medium has a formidable task. So policies may all be equally easy to promulgate, but they are not all equally easy to enforce within a given status quo. The tendency of an architecture to facilitate the enforcement of some kinds of policies more than others is the reason Schneier says that security technologies "move power [...] to one set of players from another," and the reason we argue below for design changes to the TCG specification. See also Lawrence Lessig, *Code and Other Laws of Cyberspace* (New York: Basic Books, 1999) for the most influential recent statement of the power of computer architecture to regulate behavior.

enforce policies against its owner. Those policies are security policies, and their enforcement is a security goal, just as a computer owner's attempt to forbid local software to keep any cryptographic key material private from her is a security policy and its enforcement a security goal.

There is no sense in which the conflicts in these security policies can be eliminated.[20] They are based on real disagreements between people about which outcomes are desirable.

3.4 What TCG can do

We have argued that real conflicts of interest exist within computer security and that proofs of code identity inevitably create competition concerns. TCG cannot eliminate disagreements about what is "appropriate" or "competitive." But TCG *can* make an unequivocal statement – in the Principles and in the design of its technology – that the ultimate power to make decisions about these questions will be left in the hands of the owner of a PC. After all, the owner is the one being asked to purchase a trusted computer; TCG can thus conclude that the owner is the entity that should have final responsibility to establish security policies.

The Principles already say TCG stands for "the owner's security policy" and that"it is the owner whose security policy is enforced." Perhaps the Principles are suggesting that an application developer's security policy (even if the owner dislikes it) automatically becomes a part of the owner's security policy whenever the owner chooses to use that application. That idea would lead to the strange conclusion that the actions of viruses and spyware comply with a computer owner's security policy merely because the owner decided to run them. Some spyware companies have actually made this argument, but we do not imagine that the drafters of the Principles meant to lend support to it. If TCG believes that its goal is the enforcement of the *owner's* security policy – as against the policy of an application developer or service provider, come what may – it must commit itself to producing a technology that actually does that. We will offer technical proposals toward this goal below.

4 The Principles have no built-in enforcement or publicity mechanism

We have discussed several reasons why even those who attempt to comply with the Principles may find that they offer a lack of concrete guidance.

Like the LT Policy, the TCG Principles admit at the outset that they do not contain, and that TCG does not possess, a means of enforcing them. To date, they also lack a means of publicity, although TCG could change that in the future. This means that many TPM application developers may never even

[20] They might be "resolved," in some sense, by law, but they would still be essentially present.

hear that TCG has promulgated a set of best practices. And, if they do find out about the Principles, they may well ignore them.

It is true that best practices documents can have a positive effect on an industry.[21] But the history and economics of the IT industry do not make us optimistic that procompetitive applications of TPMs – given their current architecture – will prevail without some kind of enforcement mechanism.[22]

5 Limitations of transparency

The Principles call for transparency to let users know what kinds of key migration policies a TPM will be used to enforce. We endorse transparency here as elsewhere, but we note the limitations of this principle. It suffers from the general problem that application developers remain free to ignore it. And compliance even with a generalized version of this principle – that TPM applications

[21] The Principles mention the example of RFC 1918, "Address Allocation for Private Internets," which codified a convention of using particular IP addresses for private networks. This decision has remained controversial; see, e.g., RFC 1627, "Network 10 Considered Harmful: Some Practices Shouldn't be Codified." RFC 1918 may have been particularly effective for at least two unusual reasons. First, the Internet technical community was relatively small and close-knit in 1994, when RFC 1918's predecessor, RFC 1597, was issued. That community was accustomed to informal technical co-ordination and had institutions and events that facilitated that co-ordination. The Internet community today is more far-flung and heterogeneous, and it isn't clear whether a Best Current Practices RFC document issued today would have the same sort of influence on Internet architecture that it would have had in the past. Second, RFC 1918 and RFC 1597 can be read as promises by the Internet Assigned Numbered Authority to reserve the three network blocks mentioned therein for private uses in perpetuity and to refrain from allocating them. In that sense, these RFCs could be viewed as unilateral offers by IANA and the regional IP address allocation authorities to the Internet community to use these networks without risk of conflict. IANA had special authority to make such an offer, and individual network operators could independently choose to take advantage of it or not to do so.

Tellingly, the *only* particular recommendations in RFC 1918 that purport to impose obligations on the public have been widely disregarded. RFC 1918 provided that,

> [b]ecause private addresses have no global meaning, routing information about private networks shall not be propagated on inter-enterprise links, and packets with private source or destination addresses should not be forwarded across such links. Routers in networks not using private address space, especially those of Internet service providers, are expected to be configured to reject (filter out) routing information about private networks. If such a router receives such information the rejection shall not be treated as a routing protocol error.
>
> Indirect references to such addresses should be contained within the enterprise. Prominent examples of such references are DNS Resource Records and other information referring to internal private addresses. In particular, Internet service providers should take measures to prevent such leakage.

Many networks routinely disregard this policy and many network operators are totally unaware of it.

[22] As we pointed out in our comments on the LT Policy, many of the members of TCG are well accustomed to using patent and trade secret licensing to try to ensure that technologies they develop are used in ways they consider appropriate. We find it remarkable that those so well versed in technology licensing have not found a way to enable enforcement of these Principles.

and TPM application developers should disclose what policies they enforce – would be insufficient to avoid consumer harm for at least two reasons.

5.1 OS projects like NGSCB pass TPM functions through to higher layers

Many TPM applications, including Microsoft's NGSCB operating system security project, will abstract TPM functionality for the benefit of higher-level applications, providing security services that use or encapsulate TPM functions. A TPM application that exposes code identity verification functions as a service to higher-level applications would then allow those applications to do things that the TPM application itself did not understand or control. As a result, a TPM application like NGSCB might itself be implemented in a way that complies with the Principles, but third party applications that take advantage of it might not.

Indeed, because of this process of abstraction, not all higher-layer TPM applications may even be aware that they are using TPM resources!

To put this another way, TPM applications that pass TPM functionality through to other applications may ultimately not know which policies they are implementing. This situation would frustrate the aim of transparency, because you can't disclose to a user information that you don't have. To remedy this, the principle of transparency would need to be implemented *recursively*, so that all applications that make use of TPM functionality, even indirectly, are subject to the same transparency rules as is software that communicates directly with a TPM.

5.2 Secrecy is not the only source of consumer harm

Transparency is a traditional sort of consumer protection, but it is not always enough. Today, vendors are often quite blatant about the kinds of policies they are trying to enforce with technology. Those products that come with End-User License Agreements, for example, often present the user to with a litany of purportedly eradicated consumer rights. Vendor all too often simply trumpet the fact that they plan to restrict their customers.[23]

Even when transparency requirements are followed and consumers are informed about the key migration policies that an application tends to enforce, the relevant disclosures would likely be buried amidst other information and seem relatively unexception. In effect, a program would simply mention the fact that it intends to use the TPM as an additional technical means of enforcing the restrictions it has previously pledged to impose. This sort of needle in the EULA haystack will hardly improve the consumer's lot.

[23] This is easy to see: just take a look at the shrinkwrap or clickwrap license associated with a mass-market software product, or, say, any on-line music service endorsed by the major U.S. record labels.

6 TCG should examine technical means to mitigate abuses of attestation

The Principles state that "TCG technologies and mechanisms were designed with a strong bias towards supporting implementations that follow the design principles discussed in this document" and that "there are particular design features in the TCG specifications that are in direct support of the principles [... and] are there to not only support but also to bias implementations in favour of a pricipled usage of the technology." We are aware of a few such design features, particularly those aimed at avoiding making attestation a tool for invading privacy by linking transactions. TCG should go further.

TCG can and should act now to improve the design of the TPM to resist abusive and coercive uses; it will be some time before TPMs are deployed widely enough to create major threats for consumers, and there is time to make significant changes in hardware before then. TCG is capable of finding technical means to reduce the feasibility of uses of TPMs that reduce consumers' control over software they might run on their computers. We believe that the most promising approach to this problem is a commitment to enforcing the computer owner's security policy in preference to, and to the detriment of, any other security policy.

There is a widespread but mistaken assumption that doing so would be fatal to the enterprise applications that prospective trusted computing vendors consider the most important trusted computing market. First, most enterprise trusted computing applications are not in conflict with trusting the PC's owner. Second, there may be ways to supporting applications in which the PC's owner is untrusted that do not lend themselves to widespread use in the consumer market.[24]

We offer four technical approaches to this problem for illustrative purposes.

6.1 Using only sealed storage

TCG could simply remove the attestation feature entirely, and use only sealed storage as a proxy for code identity.

In this case, the TPM owner would manage trust directly by using trusted media to boot a PC and then creating and sealing secrets. Those secrets can serve as proxies for code identity by proving that a configuration corresponds to a prior configuration that was obtained by a physically present trusted machine administrator (or, depending on security requirements, merely by the use of a particular trusted medium to boot the system). TPMs that do this would still provide major security benefits over the current PC platform, including most consumers applications advocated by TCG member companies.

An application would prove its identity – and the integrity of the software configuration – by unsealing a secret available only to it and then engaging in any

[24] Nonetheless, in our view, the interorganizational enterprise DRM market that some have described as the leading "untrusted owner" application for attestation is mostly speculative.

standard cryptographic challenge-response protocol to prove that it knew that secret. (Alternately, it could unseal a private key and use that key to generate cryptographic signatures.) Such a challenge could only be made by someone who was or who trusted the machine's owner, because only the machine's owner would know – and be capable of stating authoritatively – which secrets or keys had which meanings.

6.2 Owner override; owner gets or generates key

The most worrisome thing about TCG attestations is not that they facilitate trust in a device, but that they decouple that trust from trust in the device's owner. Most TPM applications described in the press and in presentations by TCG member companies are not intrinsically incompatible with trusting the owner. We have suggested that TCG can improve the situation by changing the meaning of attestation to include making trust in attested PCR values contingent on trust in the TPM owner.[25]

Attestation would then mean that a hardware or software configuration hasn't changed from its expected value *without the knowledge and consent of the platform owner* – but it might have been changed deliberately by the platform owner, who may have chosen not to reveal that change. This sort of attestation is useful for conventional security applications and is a strictly improvement over the status quo: with this sort of "owner override," the owner has a means of conveying authentic information to facilitate certain applications, but cannot be induced to do so when it might be against her interests.

An alternative that is functionally similar involves providing the attestation identity private keys to the TPM owner. This would allow the TPM owner (but only the TPM owner) to generate valid signatures that are indistinguishable from TPM-generated signatures, so that an attestation verifier is compelled to trust the TPM's owner in addition to the TPM itself. Another alternative is to allow the TPM owner to generate attestation identity keypairs by some means external to the TPM and then to load copies of those keys into the TPM after generating them. (We call these schemes "owner gets key" and "owner generates key.") These proposals change the TPM key management model, but do not seem to require engineering changes that are particularly large by the standards of the TPM specification. All of them have the effect of giving the TPM owner technical means to decide which security policies may be enforced on her computer, without allowing third parties to punish the owner for her choices.

[25] We first discussed the owner override concept in "Trusted Computing: Promise and Risk," available at http://www.eff.org/Infrastructure/trusted_computing/20031001_tc.php; see also "Give TCPA an Owner Override," *Linux Journal* 116 (December 2003), available at http://www.linuxjournal.com/article.php?sid=7055. We acknowledge that the owner override proposal would require user interface research that has not yet been performed and that other alternatives might provide similar benefits while requiring fewer changes to the existing TCG architecture. Thanks are due to Nikita Borisov for suggesting changes to TPM key management as an alternative to owner override.

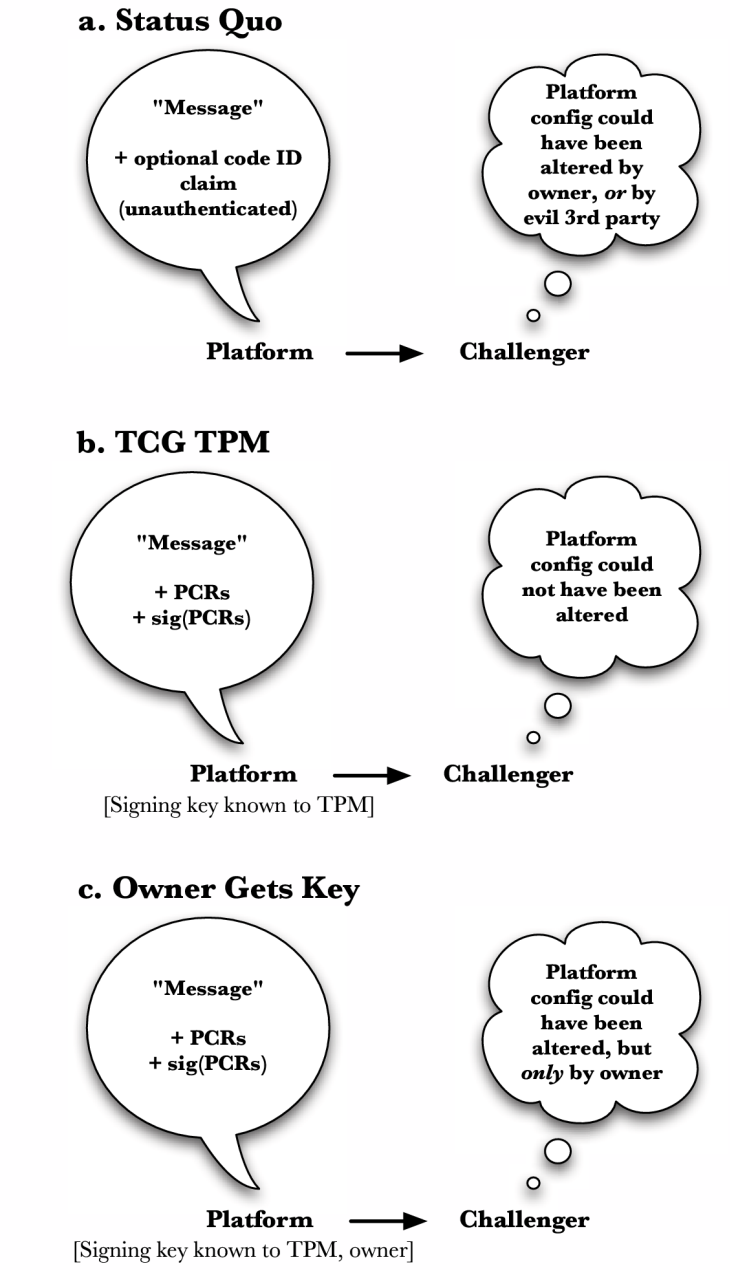


Figure 2: Possible meanings of attestation and inferences from them

6.3 Trusted third parties in TPM key management

TCG member companies have suggested that it might sometimes be desirable for TPM owners to have policies enforced against them. They offer the example of enterprise use of digital rights management technology interorganizationally to protect trade secrets or private information in the context of a negotiated contract or subsidiary relationship. They have noted that proposals such as owner override and owner gets key would undermine these applications because they would help one enterprise defeat the commitments it had made to another enterprise and thus lower the value of those commitments.

These considerations suggest an interesting question: is it possible to conceive of a redesigned TPM that helps organizations that do not trust each other to trust one another's devices, without making it easy for the same kind of feature to be used to enforce policies against the general public? One approach to this problem is to try to develop a TPM that allows only specified parties – not the entire world – to verify the endorsement on a TPM. Organizations that wanted to create trust relationships with other organizations' TPMs would have to take some specific action in the off-line world in order to establish those relationships. The trust thus created would ideally be specific to those organizations and would not generalize to allow other organizations to trust the same devices.[26]

We can thus imagine requiring a special effort and explicit out-of-band activity to acquire trust in a device.[27] We believe that this could be done using a trusted third party model in which TPMs allow independent parties – not their manufacturers – to create certificates that allow them to be used to establish trust *within a particular domain*.

We have not yet developed a specific mechanism for involving trusted third parties in the acquisition of trust. This proposal is different from the "privacy CAs" or existing trusted third parties dating back to a TCPA proposal because privacy CAs allow anyone to provide an attestation (while concealing that person's identity), where the third parties we propose allow only specified, registered parties to provide attestations to other specified, registered parties (without concealing anyone's identity).[28]

[26] For example, a company might want to let its parent company establish trust in its PCs, but the process of enabling this should not be directly transferrable to allow a mass-market software supplier to do the same.

[27] Our earlier discussions with TCG members suggest that some people will view this suggestion – and the others we present here – as inherently absurd because they make establishing trust in devices more cumbersome and expensive relative to the simplicity of the existing TCG attestation design. The difference between their view and ours may be that they some of them see the ability to establish trust in a device as an unequivocal good, whereas we see it as an equivocal good. On the view that establishing third-party trust in devices is sometimes beneficial and sometimes quite harmful, the engineering requirements for an attestation feature are a good deal more complicated than they would be if it were desirable to let anyone in the world readily trust anyone else's devices. In considering a proposal such as the "slow attestations" scheme discussed below, trusted computing vendors should consider more specifically which TPM applications they believe ought to be preserved and which might properly be eliminated.

[28] To summarize our view about anonymous attestation, we believe that anonymity is ex-

6.4 Slow attestations

Recently, many people have proposed proof-of-work or proof-of-effort schemes to limit spam and other unwanted communications.[29] These schemes rely on the insight that it might be possible to change the economics of an on-line interaction to encourage some uses of a resource while discouraging others. For example, it might be appropriate for me to send a hundred messages a day to people who may not know me, but it is probably not appropriate for me to send a hundred thousand such messages. A scheme that required me to apply a small amount of human or computational effort might make it practical for me to write to a smaller number of recipients but prohibitively expensive to try to write to all Internet users at once.[30]

It may be possible to add a similar proof-of-effort requirement to attestations, to retain the ability to use them among small groups but make it impractical to obtain attestations on a large scale from the general public. This might, for example, preserve intraorganizational use of attestation while making it harder to demand attestations in services provided to the general public. Instead of requiring an attestation verifier (or "challenger," in the parlance of many TCG documents) to submit a token, however, it would be more practical to make TPMs generate attestations with signatures that are hard to verify. Then would-be attestation verifiers would need to perform a significant amount of computation. The extra computation verifiers would have to perform could mean that attestation would be economical on a small scale, including in occasional interorganizational transactions, but not on a large scale. (This would seem to require preventing other properties of a TPM or computer from being used as a reliable proxy for code identity, which may or may not prove possible within the rest of the existing TPM architecture.)

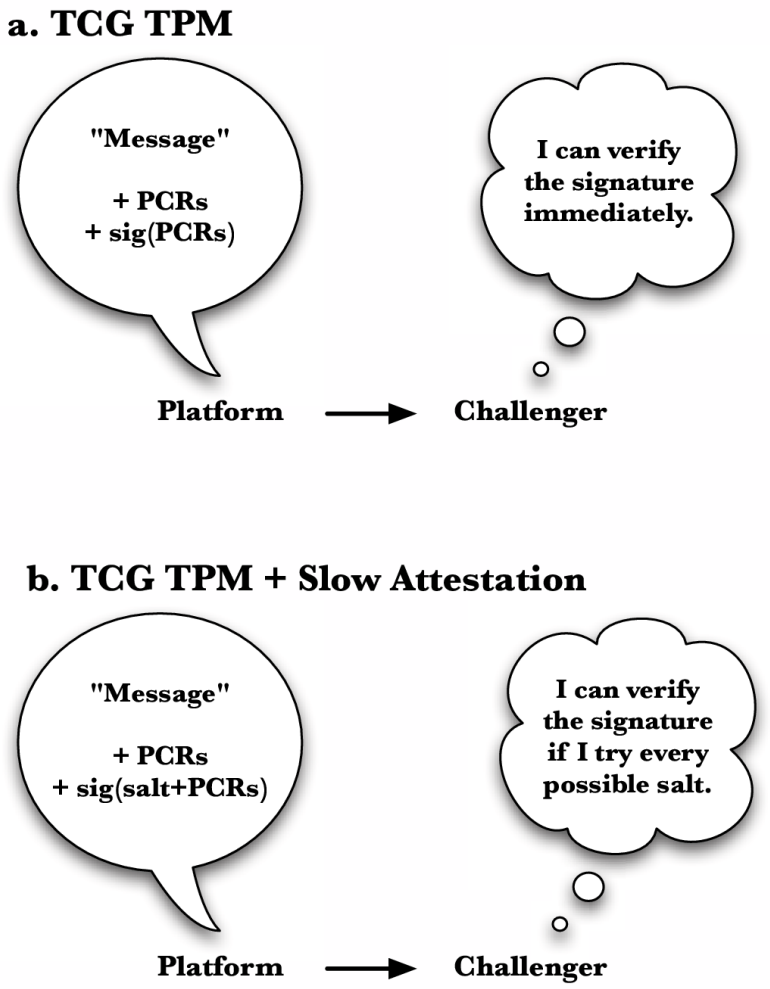
For example, as Figure 3 shows, a random salt could be generated and prepended to a list of PCR values before the TPM signs it; the signature and the PCR values, but not the salt value itself, could be transmitted to the verifier. The verifier would then need to undertake a brute-force search in order to verify the attestation. However, this approach is especially vulnerable to parallelization attacks. An improvement to this approach uses applications of

tremely valuable to the public – and hence that anonymous attestation is a strict improvement over attestation that provides personally-identifiable information. But because we do not believe that the harms of coerced attestation are solely harms to privacy, we suggest that attestations should not generally be demanded from the public in the first place. Therefore, even though anonymity is often desirable, the (rare) situations in which attestations are appropriate have relatively little overlap with the (frequent) situations in which anonymity are appropriate.

For the general public, anonymous attestation is a second-best solution; the first-best solution is the avoidance of coerced attestations.

[29] The best known examples of these may be Adam Back's Hashcash and Microsoft Research's Penny Black. See also Ari Juels and John Brainard, "Client Puzzles: A Cryptographic Defense Against Connection Depletion Attacks," in S. Kent, editor, *Proceedings of NDSS '99*, pp. 151-165 (1999).

[30] We should note that our remarks here should not be taken as endorsing any particular use of proofs of effort outside the context of trusted computing.



Using a more sophisticated approach in place of a random salt makes this approach more resistant to parallelization by the challenger.

Figure 3: Slow attestations

the the Rivest-Shamir-Wagner time-lock puzzle scheme, which attempts to solve the problem of requiring a pre-specified amount of real time to elapse before a message will be useful to a recipient. The Rivest-Shamir-Wagner scheme could be applied in several ways to TPM attestations to make them more fundamentally difficult to verify (essentially producing digital signatures that are easy to generate and arbitrarily hard to verify). These "slow attestations" would require a significant amount of computation on the part of the verifier in a way that would be difficult to speed up through parallelization. This would be a relatively small obstacle for entities that need to verify only a few attestations, but would make it inconvenient for a single entity to verify a large number of attestations.[31]

7 Conclusion

The Principles share a difficulty of terminology with the LT Technology Policy: both refer to an attestation feature as under the owner's control, or subject to the owner's control, when the owner has the ability to turn it on and off or to permit or prohibit its use on a case-by-case basis.

As we remarked in our comments on the LT Technology Policy, this sense of "control" is quite narrow. If we applied it to a computer as a whole, rather than to the TPM subsystem, it would mean that a computer owner "controls" a computer when the owner is empowered to switch it on or off.

We would similarly not say that an automotive hobbyist enjoyed full control over an automobile that could be turned on and off and driven anywhere, but that resisted aftermarket modifications. To us, the personal computer platform has been exciting, worthwhile, and valuable precisely because all its users could extend its functionality on an equal basis.

PCs with TPMs will be the first general purpose computing device sold to end-user consumers in the mass market with hardware features that can easily deter software reverse engineering and patching. In that sense, the TPM is an important change and an important loss of end-purchaser control, even though in principle that control may always be imperfect.

This problem of terminology has concrete implications. Deployment of TPM changes existing power relations and makes it easier for some people to enforce their security policies. TCG has not made clear which security policies it favors and which it disfavors.[32] To remedy this, TCG ought to begin its best-practices

[31] See Ronald Rivest, Adi Shamir, and David Wagner, "Time-Lock Puzzles and timed-release Crypto" (March 10, 1996), manuscript available at http://theory.lcs.mit.edu/~rivest/RivestShamirWagner-timelock.pdf; thanks are due to Hal Finney for the reference and for his helpful discussion of how to apply the Rivest-Shamir-Wagner scheme to this problem.

[32] It isn't enough to say that TCG is "neutral," since its work must proceed for some kind of threat model. Some things will necessarily be inside that threat model and others will be outside of it, regardless of the breadth of the policies that TPM applications might conceivably implement. Even generic security technologies – such as a PKI – have a notion of threat model and of the roles of the user and attacker.

efforts by publicly stating its threat model and the kinds of security applications it has set out to support. TCG as a whole should also undertake to speak in a more nuanced way about the meaning of "control." Finally, TCG should consider the possibility of technical refinements to future TPM specifications to deter coercive TPM applications and other applications that could shift the balance of power away from computer owners.[33]

[33]In doing so, TCG should continue to actively seek input from those whose interests are at risk, including reverse engineers, minority platform and application developers, and data migration and recovery specialists.