Privacy and Trusted Computing

Jason Reid, Juan M. Gonzlez Nieto, Ed Dawson, Eiji Okamoto
Information Security Research Centre
Queensland University of Technology
Brisbane, 4001, Australia.
{j.gonzaleznieto,jf.reid,e.dawson}@qut.edu.au

Abstract

This paper examines a model of trusted computing wherein a computing platform is able to make assertions about its current software configuration that may be *trusted* by the user and remote third parties. The privacy implications of this approach are investigated in the context of the Trusted Computing Platform Alliance (TCPA) specification. The trust relationships of the TCPA architecture are examined in detail. An analysis of the revocation requirements inherent in the TCPA design is presented, which highlights the challenges that revocation presents in the context of a large scale deployment of TCPA platforms. Finally, a modification to the specification is suggested that reduces the level of trust that must be placed on the Privacy CA.

Keywords: Trusted computing, TCPA, privacy.

1 Introduction

As the degree of dependence on networked computing devices continues to accelerate, there is a growing awareness of a substantial gulf that stands between the most basic, prudent standards of security and reliability for such critical systems and that which is actually delivered. Put more succinctly, our computers currently cannot be trusted to do their jobs properly. This situation has a number of unfortunate and undesirable consequences for safety and the continued growth and prosperity of an information based economy. These issues have been explored and analysed at length (for examples see [18, 2, 1]).

1.1 Trust in Open Computing Platforms

The openness and flexibility of the personal computer and popular commercial operating systems have been important factors supporting their widespread adoption. However, that very same openness and flexibility is proving to be a double edged sword, since it reduces trustworthiness. We use the word *trust* in the same sense as the definition in [13] - "A trusted component, operation, or process is one whose behaviour is predictable under almost any operating condition and which is highly resistant to subversion by application software, viruses, and a given level of physical interference." By openness and flexibility we mean the capability of the computing platform to execute arbitrary software. This may include malicious code that has been designed to access the resources of other programs.

Networked PC's are a vital part of modern commercial infrastructure. They are used as a low-cost communication and delivery channel for electronic commerce applications such as on-line banking and the sale of goods and services. However, commercial systems need to engender well founded trust between the participants and in the transaction process in order to reduce opportunities for fraud. In the on-line banking scenario, both the bank and the customer need to be able to authenticate each other. This is typically achieved via a combination of public key cryptography using TLS for server authentication [11], and passwords for client authentication. Both authentication methods require the client's computing platform to be trustworthy if the overall system is to be trustworthy. For reliable server authentication the public key of the bank's certificate authority must be securely stored on the client platform in a way that can resist tampering [12]. If this is not the case, an attacker can masquerade as the bank and capture the client's password. The code that implements the public key authentication protocol on the client platform must also be resistant to tampering. For password authentication, there must be a *trusted input path* that is not susceptible to snooping by rogue software.

Currently, popular PC operating systems fail on both counts. They are unable to provide a trusted path for input of sensitive information and they are unable to ensure the integrity of stored security critical information such as cryptographic keys [15]. More generally, they are unable to ensure the separation of mutually distrustful applications executing on the same device. This is a direct consequence of design decisions that have favoured flexibility, extensibility and ease of use over reliability and trustworthy operation.

There are two aspects of trustworthiness that are desirable for networked computing platforms. Firstly, the platform owner and user should be able to trust the configuration of the platform, e.g., that it is not running mali-

cious or unauthorised software that could compromise sensitive information. This requires a combination of operating system and hardware features that ensure reliable process separation and careful observance of the principle of least privilege [15]. Secondly, a platform should be able to attest information about its current configuration to another platform in a manner that the second platform can trust [8]. The second aspect allows an entity to authenticate the software configuration of a platform that is not under its control. In the case of open computing devices, this is necessary if a remote third party wishes to stipulate a policy or conditions attached to the disclosure of digital information and have some reasonable assurance that the policy conditions will be enforced. Policy contingent information disclosure is useful in many different contexts. A commonly cited and somewhat controversial example is Digital Rights Management (DRM). DRM seeks to allow an owner of copyright protected works (electronic content) to control how their content is used and transfered via devices that are not under their control. Another example is protecting the confidentiality of sensitive information such as medical records when they are released to third parties.

The first aspect of trustworthiness reflects principles of computer security that have been understood for over three decades [3, 17, 23], though they have been given little priority outside of military systems developed in the 1970's and 80's. The second aspect, remote attestation, is a more recent concept. It is an important feature of the Trusted Computing Platform Alliance (TCPA) Specification [21, 20] which is discussed in the following sections of this paper.

1.2 Overview of the Paper

Section 2 reviews the objectives of the TCPA architecture, and describes the operation of key functions including integrity protected booting, protected storage, sealed storage and remote attestation. We also make a number of observations on the extent to which TCPA realises the goals of trusted computing. Section 3 examines TCPA's credential system and privacy protection model in detail. Deployment challenges inherent in the PKI design that the TCPA architecture requires are discussed, particularly with regard to key revocation. Section 4 highlights a number of problematic design choices in the TCPA specification and suggests simple improvements that possess superior characteristics in terms of robustness and requiring reduced levels of trust in the Privacy CA.

2 Trusted Computing Platform Alliance (TCPA)

The TCPA was formed in 1999 by Compaq, Hewlett-Packard, IBM, Intel and Microsoft. It aims to "drive and implement TCPA specifications for an enhanced hardware and operating system based trusted computing platform that implements trust into client, server, networking, and communication platforms"[19]. The TCPA architecture has been designed with a range of devices in mind including PCs, laptops, servers, PDAs and mobile phones. The TCPA is a significant initiative in the development of networked computing devices, particularly because of its broad support from dominant industry players.

2.1 Objectives of the TCPA Specification

The key objective of the TCPA specification is to improve the trustworthiness and security of computing platforms. The novelty of the architecture lies in the range of entities that are able to use TCPA features as a basis for trust. These include not only the platform user and owner, but remote entities wishing to interact with the platform. The mechanism of *remote attestation* allows remote third parties to challenge the platform to report details of its current software state. On the basis of the attestation, third parties can decide whether they consider the platform to be trustworthy.

A closely related objective is to provide reliable, hardware based protection for secrets such as passwords and cryptographic keys. Since open computing platforms can run arbitrary software, this objective aims to ensure that protected secrets will not be revealed unless the platform's software state meets clearly defined and accurately measurable criteria. This objective aims to provide protection against malicious code, a critical aspect of engendering trust in open platforms.

2.2 How TCPA Trusted Computing Works

This section explains the TCPA specification in the context of a PC implementation. Details will vary for different types of platforms such as mobile phones and PDAs though the basic concepts remain the same.

2.2.1 Architectural Modifications

The architectural modifications required by the TCPA specification include the addition of a cryptographic processor chip to the motherboard, called a Trusted Platform Module (TPM). The TPM must be a fixed part of the

device that cannot (easily) be transferred to another platform. The TPM provides a range of cryptographic primitives including random number generation, SHA-1 hashing, HMAC-SHA-1, asymmetric encryption and decryption, signing and verification using 2048 bit RSA, and asymmetric key pair generation. There is also a a small amount of protected storage for keys. There is no support for symmetric cryptography. Currently available TPMs are based on smart card processors.

Under the current TCPA Protection Profile [22] the TPM is required to be tamper evident as opposed to tamper resistant. It is only required to provide adequate protection against a "casual breach of security by attackers possessing a low attack potential". The Protection Profile does not require resistance against power analysis [14], a powerful class of non-invasive attacks that can recover protected cryptographic keys by analysing the processor's power consumption. Power analysis attacks do not necessarily leave any signs of tampering. In section 3.2 we discuss the implications of this for TCPA's key revocation requirements.

2.2.2 Integrity Measurement and Reporting

TCPA security services build on an integrity protected boot sequence, a patented [7] technique that was introduced by Arbaugh [6, 5]. Integrity protected booting is fundamental to the design of the TCPA architecture. Figure 1 illustrates the process with numbers in parentheses denoting the sequence of events. The boot process starts in defined state with execution of the BIOS boot block code. The BIOS boot block is called the Core Root of Trust for Measurement (CRTM). Since it starts the booting and measurement process, it is implicitly trusted. The core idea in integrity protected booting is that executable code and associated configuration data should be measured *before* it is executed. Accordingly, the CRTM takes a hash of the BIOS code (1) and stores the value in the TPM (2). The measurement is stored in a Platform Configuration Register (PCR) in the TPM. PCRs cannot be deleted or overwritten within a boot cycle. They are *update only* using a simple chained hash technique that works as follows (where $\|$ denotes concatenation):

$$\texttt{UpdatedPCRValue} = \mathit{Hash}(\texttt{PreviousPCRValue} \,\|\, \texttt{MetricToStore}),$$

The CRTM then passes control to the BIOS code (3) and the boot process continues following the same pattern. If any executable stage in this chain has been modified, the change will be reflected in the hash value. Since the PCRs can only be updated, the modification cannot be designed to hide itself when it is given control of the CPU.

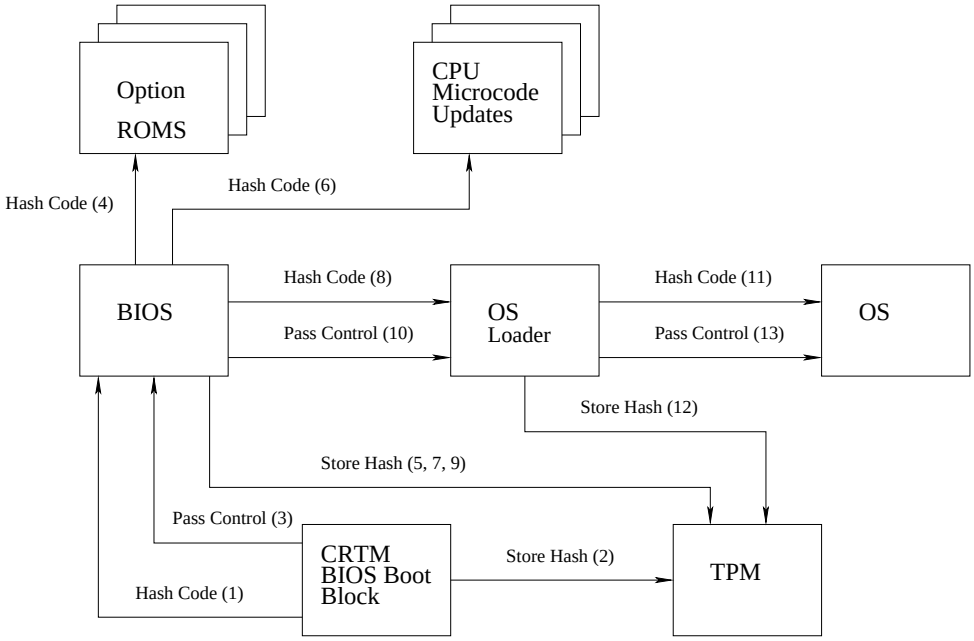

Figure 1: TCPA Integrity Protected Boot Sequence.

TCPA supports two modes of booting, *authenticated boot* and *secure boot*. In the latter mode, the platform owner can define expected PCR values that are stored in special TPM registers. If a PCR value does not match the expected value for that stage of the boot process, the boot can be terminated. Authenticated boot does not check actual values against expected values.

2.2.3 TCPA, DRM and Software Licence Enforcement

The TCPA initiative has attracted a degree of controversy, particularly in regard to digital rights management and software licence enforcement applications. It is worth clarifying that the TCPA specification itself, does not allow a third party to control which operating system and application software a platform owner can run. Therefore, the architecture does not provide a mechanism for software licence enforcement where a platform boot can be terminated by a third party, (perhaps the software licensor) if usage conditions are not met, (e.g. the licence has expired). However, it is possible to

implement an operating system or application that *uses* TCPA features to terminate OS booting or application loading on the instruction of a third party. Similarly, TCPA features can be *used* by applications to implement highly restrictive DRM regimes, capable of censorship and the erosion of fair use rights, as has been pointed out by Anderson [4]. The requirements of DRM applications have been considered in the TCPA architecture design [9] but it is up to the implementers of operating systems and application software to choose (or not) to build systems with these features. The TCPA specification does not provide them.

2.2.4 TCPA and Operating System Security

TCPA assumes a system can be trusted if the PCR registers match values expected by a relying party. The expected values must be those of a known secure configuration. This assumes that a secure and trustworthy configuration actually exists. As we discussed in Section 1.1, current commercial operating systems are not trustworthy due to poor process separation, insecure memory management, non-observance of the principle of least privilege, and lack of a trusted path for input and output [15].

To illustrate this point, consider that a *plug'n play* capability requires the dynamic loading of device drivers that can execute in kernel mode. Therefore, in this architecture, device drivers must be trusted since they can have unrestricted access to memory. This is not compatible with reliable separation of mutually distrustful code as a device driver that an application does not trust may load while that application is running. A rogue device driver could compromise secrets such as cryptographic keys or sensitive data in the application's memory space.

TCPA will not make such operating systems secure. It will merely allow reliable identification of an insecure configuration. TCPA does not correct flawed (from a security perspective), operating system designs or solve code quality problems.

2.2.5 Protected Storage

The TPM can be used to protect cryptographic keys from compromise by malicious software. TCPA provides greatly enhanced protection for signing keys without requiring any modifications to current operating systems. This is because the TPM can generate *signing only* key pairs and perform all signing operations itself, not releasing the key. The TPM does not do bulk symmetric encryption. Rather, it stores symmetric encryption keys, releasing them to the operating system environment when the required authentication

information is provided. Once released, the keys are reliant on the protection of the operating system.

The TPM directly stores only a single asymmetric storage key. This key is used to encrypt immediate child nodes in a storage hierarchy. These immediate child nodes can encrypt keys of further descendant nodes allowing the storage hierarchy to be extended without limit. The TPM must perform an asymmetric decryption to traverse each node of the hierarchy so the access time for protected objects will grow as the depth in the tree grows.

2.2.6 Sealed Storage

Sealed storage allows the release of a protected key to be conditional on the current status of PCR registers. Access to protected objects can therefore be conditional on the software state. The required PCR values can be defined in two ways. Firstly, an object can be tied to the PCR values current at the time the object was sealed. Secondly, a third party can define the required PCR values, which allows them to stipulate the necessary software environment for the key to be released. In the context of a DRM application, this would allow them to send encrypted content with the knowledge that it will not be accessible unless the platform is configured according to their wishes.

3 TCPA Remote Attestation and Privacy Protection Model

The TPM has a key pair called the *endorsement key pair* that is generated within the TPM at the time of manufacture. This key pair cannot be changed or erased and the private key is never released to the outside by the TPM. A so-called *TPM Entity (TPME)*, normally the manufacturer, provides a certificate of the endorsement public key called the *endorsement credential*. The endorsement key pair is unique to the TPM and, hence, its use in transactions with other parties would provide a means of unambiguously identifying the TPM. In order to protect the privacy of the trusted platform, the TCPA specification defines a pseudonymous identity credential scheme in which the endorsement credential is used by the TPM to obtain multiple *identity credentials* from *Privacy Certification Authorities (CAs)*. The endorsement key pair is only used in the identity credential request protocol. It cannot be used for general transactions. An identity credential is a certificate by a Privacy CA on an identity public key generated by the TPM. The privacy afforded by the scheme relies on the trusted mediation of the Privacy CA who knows the

binding between the platform identifiers (the endorsement credential) and the issued (pseudonymous) identities.

Remote attestation allows a TCPA platform to prove the state of its current software environment and its status as a genuine TCPA platform to a third party. An identity key pair is used to sign current PCR values. Figure 2 illustrates the attestation procedure. A TCPA platform seeking service from a provider is challenged by the provider to attest on its current configuration. The provider avails the service once satisfied that the TCPA platform is genuine, and that the current software environment is a trusted one. Different identity keys can be used by the TPM in different remote attestations to protect its anonymity.

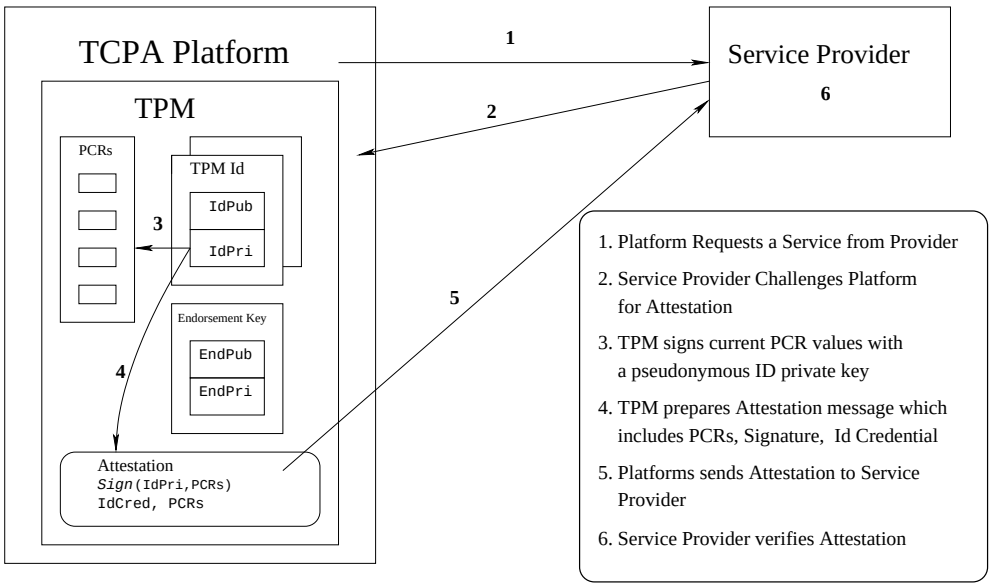

Figure 2: TCPA Platform Attestation to a Remote Entity.

3.1 Identity Credentials

In order to obtain an identity credential from a Privacy CA, a TCPA platform must show the following three different credentials:

1. A TPM endorsement credential signed by a TPME that attests that the identified TPM is genuine;

2. A *platform credential* signed by a *Platform Entity (PE)* (e.g. the platform manufacturer) that vouches that the TPM identified in the endorsement credential has been integrated into a platform that conforms to design;

3. A *conformance credential* signed by a *Conformance Entity (CE)* attesting that the TPM and platform designs conform with the TCPA specification.

Figure 3 shows a simplified version of the identity credential issuing protocol run between the trusted platform (TP) and the Privacy CA (PCA). Table 1 explains the notation used in the protocol description. Firstly, the TPM generates a new identity key pair (`IdPub`, `IdPri`). The public identity key `IdPub` is then sent to the Privacy CA together with the endorsement, platform and conformace credentials, denoted by `EndCred`, `PlaCred` and `ConCred`, respectively. In order to bind the request to the identity key pair, the TPM uses `IdPri` to generate a signature on `BindData`, which includes a hash of the Privacy CA's public key and `IdPri`. The signature is attached to the request. On receipt of the request, the Privacy CA verifies the submitted credentials and the signature. If the verification is succesful, the Privacy CA proceeds to create the identity credential, essentially a certificate on `IdPub` signed by the Privacy CA. The identity credential is then sent to the TP encrypted under the endorsement public key `EndPub` of the TPM. Encryption of the credential ensures that only the TPM identified in the identity request can succesfully obtain the credential. Further protection is achieved by a mechanism in the TPM that only allows the decryption of identity credentials whose request originated in the TPM itself.

`EndPub, EndPri`	endorsement key pair
`IdPub, IdPri`	attestation identity key pair
`EndCred`	endorsement credential
`PlaCred`	platform credential
`ConCred`	conformance credential
`IdCred`	identity credential
$Enc(\mathtt{e}, \mathtt{m})$	asymmetric encryption of message `m` using key `e`
$Sign(\mathtt{s}, \mathtt{m})$	signature on message `m` using key `s`

Table 1: Protocol Notation

1. $TP \rightarrow PCA$: IdPub, EndCred, PlaCred, ConCred, $Sign$(IdPri, BindData)
2. $TP \leftarrow PCA$: Enc(EndPub, IdCred)

Figure 3: Identity Credential Issuing Protocol

3.2 Credential Revocation

As discussed in Sect. 2.2.1, the current TPM Protection Profile does not require a TPM to be tamper resistant, only tamper evident. Therefore, attacks that result in the compromise of the endorsement secret key (or any other key) should be expected to occur frequently, when trusted platforms are used, for example, in high value transactions or DRM applications. Recovery of this key allows an attacker to create a virtual trusted platform that is entirely under their control. Publication of a valid endorsement key pair would allow widespread impersonation of the trusted platform, without the trust.

The TCPA specification acknowledges that "the trustworthiness of the architecture is vulnerable to the compromise of a single TPM endorsement private key"; however, no provision for credential revocation is included. It is clearly important that endorsement credentials can be revoked for TCPA to realise its full potential. Privacy CAs need to confirm that an endorsement credential has not been revoked before they issue an identity credential based on it. Similarly, service providers may need to confirm that a pseudonymous identity has not been revoked before they rely on it.

The specification also claims that "certain forms of revocation scheme can be retrofitted, should it become necessary at some time in the future." We notice, however, that the current specification severely hinders the deployment of any mechanism capable of addressing the revocation of credentials in an efficient and effective manner. To see this, we firstly must appreciate the complexity inherent to certificate revocation, a task that is proving itself to be a major impediment in the successful deployment of a global public key infrastructure (PKI) [16]. We should notice that the intended scope of TCPA is also global. Scenarios in which credentials may need to be revoked include the following.

1. A TPM endorsement private key is compromised. Then, the endorsement credential and any associated identities credentials have to be revoked.

2. An identity private key is compromised. Then, depending on the revo-

cation policies being implemented by the different CAs, different scenarios are possible, including the following:

(a) The corresponding TPM endorsement credential is also deemed compromised and revocation is effected as in scenario 1;

(b) All associated identity credentials within the same issuing Privacy CA need to be revoked.

(c) Only the compromised identity credential requires revocation.

These scenarios represent a subset of the potential situations that require revocation of credentials. Other circumstances that would result in revocation are the compromise of the signing keys of entities such as manufacturers or CAs.

The propagation of revocation amongst associated credentials adds complexity to the revocation mechanism when compared with traditional PKIs. It not only increases the amount of certificates that need to be revoked within the infrastructure, but also demands extra functionality to allow credentials that are associated to be traceable. The above revocation scenarios show that there are situations in which it may be required to find the endorsement credential behind a given identity credential, as well as to find all the identity credentials within the domain of a Privacy CA that are associated to a given identity credential.

Informally, we can define the security requirements needed to support traceability of credentials as follows.

- *Revocable Anonymity*: It should not be possible for anyone (except for a designated Privacy CA) to link an identity credential to the associated endorsement credential.

- *Revocable Unlinkability*: It should not be possible for anyone (except for a designated Privacy CA) to link any two associated identity credentials.

We say that the credential scheme provides *credential traceability* if it satisfies the above two properties. Clearly, the capability to revoke anonymity implies the capability to revoke unlinkability. The Privacy CA acts as a revocation authority that can be called upon in special circunstances, e.g. as part of the credential revocation process, to reveal the binding between credentials.

We can further qualify the (anonymity and unlinkability) revocation process as *strong* or *weak* depending on whether the Privacy CA can provide cryptographic proof of the link between credentials. The current TCPA architecture provides weak traceability, i.e. it allows Privacy CAs to revoke the

anonymity (and hence the unlinkability) of identity credentials, but no proof can be produced by the Privacy CA to demonstrate the link between them. The Privacy CA is trusted to claim only genunine mappings between issued identity credentials and TPMs. This trust, if violated, allows the Privacy CA to frame a TPM by asserting an incorrect mapping. If an identity key was used for misbehaviour, a service provider could request revocation of the associated TPM endorsemnet credential. If the evidence of misbehaviour were sufficient, the Privacy CA could claim an incorrect mapping resulting in revocation of an 'innocent' TPM. In Sect. 4 we show that strong traceability can be obtained at very little extra-cost.

Bearing in mind the intricacies of credential revocation, it is not difficult to resolve that the decision by the TCPA to only require tamper evidence for TPM chips complicates the retrofitting of any revocation mechanimsm; for it considerably increases the number of credential revocations due to key compromise that the mechanism must deal with. Similarly, the lack of strong credential traceability predetermines a revocation scheme which places unnecessary trust on Privacy CAs, thus limitting the scheme's robustness.

4 Minimising the Trust on the Privacy CA

Credential authenticity is the most basic security requirement for any credential system. It should not be possible for any user to generate a credential without the approval of the corresponding trust provider. In the TCPA architecture, the trust providers for the endorsement, platform and conformance credentials are the TPME, the PE and the CE, respectively. The possesion of these certificates by a TCPA platform is what enables service providers to trust the attestation process. As explained in Sect. 3.1, posession of the credentials is not proved directly to the service provider, but indirectly through the Privacy CA. Hence, service providers must trust that the Privacy CA will not issue identities to non-genuine TPMs.

Since the reason for including Privacy CAs in the TCPA architecture is the provision of privacy to platforms, it appears that the additional reliance on the Privacy CA to check the authenticity of the credentials is an undesireable side effect of the design. Therefore an alternative credential scheme without the extra trust on the Privacy CA would be preferable. This is in line with the general security design principle of minimising the trust vested on third parties, which results in more robust schemes and for which cryptographic techniques are mainly employed. Anonymous credential schemes are an active area of research. Recent proposals such as those by Camenisch and Lysyanskaya [10] and Verheul [24] deserve further investigation as to their

suitability to the TCPA environment.

As pointed out in Section 3.2, Privacy CAs cannot prove that an identity credential that they have issued was actually requested by the TPM with the matching endorsement credential. As a consequence, Privacy CAs cannot be made accountable for the fabrication of illegitimate identity credentials. This can be easliy fixed by modifying the identity credential issuing protocol so that the requesting TPM generates a signature using the endorsement private key to bind the requested identity credential to the endorsement credential. For example, the signature from the original protocol (Figure 3) could be replaced by

$$Sign(\texttt{EndPri}, \texttt{EndPub} \parallel Sign(\texttt{IdPri}, \texttt{BindData})),$$

The Privacy CA would now have to verify both signatures and store them in case it needs to show proof of the binding between the identity and endorsement credentials. The modified protocol is shown in Figure 4.

1. $TP \rightarrow PCA$:	$\texttt{IdPub}$, $\texttt{EndCred}$, $\texttt{PlaCred}$, $\texttt{ConCred}$, $Sign(\texttt{EndPri}, \texttt{EndPub} \parallel Sign(\texttt{IdPri}, \texttt{BindData}))$
2. $TP \leftarrow PCA$:	$\texttt{IdCred}$

Figure 4: Modified Identity Credential Issuing Protocol

5 Conclusions

We have reviewed the motivations that are driving a renewed interest in trusted computing, an area that was studied extensively in the 1970s and 80s but has received very little attention until the recent surge of awareness created by the release of the TCPA specification.

We have described the objectives of the TCPA architecture and reviewed key aspects of its functionality. This has included observations on the extent to which TCPA delivers important trusted computing features. While the TCPA specification is an important step toward the goal of trusted computing, it only provides part of the solution. The improved protection for signing keys and reliable platform authentication features are of immediate benefit. However, TCPA does not address more fundamental requirements for trust that can only be delivered by an appropriately designed operating system.

We have reviewed TCPA's credential system and privacy protection model in which the Privacy CA plays a critical role. We presented a detailed analysis of the revocation requirements inherent in the TCPA design, highlighting

the practical challenge that revocation presents in this context. We explored ways of reducing the trust that must be placed on the Privacy CA, and provided concrete suggestions to achieve this goal.

References

[1] Critical infrastructure protection - significant challenges in developing national capabilities, April 2001. Report to the Subcommittee on Technology, Terrorism, and Government Information, Committee on the Judiciary, U.S. Senate.

[2] *Critical Infrastructure Assurance: Information and Communications*. U.S. Department of Commerce National Telecommunications and Information Administration (NTIA), 2001-11-13.

[3] J. Anderson. Computer security technology planning study. Technical Report ESD-TR-73-51, AFSC, Hanscom AFB, Bedford, MA, October 1972. AD-758 206, ESD/AFSC.

[4] Ross Anderson. TCPApalladium frequently asked questions. http://www.cl.cam.ac.uk/users/rja14/tcpa-faq.html accessed 13 March 2003.

[5] W A Arbaugh. *Chaining Layered Integrity Checks*. PhD thesis, University of Pennsylvania, 1999.

[6] W. A Arbaugh, D J Farber, and J. M Smith. A secure and reliable bootstrap architecture. In *Proceedings 1997 IEEE Symposium on Security and Privacy*, pages 65–71, May 1997.

[7] W. A. Arbaugh, D J. Farber, J M. Smith, and Angelos Keromytis. A secure bootstrap process, US patent 6,185,678, February 2001.

[8] B Balacheff, D. Chan, L. Chen, S. Pearson, and G. Proudler. How can you trust a computing platform? In *Proceedings of Information Security Solutions Europe (ISSE 2000)*, Barcelona, Spain, September 2000.

[9] B Balacheff, L. Chen, S. Pearson, D. Plaquin, and G. Proudler. *Trusted Computing Platforms - TCPA Technology in Context*. Prentice Hall, 2003.

[10] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. Report 2001/019, Cryptology ePrint Archive, March 2001.

[11] T. Dierks and C. Allen. RFC 2246: The TLS protocol version 1, January 1999. Status: PROPOSED STANDARD.

[12] Carl Ellison and Bruce Schneier. Ten risks of pki: What you're not being told about public key infrastructure. *Computer Security Journal*, 16(1):1–7, 2000.

[13] ISO/IEC. Information technology - Open Systems Interconnection - Evaluation criteria for information technology, 1999. Standard ISO/IEC 15408.

[14] Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Michael Wiener, editor, *Advances in Cryptology – CRYPTO '99*, number 1666 in Lecture Notes in Computer Science, pages 399–397. International Association for Cryptologic Research, Springer-Verlag, Berlin Germany, August 1999.

[15] Peter A. Loscocco, Stephen D. Smalley, Patrick A. Muckelbauer, Ruth C. Taylor, S. Jeff Turner, and John F. Farrell. The inevitability of failure: The flawed assumption of security in modern computing environments. In *21st National Information Systems Security Conference*, pages 303–314, Arlington, VA, October 1998.

[16] R. Rivest. Can we eliminate certificate revocations lists? In *FC: International Conference on Financial Cryptography*. LNCS, Springer-Verlag, 1998.

[17] Jerome H. Saltzer and Michael D. Schroeder. The protection of information in computer systems. *Proceedings of the IEEE*, 63(9):1278–1308, September 1975.

[18] Fred Schneider, editor. *Trust in Cyberspace*. National Academy Press, 1998. Committee on Information Systems Trustworthiness, Computer Science and Telecommunications Board, Commission on Physical Sciences, Mathematics, and Applications, National Research Council.

[19] Trusted Computing Platform Alliance TCPA. http://www.trustedcomputing.org.

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

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

[22] Trusted Computing Platform Alliance (TCPA). Trusted platform module protection profile, July 2002. Version 1.9.7.

[23] U.S. Department of Defense. Trusted computer systems evaluation criteria, August 1983.

[24] Eric Verheul. Self-blindable credential certificates from the weil pairing. In *ASIACRYPT: Advances in Cryptology – ASIACRYPT: International Conference on the Theory and Application of Cryptology*. LNCS, Springer-Verlag, 2001.