Trusted Computing and the Trusted Platform Module:
What All the Fuss Is About

Bill Hewitt

Due 4/13/06

CS182 JT

There has been a great deal of talk about computer security in recent years as more and more systems gain access to the Internet. Worms, viruses, trojans, spyware, and other kinds of malware that seem to work their way through many lines of defense can make using and maintaining a connected computer quite a chore. Due to this, a variety of companies, consortia, and standards bodies exist with the expressed purpose of developing solutions to make our computing experience safer and more confidential. Trusted computing, one of the more contentious realms within this broad field, received a lot of negative attention with the announcement of Microsoft's Palladium initiative (now called the Next Generation Secure Computing Base, or NGSCB)[1] a few years ago. The Trusted Platform Module (henceforth TPM), a hardware chip developed by the Trusted Computing Group (henceforth TCG), provides many of the security and confidentiality features essential to the initiative. Because of its association with the extremely controversial NGSCB, the TPM found itself the recipient of a lot of completely unjustified bad press. Due to its broad applicability, the TPM provides developers with a great deal of power and few restrictions on its use. Assigning responsibility to the TCG for what implementers can do with the TPM, however, is equivalent to blaming processor manufacturers for enabling computer viruses. Regardless of the media frenzy, the TPM enables the creation of extremely useful tools and provides no ill intent toward end users by itself.

The TCG formed in 2003 as a not-for-profit group of technology companies with the goal of developing an open, standardized security solution that would allow for secure operation of computing systems, privacy protection for end users, and easy interoperability of components.[2] Since debate persists as to the actual definition of "trusted computing", assume for the sake of this paper that it is embodied in this goal.

1 See http://www.microsoft.com/resources/ngscb/default.mspx for details
2 Trusted Computing Group. "Trusted Computing Group Backgrounder". https://www.trustedcomputinggroup.org/news/TCGBackgrounder_112105.pdf, 4/10/06

The TCG's membership contains a sampling of over 120 organizations from many disparate areas of the computing field, such as AMD, Intel, IBM, Microsoft, Sun, Dell, Sony, Dartmouth College, Lockheed Martin, Motorola, and Verisign.[3] The group itself is the successor to the Trusted Computing Platform Alliance (TCPA) founded by IBM in 1999 and retains much of the TCPA's former membership. Their main product, the TPM specification, defines a hardware chip that performs a variety of security-related functions.[4]

While you may have never heard of the TPM chip before, it is already in widespread use. Twenty million of these chips shipped in 2005, and up to 250 million are expected to be in use by 2010. There are several different implementations available from a variety of vendors. The TCG has already published the second edition of its specification, meaning that two generations of hardware chips exist. The most prominent chips are the models made by Infineon, the SLD 9630 TT (discontinued; based on revision 1.1 of the specification) and SLB 9635 (based on revision 1.2). Intel includes TPM chips integrated into many of its recent platform chipsets as well.

In practice, the TPM's main functions are platform monitoring, secure storage, encryption operations, and authentication services.[5] It provides public/private key pair generation, symmetric key generation, system integrity monitoring at boot and throughout the system's uptime, encrypted storage of user's keys, passwords, and certificates, several encryption algorithms, and secure authentication of the platform to which it is bound. In addition, the TCG design goals insist that the chip must be cost effective to deploy on a large scale, it must not hinder the legal exportability of its

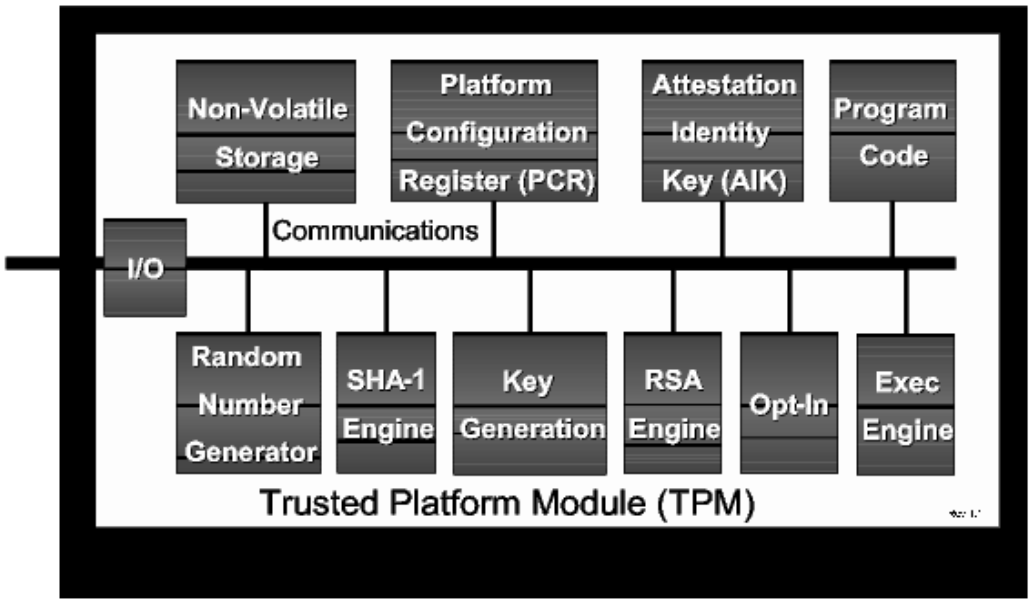
3 For a complete and current list, see https://www.trustedcomputinggroup.org/about/members/

4 When this paper refers to the TPM, assume that it is talking about the "ideal" TPM defined by the TCG's specification rather than any specific hardware implementation.

5 Infineon Technologies AG. "Infineon's Trusted Platform Module", http://www.silicon-trust.com/trends/comp_tpm.asp, 4/05/06

platform throughout the world market, and it must protect the user's privacy. Note that this paragraph does not mention the following uses that are commonly attributed to the TPM by the media: DRM, preventing the use of unlicensed software by checking a serial number database, and verification that the platform's hardware and software is "certified" by the TCG. In short, the TPM does not do any of these things.[6]

The TPM itself consists of several major components, as depicted below.[7] A brief description of each follows, with a more detailed explanation of each part available from the specification itself.[8]




The I/O Controller is a fairly simple component and has a fairly loose specification. It manages the TPM's interface and communication to the outside platform as well as controlling and routing internal signals. Additionally, it enforces all the access control required by the opt-in mechanism and other components.

6 For a more in-depth explanation, see: Safford, David. "Clarifying Misinformation on the TCPA". http://www.research.ibm.com/gsal/tcpa/tcpa_rebuttal.pdf, 4/10/05

7 Figure from https://www.trustedcomputinggroup.org/groups/TCG_1_0_Architecture_Overview.pdf

8 The TPM specification is still changing. This paper is based on version 1.2, revision 94. The newest revision of the document should be available online at https://www.trustedcomputinggroup.org/specs/TPM.

The Cryptographic Co-Processor (CCP), a major subdivision of the TPM, contains the various cryptographic engines used by the TPM as well as the Random Number Generator vital to these components. Its functionality must include at least RSA key generation, RSA encryption/decryption, SHA-1 hashing, and random number generation for the sake of interoperability. However, the specification explicitly allows other asymmetric algorithms such as DSA or elliptic curve. All storage keys and identity keys must at least match the strength of a 2048 bit RSA key, which should be enough to offer sufficient protection from malicious access. One important distinction to make is that the CCP is **not** a cryptographic accelerator, and no minimum throughput numbers appear at any point in the specification.

The RSA Engine must support 512, 768, 1024, and 2048 bit keys, with a minimum recommended size of generated keys being 2048 bits. The specification does not require any particular implementation of RSA, which allows the implementer some flexibility. It does specify a public exponent of $2^{16} + 1$. All signing done within the TPM must use RSA encryption or risk the signature not being accepted by other TPM devices. The SHA-1 Engine provides the primary hash algorithm used by the TPM and requires 160-bit keys. The HMAC Engine, whose implementation is dictated in RFC 2104 and involves turning a keyless hash function into a keyed hash by incorporating a cryptographic key, allows the chip to detect proof of knowledge of AuthData (discussed later) and proof that incoming requests are authorized and have not been tampered with prior to arriving.

Symmetric encryption algorithms are also used by the TPM, but only internally, as they can not have user-accessible interfaces. These see use primarily in encrypting authentication exchanges and encrypting internal data that was fed into the TPM from outside sources. The TPM must use the Vernam one-time pad mechanism

with XOR. The specification explicitly allows AES as an alternate algorithm or any other algorithm that the implementer feels is sufficient.

The Key Generator creates both RSA key pairs and symmetric keys used for encryption. The Key Generator does not have a strict specification, except that it should not use data that has existed in a non-protected location as a key and all nonces need to be taken from the TPM's Random Number Generator (RNG).

The RNG itself consists of a state register, a collector of either entropy or unpredictable data such as thermal noise or clock offsets, and a post-processor with a hashing function. The state register is a protected location inside the TPM's non-volatile memory that stores the current state of the machine. It can also be implemented as a combination of one volatile register and one non-volatile register, which is a bit of clever design allowing developers to use flash RAM (which wears out after a certain number of writes) as the non-volatile storage. The volatile register is simply written to the non-volatile register when the TPM detects a power-down. The entropy collector filters the input data to make sure there is no bias and makes an attempt to correct it if there is. This allows the TPM to produce good random numbers without needing a dedicated source of hardware entropy.

The Opt-In Component maintains the state of various flags, such as whether the TPM is enabled or disabled. An important part of this is tied to the fact that the platform operator must be physically present at the machine in order to change the state of these flags. The particular method of asserting physical presence is left up to the implementation, but an example of requiring local keyboard input (which can be verified by establishing a trusted path between the keyboard and the platform) is given.

The Execution Engine does just that: executes the function calls (stored in the

Program Code section of the chip) that the chip receives on its I/O bus. The EE makes sure the security and integrity of the chip and the data it protects is properly maintained. The TPM ships with both volatile and non-volatile RAM for storing secret data and computational variables.

Finally, the Platform Configuration Registers (PCR) are 160-bit storage locations for integrity measurements. There are at least 16 PCRs on the TPM. There are a large number of values to be measured and stored, and the result of the new measurement cannot overwrite the old measurement (or a malicious user could overwrite a value that indicated tampering with a known good value, subverting the detection mechanism). Thus, the TCG came up with a clever trick to deal with the fact that each measurement must be individually stored. As you may have noticed, the PCR contain the same number of bits as the output of the SHA-1 Engine. This is because the PCR holds a hash of all the previous updates, and when a new metric must be stored it just hashes the value of the new measurement concatenated onto the old measurement. This makes it a very difficult system to break into, as you would have to somehow reverse the hash computation (something that is currently infeasible) to determine the input message.

There are a few other crucial parts of the TPM's operation not represented in the diagram, mostly because they are not physical components. The Endorsement Key (EK) is a 2048-bit key pair, the public key being the PUBEK and the private key being the PRIVEK. The EK is actually generated by the manufacturer and put into the TPM prior to its placement in a platform, as it is used during validity testing for the TPM.

The PRIVEK serves as the main private key for the TPM. As such, its exposure outside of the TPM would invalidate the TPM's entire security capability. Thus, it remains shielded at all times. Any computation done with the PRIVEK must

be done inside the TPM. The PUBEK does not present a security concern. However, if it is associated with some kind of personal information such as a platform identifier (e.g. the EK or an AIK, which will be explained shortly), it can become personally identifiable information. This is a major privacy concern, so the association of PUBEK with personal information should be controllable by the user.

An Attestation Identity Key is a 2048-bit RSA key that aliases the EK. It is used for signing data that is generated internally to the TPM but may be available outside. The EK cannot be used for this due to security reasons and privacy concerns. A "virtually unlimited" number of AIK can be generated by the TPM.

The TPM performs both authorization[9] and authentication[10] functions. Each object withing the TPM that has any kind of access restriction has a 160-bit shared secret embedded within it. This shared secret is called the object's AuthData. If a subject can provide proof of knowledge of this AuthData, the subject is granted full access to that object. This means that to the TPM, AuthData is the sole way to authenticate whomever is trying to access its objects. However, to an application, the AuthData will most likely be used to authorize access to the TPM for functions such as OS login or file system access. Due to its sensitive nature, AuthData should never be available in the clear and should be closely monitored when outside the TPM.

Finally, version 1.2 of the TPM specification contains a stipulation that it must contain a mechanism to prevent dictionary attacks on its data. However, it does not give any further instruction about this aspect aside from an example. This example shows a simple authorization attempt counter that locks the chip for an increasing time period based on the number of failed attempts, with various contingencies based on the TPM being reset or unlocked by its owner.

---

9 Granting a subject appropriate access to an object
10 Providing proof of ownership of an object or identity

The mere existence of these components means little on its own. The important part of the TPM is what it enables users and developers to do. One of its major capabilities is sealed storage.[11] The principle behind sealed storage is the ability to cryptographically restrict access to a file to a specific set of subjects. It can also be done on either the hardware level or at several different depths of the software level (driver, OS, application). When a subject seals a piece of data, it uses a code ID (such as a hash of the code of the subject program) and the code ID of the other subject allowed to access the data. When using the TPM to verify that each program is who they say they are, this provides both confidentiality and data integrity assurance. The data is securely protected, and when an accessor unseals the data, it is provided with the ID of the subject that sealed the data. This ensures the accessor that the data came from the proper source. Another feature of TPM is the secured boot process. In this situation, the operating system kernel can verify itself at various stages in the boot process using the TPM. If the program's code ID doesn't change, this will ensure that the OS itself has not been tampered with or corrupted.

The most controversial feature of the TPM is attestation, which allows subjects to authenticate their code ID to a third party. This allows for such services as signed key propagation, secure certificate validation, and authenticated interprocess communication. An especially interesting application of this is network security. The network switch or router can make its clients attest, finding out if they are running compromised or insecure versions of their operating system software. If so, the switch could redirect them to a patch website or prevent them from accessing the network at all. The issue that had many people up in arms was the fact that in version 1.1 of the

11 England, Paul; Lampson, Butler; Manferdelli, John; Peinado, Marcus; Willman, Bryan. “A Trusted Open Platform”. *Computer*, July 2003

specification, attestation required that PUBEK had to be transferred to a trusted third party. While this does work, it is suboptimal due to the fact that PUBEK being in the wild can lead to personally identifiable information being uncovered about a subject (as discussed earlier). Thus, v1.2 of the specification includes a provision for something called Direct Anonymous Attestation. This is based on a cryptographic technique known as Zero Knowledge Proofs in which a subject can prove knowledge of a secret without actually exposing any information about the secret itself. This allows for completely anonymous attestation to third parties.

While the specification is certainly powerful and its security and privacy aspects continue to improve, there are a few minor issues remaining. For one, it is made very clear that the chip is not physically tamper-resistant. Thus, someone with an oscilloscope and a lot of time could probably derive much of the data stored on the chip (but probably not PRIVEK, which never leaves the chip). Also, someone with local access to the machine can reset the TPM or turn it off, making it unsuitable for workstations in public environments. This also means that while others can't access your private data if they steal your laptop, the TPM doesn't do anything to protect data outside of the TPM itself because it can be cleared easily. Finally, as a result of public pressure users are able to turn the TPM completely off. While this is certainly a useful feature, TPM platforms ship with it set off by default. This means that most users will never even know it exists, let alone turn it on. This could lead to an IPv6-like situation where everyone has the capability but nobody actually utilizes the technology.

As shown, the TPM itself is not a devious tool of the corporate juggernauts of technology to simultaneously report you to the police for stealing software, render your mp3s unplayable, and automatically disallow non-accredited hardware manufacturers from ever making functioning devices. It is just a chip that does

encryption and secure storage. The specification is open and shared among over one hundred companies. The technology actually has the ability to protect users from things like cracking, snooping, and malware. However, it does possess enough power to allow people who don't have the consumers' best interests in mind to do some devious things. Then again, so does the Internet...and just like the Internet, if you don't like the TPM, you can always turn it off.