TCPA and Palladium

Are the Benefits of the Hardware Changes Enough to Justify an Upgrade?

Omar Bakr, Hareesh Nair, Aman Narang, Tony Scelfo
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology, Cambridge, MA

December 11, 2002
MIT Course 6.805
Prof. Hal Abelson

ABSTRACT

TCPA and Palladium, two proposed hardware changes to the x86 architecture, hope to solve a number of current security issues on the open net. In this paper, we take a look at six hot security issues and determine what TCPA and Palladium add. If the hardware enhancement makes a significant step in solving a specific security concern, we consider the practical concerns of switching to a new hardware architecture and if the upgrade is worth the added security. We find that in most cases, there is not a significant increase in security to justify the process of upgrading to new hardware.

TABLE OF CONTENTS

INTRODUCTION

In 1999, HP, Compaq, Microsoft, IBM and Intel formed the Trusted Computing Platform Alliance (TCPA). This organization was formed with the express purpose of designing a new element for computer architecture. Specifically, TCPA is exploring the possibility of a "trusted platform", one that promises greater information security. The TCPA envisions that all new computers will be fitted with an encryption chip, one that would form a "root of trust" among the user, the hardware and software, and other users. Trusted computing intends to add several notable features to existing computer platforms, including the ability to authenticate and attest the user's platform configuration to concerned parties, and the ability to protect the platform secrets by using tamper-resistant chips. The new platform promises to protect users' data from malicious code (e.g. viruses, Trojans), to reduce the possibility of impersonation, and decrease the likelihood of denial of service attacks. In sum, the TCPA pledges to "increase consumer and business confidence".

Having a protected "root of trust" allows the computer to store an encrypted identify. Because the machine's identity can be securely stored on the encryption chip, software can be built on top of the hardware to allow for strong attestations. Attestations would allow truths to be generated about the serial numbers of the hardware, the software installed on a machine or even the software running. These truths can be reported to remote computers so that the remote computers can positively identify the machines that they are connecting to. Attestation is one of the strongest benefits of the TCPA and Palladium initiative. Corporate environments will want this new feature because it will allows them to positively identify all the machines on a network. The consumer market will find this new feature attractive because it allows for an operating system to be engineering such that stolen hardware can not be used on a system connected to the internet. Both industry and the consumer will find this new feature attractive, but for whatever reason, TCPA and Palladium are being marketed as a revolutionary architecture that will end many of the security problems that exist in computers today.

There is a definite threat of security breaches in the computing environment. According to the Computer Security and Crime Survey conducted by the Computer Security Institute (CSI) and the Federal Bureau of Investigation (FBI), 90 percent of respondents detected security breaches in the past 12 months. Of these, 80 percent acknowledged financial losses due to security breaches. Among the respondents, 45 percent (223 respondents) were willing to quantify their financial loses to a value of $456 million.[1] Figure 1 illustrates a breakdown of the financial loses due to security breaches.

[1] Power, Richard. 2002 CSI/FBI Computer Crime and Security Survey. Computer Security Institute. 2002

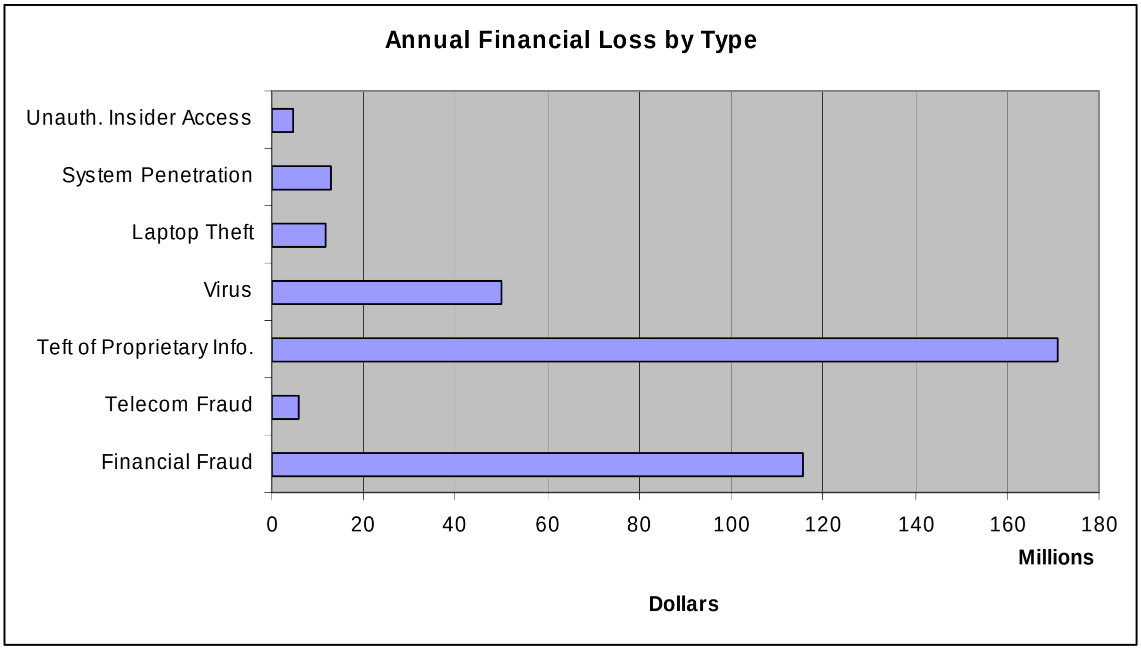

Figure 1

We look at several of the most significant threats that occur on a networked computer system. These threats include as malicious code, such as viruses and Trojan horses, data theft through both hardware and software attacks, and the identity verification problems that plague internet commerce and communication. We show that TCPA and Palladium serve only to strengthen several encryption techniques that are used in addressing these threats today. By simply strengthening today's techniques, TCPA and Palladium do not drastically improve the security of computers. With the exception of several very specific cases, such as a small application for conducting back transactions, TCPA and Palladium do not go far enough into securing computers to justify a change to the motherboard, operating system and application framework.

Most of the benefits of trusted computing fall under the category of key storage and curtained memory. Key storage can be solved under today's system architectures. An example of a secure key store that can be implemented today is a smart card that is used to store cryptographic keys. The card can then be removed from the computer, making the data that is encrypted on the computer safe from hardware and software attacks. Although this method is not air tight, it goes a long way to solve the problems of protected key storage. Curtained memory, on the other hand, can not be achieved without a hardware change. However, curtained memory only serves to protect the data of one application from a different application running on the same machine. Virus protection software can serve to prevent malicious applications from existing and running on a computer today.

TCPA and Palladium serve only to add a limited set of new features to business and consumer applications. The average user will not appreciably benefit from TCPA and Palladium, so there is no significant reason to spend the money to replace most business or home computers with a TCPA or Palladium system. Hardware attestation can significantly benefit a user when it comes to interconnectivity of devices. Devices such as palm pilots, printers and cell phones, to name a few, can have much stronger identification properties on a TCPA or Palladium system, where hardware attestation can be used. The significant benefits found in embedded devices are much greater than the few benefits caused by using TCPA or Palladium in the robust personal computers that exist today.

The rest of this paper is organized as follows: Section 2 explains what a trusted computing platform is. Section 3 discusses the threat of malicious code. Section 4 presents the threat of stolen hardware. Section 5 describes the threat of e-mail authentication. Section 6 discusses spam. Section 7 addresses the threat of spoofing.

Section 8 describes the threats involved in P2P resource sharing. Section 9 concludes the paper and suggests why TCPA and Palladium are not worth the upgrade.

1 TRUSTED COMPUTING

1.1 What is Trusted Computing?

Why create a trusted computing initiative like TCPA and Palladium?

1. Platform Authentication and Attestation:

Allow challenging parties to identify your platform and its properties.

2. Platform Integrity Reporting:

To reliably measure and report on the platform's software state.

3. Protected Storage:

For Hardware protection of secrets (e.g. private keys) using newly-added tamper resistant chips.

1.2 TCPA and Palladium Architectures

TCPA[2][3]

TCPA adds two new security components to a computing platform leaving existing components unchanged. These are, a) the Trusted Platform Module (TPM), used for storing integrity metrics and other platform secrets, and b) the Core Root of Trust Measurement (CRTM), the first software that runs during the boot process. To trust the TPM, the manufacturer must certify that it's genuine. Together, the TPM and the CRTM,

[2] Pearson, Preneel, and Proudler. *Trusted Computing Platforms: TCPA Technology in Context*. Prentice Hall PTR, ISBN: 0130092207, 1st edition, July 22, 2002

[3] Presentation at MIT Lab for Computer Science on *Trusted Computing Platform Alliance* by Joe Pato of HP Labs, 17 October 2002

is the simplest hardware enhancement required to form the “root of trust”. Once integrated with other components on a given platform, it provides the following features:

-Authenticated boot:

During the boot process, the first thing that runs on a trusted platform is the CRTM. The CRTM then takes the hash value of the next component to be loaded (the BIOS) and reports it to the TPM. The TPM then stores it in one of its registers. Control is then passed to the BIOS which recursively provides the same function to the next module to be loaded (the OS Loader). This process continues until the entire operating system is loaded. When a third party challenges the integrity of the platform, it is provided by the values stored in the TPM registers during the boot process, and based on these values the challenging party decides whether the platform is in a trusted state. Figure 2 is a diagram of the authenticated boot process.

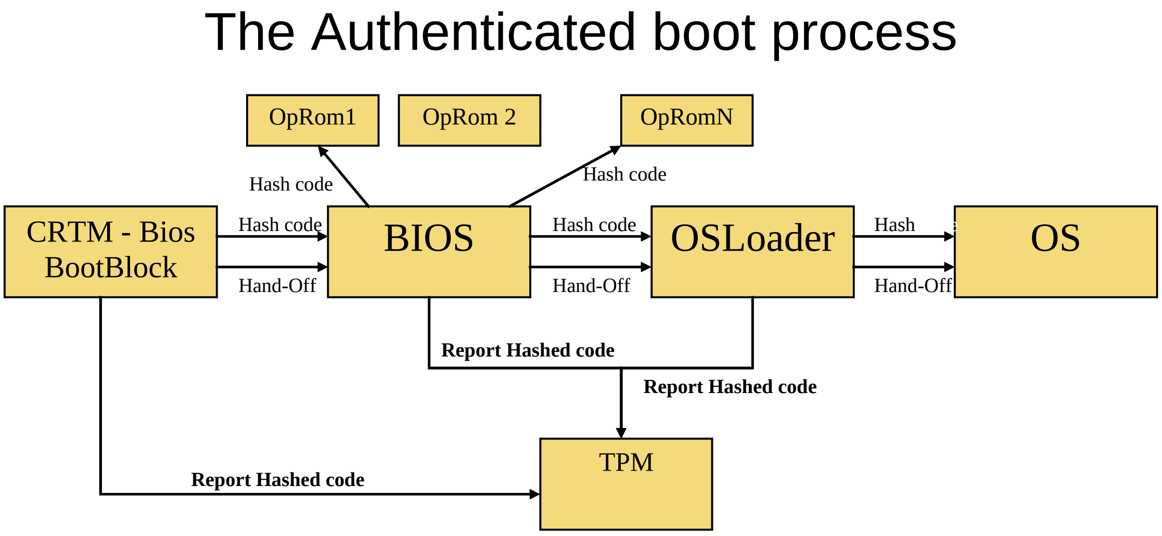

Figure 2

-Sealed Storage

Currently, most computing platforms protect data by encrypting it and storing the encrypted version in some storage media (e.g. Hard disk). However, this data remains vulnerable since there is no means to safely store the keys used for encryption. TCPA solves this critical problem using a tamper resistant chip, the TPM. The TPM has some security functions built in to it, such as encryption, signing, hashing and random number generation. This gives it the capability to sign crypto keys so applications can safely store them on disk for later usage. However, since the TPM has a very small amount of storage, there is a limit on the number of applications (or software states) that can take advantage of the TPM. However, this problem can be solved by adding another level of indirection. For example, the operating system can implement a key management service that interacts directly with the TPM; the OS can use the TPM to sign its own keys and use these keys to sign keys generated by other applications. If the OS completely isolates installed applications from the TPM, it can allow users to define there own policies for key management and access.

Strictly within the PC regime, TCPA lacks some functionality and security features that are provided by Palladium. TCPA, however, is cheaper since it adds to the hardware instead of redesigning it. Processor and memory manufacturers, for example, can use the same chips that they use today, thereby limiting the scope of the effort required for this industry-wide initiative. Also, legacy software may continue to run as if it was on a TCPA equipped computer. Finally, the TCPA architecture is not restricted to PCs and may be implemented on a wide variety of electronic devices including PDAs and cell phones. Figure 3 shows the components of the TPM.

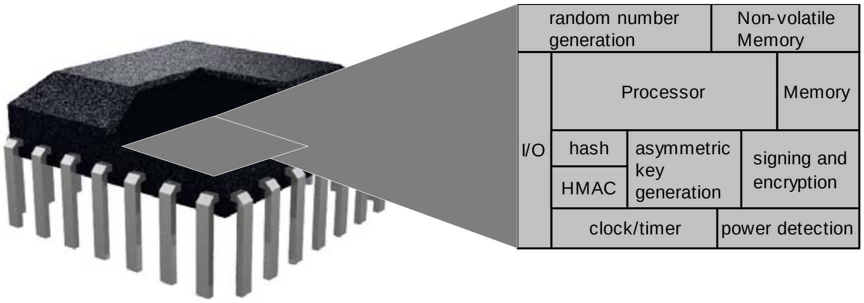

Figure 1

Palladium[4]

While TCPA is an attempt to enhance security in a computing platform by making a limited set of changes to ensure backward compatibility, Palladium is a larger scale initiative that requires significant hardware change. Besides the addition of new security components such as the Security Support Component (SSC which is Palladium's version of TCPA's TPM, Palladium also includes key modifications to the CPU, memory, and input/output architectures. Legacy applications, therefore, may have to be rewritten to take advantage of the new features found on the right hand side in Palladium.

-Trusted Mode vs. Standard Mode (right hand side vs. left hand side)

Traditional processors contain a control bit to distinguish between modes of operation: kernel (privileged) mode and user mode. This bit controls access to system resources such read/write to disk, memory allocation and network use. For security reasons, only processes running in kernel mode are granted access to these resources. Palladium introduces a new bit to this architecture, one to distinguish between trusted

[4] Presentation at MIT Lab for Computer Science on *An Overview of Palladium* by Brian A. LaMacchia of Windows Trusted Platform Technologies, 17 October 2002

mode and standard mode. And this time, only applications running in trusted mode are granted access to new security features introduced in Palladium. Colloquially, trusted and normal modes are referred to as the right and left hands sides respectively. Figure 4 shows the separation of trusted mode and standard mode.

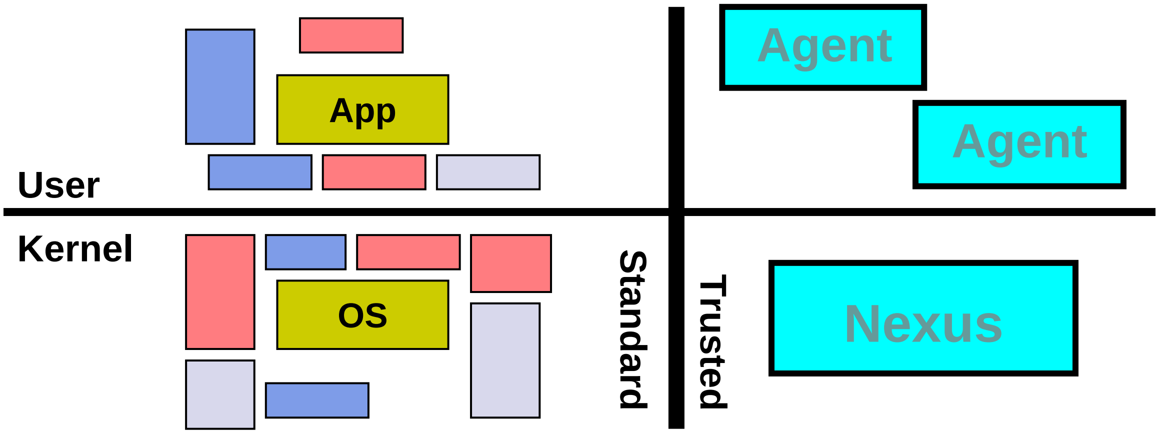

Figure 4

-The Nexus:

Once a user switches into trusted mode, the processor must boot the "Nexus". The Nexus is Microsoft's security kernel and provides a secure operating environment allowing what Microsoft calls agents to take advantage of the new Palladium functionality. The Nexus and the agents it calls need to be as small as possible to minimize bugs. The SSC verifies the code identity of the Nexus by computing and checking its cryptographic hash (Palladium uses SHA1) and storing it in one of its read-only registers. The Nexus interacts directly with the security services provided by the Palladium hardware and recursively provides the same services to agents running on top of it. For example, the Nexus may use the SSC to securely store its own keys, then use these keys to securely authenticate and store keys used by other agents. Agents may be standalone or provide services for other applications that run in standard mode. Figure 5 illustrates the relationship between agents, the Nexus and the SSC.

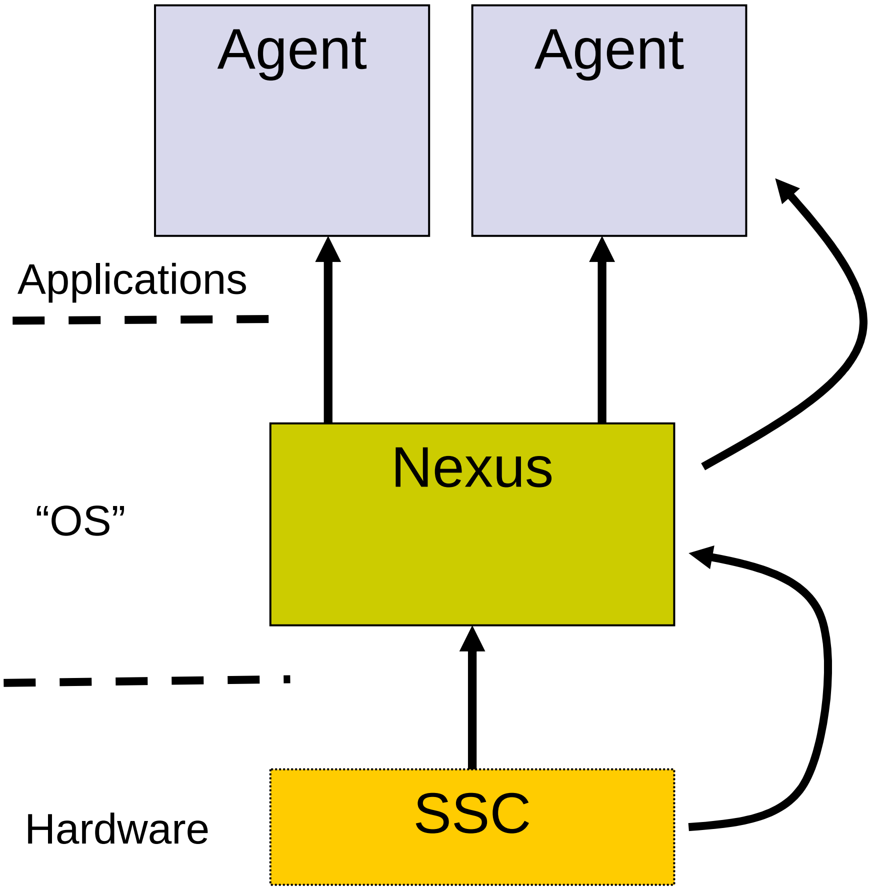

Figure 5

-Curtained Memory:

Palladium provides hardware based process isolation. The memory architecture is modified to allow pages of physical memory to be marked as "trusted". Trusted memory pages can only be accessed in trusted mode. Trusted applications, therefore, are not affected by bugs in the operating system running on the non-secure left hand side.

-Sealed Storage:

As described in TCPA: Sealed Storage.

-Secure IO:

Although secrets may be stored securely, the communication between trusted applications may be vulnerable. For example, when a trusted device that is connected to the USB bus communicates using an unencrypted channel with the nexus, it happens in the clear; any distrusted device listening on the same bus can snoop on the channel and expose these secrets to other malicious applications. To avoid this loop-hole, Palladium encrypts all Input/Output channels in trusted mode. In order to do this, Palladium requires an encrypted USB bus as well as an encrypted channel to the video card. As a result, whatever the user types on the keyboard remains confidential until it shows up on the screen; also known as "fingertip-to-eyeball" security as shown in Figure 6.

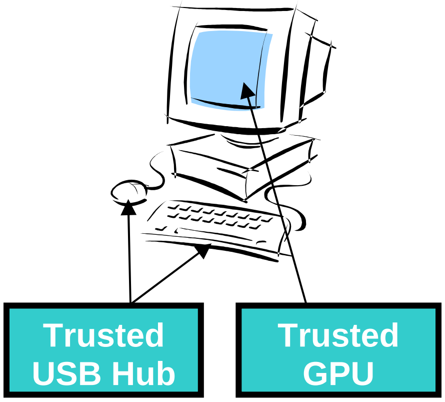

Figure 6

2 MALICIOUS CODE

2.1 Threat Model

Malicious code (viruses, worms, etc.) has been shown to be the most common cause of security infractions on an annual basis among respondents in the CSI/FBI survey. In 2002 alone, 85 percent of the respondents reported an attack by virus or worm outbreaks. Among the 188 organizations willing to respond in the CSI/FBI survey, the financial losses by such attacks of malicious code has steadily increased to $50 million, amounting to an average loss of $283,000 per organization. The total financial loss of 500 large organizations surveyed by the CSI and FBI amount to at least $150.1 million in losses due to such malicious code including *Lovebug*, *CodeRed*, *Melissa*, etc[5]. However, *Computer Economics* estimates the total worldwide impact for malicious code to have reached approximately $13.2 billion in 2001[6].

In order to model the threats posed by malicious code, the methods of propagation must first be examined. A typical virus is code that attaches itself to a program or an e-mail and is executed unknowingly, causing potential replication and damage to the end user. A worm is software that searches for vulnerabilities in network security and automatically transmits itself, through these holes to other computers on the network. Using the security hole, it can then search for additional gaps and rapidly spread across numerous networks throughout the world. Trojan horses are computer programs that appear to do one function but instead perform malicious actions once executed; these are

[5] Power, Richard

[6] http://www.computereconomics.com/article.cfm?id=133

not self replicating[7]. A secure computing environment must address network vulnerabilities to prevent worms such as *CodeRed* to disrupt network communications. Major e-mail viruses such as *Melissa* propagate its untrustworthy code through auto-exec commands and abilities to take advantages of weaknesses in the coding languages used by some e-mail clients.

2.2 How the Problem is Addressed Today

2.2.1 Signatures

Signatures are generated based on an application's unique hash. Current methods to generate hashes are good enough to provide code identification with reliability. Next, certificates are created to assert that a trusted source has verified that the code does not act maliciously on a system. If the user trusts the certifier, then he may trust the code to be safe. In Windows, there is a predefined list of trusted root keys. These root certificates are baked into the operating system, and stored in a location in the file system that is kept secret within Microsoft. The only way to maliciously get around certification is to modify the root trust list to include a new certificate. This new certificate can be used to certify code that has been signed with malicious intent. This attack would be fairly difficult to implement, but is certainly possible under today's system architectures. Once the location of the root trust list is discovered, it will be the same on every Windows installation, so successive attacks on other systems will be significantly easier after an initial successful attack.

[7] http://www.howstuffworks.com/virus2.htm

2.2.2 Sandboxing

Virtual machine software, Java for example, runs code in a memory protected environment. This sub-environment prevents the code that is running from accessing system resources at or above the level that the virtual machine itself is running. If access to the disk is required, then the virtual machine may set up a channel to allow it. This channel allows the virtual machine to restrict what disk access the application has thereby protecting sensitive data on the disk.

2.2.3 Bounded Memory

The concept of Bounded Memory is similar to sandboxing, although this works at a different level. Most current programming languages require the compiler to check memory boundaries each time a program writes to memory. This has a huge advantage as one can set an upper bound on the region of memory that the application has access to. A common hole used for system attacks is to cause an application to store data into memory and in the process write over other sections of memory that previously contained code that the operating system was going to run. This allows the hacker the ability to force a remote system to run arbitrary code, thereby giving him control.

2.2.4 Type Checking

Checks are made to ensure that the data that stored into a variable is of the expected type. Often times when data that is not the right type is stored in a typed variable, run time errors can occurs. These errors can sometimes be exploited to run arbitrary code on the system. Type checking can prevent such a run time error from happening in the first place. By preventing the error, the system is able to prevent a software attack.

2.3 How Palladium and TCPA can Help

2.3.1 Sealed Key Storage

Keys are stored in the TPM under TCPA and in the protected store under Palladium. The sealed storage module associates the key with a unique hash of the application that has stored the key. When a different application requests the key, Palladium and TCPA will generate a hash of that application and report it to the sealed storage module. If the hash has access to the requested key, then the key will be returned. Else, the application will be denied access.

2.3.2 Curtained Memory

Under Palladium, when an application runs on the right hand side, in the protected memory mode of the processor, the physical memory accessible to the application is different from the physical memory accessible to a left hand side application. The nexus ensures that an application running on the left hand side has access to areas of the left hand side memory only, and similarly for the right have side. This method will probably be very similar to the techniques used by modern operating systems to isolate memory between applications. The key benefit of Palladium is that applications that run on the left hand side will have no physical access to the memory of applications that run on the right hand side. This adds significant robustness compared to memory isolation that is available in modern operating systems. Under TCPA, where there is no left hand side versus right hand side distinction, an applications' hash is used to isolate memory application at the chip level. The processor itself manages the memory based on the hash generated by an application that wants access to the memory.

2.3.3 Code Signing

Code signing works today. Palladium and TCPA simply strengthen cryptography and signing because keys are secured in hardware in the sealed storage device. For example, a user can store the public key of some software signing authority in the sealed storage module under TCPA and Palladium. When code is authenticated, the public key of the signing authority can be compared to the key that is stored in sealed key storage. This means that the user will know if the identity of the certifier has changed. At some point however, the user will still need to trust the certifier who signs that an instance of code does not behave maliciously. The list of trusted signing authorities is stored in a secret location on today’s computers. If that location is discovered and exploited, the trusted authorities can be modified, allowing sources that are not trusted to sign code. With a TCPA or Palladium system, the trusted list can be encrypted using a key that is stored in the sealed storage module. This will prevent a software attack from modifying the list of trusted signing authorities.

2.4 Problems that Still Exist

2.4.1 Preventing Data from Being Erased

TCPA and Palladium both address the issue of denying malicious code access to sensitive content by encrypting the data on disk. However, malicious code can still erase any data stored on disk, and both TCPA and Palladium currently provides no solution to this problem. However, there is a potential solution, namely “Curtained Storage” that addresses this problem and involves a simple addition to Palladium. The idea behind curtained storage is the same as curtained memory: some blocks in the hard disk can be marked as “trusted”, and the disk controller must be modified to deny access to a block marked “trusted” unless the machine is in “trusted” mode. This feature is not in the Palladium specification, but could be added as Palladium evolves towards completion.

2.4.2 Need for Trusted Third Party

Code signing requires the need for a trusted third party to assert that a certain piece of code is safe.

2.4.3 Weakness of Limitations

The strength of the sandboxing method is also its weakness. While it allows for a protected environment through a virtual machine that has its own sub environment with restricted access to resources, it gives the user much less control and limits the versatility of an application. The more protected you make a virtual machine’s environment by limiting access to sensitive information on disk, the less ability an application has to get at the resources that it needs to function.

A large hole that exists with respect to system attacks is unauthorized writes to memory. By overwriting important information that is part of the OS for example, hackers can take over control of your machine. While bounded memory writes can protect against this exploit, one runs into the same restrictive problems that sandboxing causes. The less ability you have to modify important information in memory, the less control you have as a user to perform desired tasks.

2.5 Is the New Architecture Practical?

With sealed storage of keys in hardware, instead of fighting off a virus, there is a paradigm shift that focuses on protecting sensitive data. This allows for more user control in that sensitive data is protected on the right-hand side versus restricting what applications can do. While code-signing methods can still be bypassed, for example, malicious code that attempts to read sensitive data on the right-hand side will be restricted by the TPM. Application specific access to information will also prevent third party code from deleting files on disk as only the specific application/applications used to create the data will have any level of access to that data. Palladium allows for protection of sensitive data against malicious code but it comes with a price. Not only will applications that require encryption need to be re-written to match the specification for Palladium, the fact that data access can be completely application specific may make excellent third-party software often unusable. The operating system that is managing the storage of keys in the storage module can work in conjunction with an application to limit the access to a specific application’s keys. In this situation, one application would be able to store encrypted data on the disk in such a way that no third party application could decrypt the data because the key management module could deny access to the encryption key. If, for example, the license for a user’s program that was used to create a certain piece of data runs out, that user may have absolutely no access to that data. This is a potentially huge problem for the consumer.

3 STOLEN HARDWARE

3.1 Threat Model

Stolen hardware, including laptops, hard drives, and other media containing proprietary information valuable to third party clients, comprises a great threat towards computer security. Of 21 respondents in the CSI/FBI survey, the average loss per organization due to proprietary theft was approximately $6.6 million. According to the Office of the National Counterintelligence Executive, the "Annual Report to Congress on Foreign Economic Collection and Industrial Espionage" cited a figure of $100-250 billion in lost sales[8].

As intellectual property and proprietary information can be a corporation's most valuable assets, it is increasingly more important to provide security as computers increasingly are becoming the methods of safeguarding those secrets. One highly publicized incident of proprietary theft though hardware and software means was brought to trial in 2001. Two former employees of Lucent Technologies and an accomplice stole hardware and software marked proprietary from Lucent's Pathstar Access Server for use in their new business venture, ComTriad Internet. Many other companies that worked with Lucent to provide proprietary licensed software or custom built hardware were also affected by the thefts – this case is a pure example of the necessity to protect hardware and the information that resides within it[9]. Physical security measures on a typical computer can easily be broken and tampered with. A stolen hard drive or CD containing valuable information is readily accessible without supplementary protection. If data on stolen hardware is protected by a password, there might be ways to crack it. Even if the computer is protected by a biometric, as shown by the lucent case, there is not too much that can be done when the perpetrators already have appropriate physical access.

[8] Power, Richard
[9] http://www.cybercrime.gov/lucentSupIndict.htm

3.2 How the Problem is Addressed Today

3.2.1 Password Encryption

Files can be stored in a proprietary format that can only be read by a limited collection of applications. A password can be encrypted into the file. If the user can not match the file that has been encrypted into the file, then the application can deny access to the file. If the hard disk is stolen and plugged into a different computer, then the thief will not be able to immediately decrypt the data that is on the disk. However, pass phrases are not often longer than 16 characters making a brute force attack on the password possible.

3.2.2 Remote Key Storage

Files can be encrypted on disk using a key that is stored on a remote computer. This method is useful in network situations where it is essential to protect the data on a drive if it is stolen out of a terminal machine. The centralized key server can be configured to only accept connections from certain computers on a company intranet. If a hard drive is stolen and taken off site, it will be impossible to access the key that is needed to decrypt the data stored on the hard drive. This solution is only applicable in a company network environment, something that the average consumer does not have.

3.3 How Palladium and TCPA can Help

3.3.1 Sealed Key Storage

Under both Palladium and TCPA, files can be encrypted with keys that are stored in the sealed storage module. If the hard disk is stolen and plugged into a different computer, the files can not be decrypted because there is no record of the key on the hard disk. Only encrypted files will be protected, so any files that are left unencrypted will be just as vulnerable as they are in current computer systems.

3.3.2 Application Security

Applications can create unique signature keys and store those in the sealed storage module. If the hard drive is stolen and inserted into a different machine, the application can recognize the absence of the signature keys and refuse to load. This can apply to the operating system itself or individual applications.

3.3.3 Attestation

Both Palladium and TCPA provide ways for the operating system to attest truths about the hardware in a system. Attestation can be used at boot time to submit hardware IDs to a mandatory verification server. If the computer is connected to the internet, then the unique identity of all the hardware can be checked by the verification server to see if any of it has been reported stolen. In this model, it will be possible to limit the use of stolen hardware because there will be no way to have the operating system allow the system to function once it has determined that a piece of the hardware has been reported stolen. A thief would only be able to use stolen hardware if there was no way for the operating system to contact the verification server, and therefore the thief would not be able to use the stolen hardware on the internet.

3.4 Problems that Still Exist

3.4.1 Computers can Still be Stolen

If a computer is stolen, all the data on the computer is potentially compromised. It may be password protected or encrypted, but the decryption mechanism must also be stored on the stolen computer. For large networks, the keys used to encrypt data may be stored remotely. This solution clearly works in this narrow scope. This solution is impractical however for the average user at home.

3.5 Is the New Architecture Practical?

Sealed key storage using Palladium or TCPA will allow a user to securely encrypt data on the hard drive. If the hard drive is stolen, the data is still secure because the TPM chip is tamper resistant. The keys will self-destruct if someone tries to decipher the keys using the TPM thereby leaving no way to decrypt your data. All of this is only valid for the data on the right hand side and has been encrypted on the hard drive to begin with. Although this solution has added benefit, it is not enough for the average user who doesn’t have sensitive information on his machine, which if stolen by a third-party, may cause significant harm. For large data-centers, servers and corporate users, sealed key storage is useful. As explained, however, remote key storage works just as well as hardware-sealed key storage in this case.

4 E-MAIL AUTHENTICATION

4.1 Threat Model

E-mail is rapidly becoming the prime method of communication as Jupiter research reports that approximately 12 billion e-mail messages were sent daily in 2001. The number of corporate mailboxes is growing at a rate or 32 percent per annum[10].

The most common issue with secure e-mail is invasion of privacy; the sender of the message would like for the intended recipient to be the only party to retrieve the

[10] http://www.directcon.net/dcweb/spam/stats.htm

message. Additionally, the integrity of e-mail is a major concern. It is important that the message content not be altered or tampered with while being processed through the mail systems. Finally, the third major threat to e-mail security is authenticity of the sender. The true identity of the sender of the message must be trusted and known to guarantee proper security in e-mail. Issue of eavesdropping are currently a big concern in 802.11 wireless networks as attackers can gain admittance to a network and disrupt transmissions[11]. Although e-mail clients are improving security by means of encryption, if any of the three measures are broken, e-mail security cannot be guaranteed.

4.2 How the Problem is Addressed Today

4.2.1 Signatures

RSA keys can be used to sign e-mail before it is sent. When someone receives signed e-mail, they can use the RSA public key of the sender to verify that it was signed with the proper private key. In order for this system to work, the recipient must have a valid copy of the sender's public key. In addition, the private key must be kept secure from other users who want to assume an identity and send a fake e-mail.

4.3 How Palladium and TCPA can Help

4.3.1 Signature Storage / Verification

TCPA / Palladium provide a way for the user to securely store a signature key in the sealed storage module. Because the keys are stored in the secure hardware, there is no

[11] http://www.oreillynet.com/pub/a/wireless/2002/05/24/wlan.html

way for an imposter to copy the user's private keys onto a different system and use those keys to send fake e-mail. Today it is possible to extract private keys from a system and use those keys from a different location.

The recipient can store a sender's public key in the sealed storage module. In this way there is no way that the public key of the sender can be falsely published causing the user to verify a fake signed message with a fake public key.

4.3.2 Strong Pass Phrase

Another risk of e-mail authentication is that an imposter uses someone else’s computer and pretends to be that individual and sends out e-mail. The only way to prevent this is to require a password or biometric identification before sending the file. The password hash can be encrypted under a key that is stored in the sealed storage module. Encrypting the password hash will prevent an arbitrary application from brute force attacking the password because the password will be encrypted under a key that only the e-mail program has access to. Under Palladium, where there is a secure IO channel, there will be no way to have a program run to spy on the user's keystrokes to catch the password that is used to send e-mail. Under TCPA this is not made as clear, but there could be a system set up that would not ask for the e-mail authorization password unless a certain hash of running applications is identified. This would prevent a password snooping application from running in the background.

4.4 Problems that Still Exist

4.4.1 Origin of Signatures

Regardless of how signatures are created and transferred on a network, it will remain challenging to positively verify that a signature belongs to an individual. This problem is solved today by creating signatures in the presence of others who attest that the signature truly belongs to the individual who it is created for. Apart from how the keys are created, an individual will need to be trusted to attest that the key was properly created to start. This problem can not be solved by any hardware of software modification to the system of using digital signatures.

4.5 Is the New Architecture Practical?

Palladium and TCPA use a secure signing key that is stored in the sealed storage module. This will allow you to decipher, with certainty that the e-mail came from a specific machine. If there is some authority that has a list of machines and its owners, a user can know with certainty that the e-mail came from a trusted person. However, a source that keeps a list of machines and its owners is far too difficult to practically implement today. The only added advantage Palladium has is that a user can know with more certainty that the machine's sealed keys are not compromised. Again, this is a marginal benefit that does not justify an upgrade to a Palladium or TCPA equipped machine.

5 SPAM

5.1 Threat Model

Spam is a highly controversial topic because it exists at the crossroads of the fundamental rights for freedom of speech and privacy. Spam can be defined as unsolicited bulk e-mail sent by an entity unknown by the receiver. There are obvious advantages to spam regarding publicity and promotion; it is one of the most economical and far reaching methods for disseminating information about a specific interest or cause. Jupiter Media Matrix predicts an amount of 268 billion annual spam messages by 2005, costing the spammer an average of $.0000032 per message relayed[12]. Since traditional advertisement costs are borne by the advertiser, spam provides an outlet that shifts the bulk of the cost to the ISPs and end-users. These astonishing figures give reasoning for the fact that spam usage is a growing phenomenon.

Spam itself does not typically have malicious intent upon the receiver. However, it is a major concern to the public, and its regulation and control must be addressed for several reasons. One of the primary concerns to the public is that spam costs individuals time and productivity. Given the high volumes of daily occurrences, notification of incoming mail can provide a major distraction for end users attempting to maintain legitimate electronic mailboxes. More than simply checking mail, spam disrupts the integrity of e-mail as a means of effective communication. Serving as noise in the stream of solicited e-mail messages, spam decreases the signal to noise ratio of e-mail communication; this can significantly disrupt productivity as users may close or ignore mailboxes due to the high volumes of unwanted mail, possibly losing important mail as well. Additionally, spam costs the Internet Service Provider (ISP) industry $8-10 billion worldwide in bandwidth costs alone according to PC Magazine[13]. With spam being a global problem, it is not easy to track it when arising from insecure servers.

[12] http://www.buzzle.com/editorials/11-2-2002-29468.asp
[13] http://www.buzzle.com/editorials/11-2-2002-29468.asp

5.2 How the Problem is Addressed Today

5.2.1 Black Lists

Black lists are compiled and shared on the internet to allow an e-mail client to check to see if an e-mail is from a known spam sending domain. If the e-mail is determined to be from a known spam source, then the client can choose to delete the e-mail or filter it into a spam folder in the e-mail client. This method is good for detecting large quantities of spam that originates from spamming domains; however, this technique does nothing to catch spam that comes from new spam domains or from otherwise trusted domains.

5.2.2 Filters to Allow only Approved E-mail Senders

Rising in popularity is the idea to have an allow list for trusted e-mail sources. By filtering out all e-mails except from ones that come from trusted individuals, spam can be completely eliminated. The problem with this technique is that the recipient will need to have a list of all possible senders that are allowed to send e-mail to the recipient. It is very likely that e-mail will come from a sender who is completely legitimate, but unexpected. In this case, the possibly important e-mail will be filtered out just because the recipient did not predetermine the sender to be trusted. This technique is incredibly effective at eliminating spam, but at the same time drastically reduces the benefits of e-mail by preventing a recipient from receiving important e-mail from unknown senders.

5.2.3 E-mail analysis

Spam filters are being made today that use intelligence algorithms to look at the text of an e-mail to determine if the content is spam. This technique often picks up trigger statements such as “click here to remove yourself from the list” or “act now, once in a lifetime opportunity." Intelligence algorithms can be trained against sets of spam resulting in the ability to filter a large majority of the e-mail that someone receives. The problem with intelligence algorithms is that they sometimes produce false positives and identify important e-mail as spam. This is a huge problem, considering that it is far worse to lose one important e-mail than to receive a few hundred spam e-mails. These kinds of filters are good, however, at flagging e-mail as spam so that the user can look through it quickly to determine if any of it may be important before deleting the messages.

5.3 How Palladium and TCPA can Help

5.3.1 Signatures

Palladium and TCPA both provide strong ways for users to store RSA signing keys. These signing keys can be used to sign e-mail before sending it. When a user receives an e-mail, that user can check to see if the signature matches ones that are already known, or can go to a CA to see if the signature came from an individual that the recipient is likely to trust. This method would allow a CA to return information to the recipient's e-mail program saying that the e-mail came from a business or private citizen. The recipient may have configured his e-mail program to ignore e-mail that is sent from a business unless the business is on a trusted list that is maintained by the e-mail client. Signatures and CAs will allow e-mail clients to gather more information about the origin of an e-mail message so that better decisions can be made as to whether or not the e-mail message is spam.

5.4 Problems that Still Exist

5.4.1 Palladium can be Turned Off

The idea of an originating machine signature can help to solve the problem of spam. All new computers could use the sealed key to assign you a certificate signed by your machine assuming your machine is trusted. This would make tracing back to the computer that is sending the spam much simpler. The big problem here however is that you are putting the onus on the spam sender to declare who he/she is - exactly what he/she is trying to avoid. Also, since Palladium can be turned off, spam senders will choose to do so while sending out spam. At the receiving end, your e-mail client might require that all e-mail be received only from a Palladium equipped computer, but this clearly is impractical and hinges on the assumption that everyone will switch to Palladium soon after is released.

5.5 Is the New Architecture Practical?

Palladium adds very little with respect to blocking spam. Perhaps a combination of anti-spam software along with an accurate way to trace a machine will help the initiative, but this marginal benefit doesn't outweigh the cost of switching to Palladium.

6 SPOOFING

6.1 Threat Model

Forging the source address information is part of a growing problem in cyber security known as spoofing. This procedure is conducted by first finding the Internet Protocol (IP) address of a trusted port to the victim. The packet headers of information relayed must then be modified so the packets appear to be originating from a trusted port instead of from the hacker. The possibilities for malice are very high as spoofing can allow an attacker to engage in identity forging and conduct secure transmissions acting as a legitimate secure web site. Given trust, certificate based authentication systems can be tricked into allowing entrance to a hacker posing as a privileged user. Ensuring authentication of users on both ends is a necessity for trusted computing. Spoofing is a threat that must be addressed quickly as it affects large and small users alike.

Financial fraud and unauthorized access are real threats to computer security. In 2001, two employees of Cisco Systems illegally tied to purchase $8 million in stock with unauthorized access. Financial fraud, a significant threat caused by spoofing, cost 25 respondents in the CSI/FBI survey a total of $115.8 million in 2002[14].

6.2 How the Problem is Addressed Today

6.2.1 Certificate Authorities

Certificate authorities (CA) maintain records of certificates that the CA has authenticated as belonging to the proper server. When a client connects to a server and gets a copy of the certificate from that server, there is a signature on the certificate stating which CA has certified the certificate. The client can then go to the CA and check to see if the host computer is the same one that was used to generate the certificate that the CA authenticated. This security prevents a fake server from identifying itself with a stolen certificate. The CA will help the client determine that the computer that the certificate was originally generated on was different, and therefore there is a possibility that someone is intercepting and redirecting the connection.

[14] Power, Richard

6.2.2 SSH Key Fingerprints

When an RSA key pair is generated for SSH, there is a corresponding unique key fingerprint. When a client connects to the host for the first time, it often stores the key finger print locally. This allows the client to verify the key fingerprint the next time that the connection is made. If the key fingerprint is the same, then the client knows that the server is using a different RSA key pair. This will certainly allow the client to know if the server has been honestly changed, but this method of protection is not as likely to work if the server is maliciously spoofed in a way that the server is using the same RSA key pair as the original server.

6.3 How Palladium and TCPA can Help

6.3.1 Sealed Key Storage

Key storage prevents a malicious individual from extracting the RSA key pair in order to create an identical malicious server to fake transactions to the client. Without being able to extract the keys, then the client computer will be able to use RSA fingerprinting to identify if the server has been compromised and choose not the send sensitive data.

6.3.2 TCPA Application Hash

On a TCPA system, the authenticated boot process will provide a hash of all the applications running on the system. Once a client has connected to a server, the client can request this hash to determine if the running applications are trusted not to have flaws that will allow sensitive data to be leaked.

6.3.3 Palladium Nexus

Under Palladium, if the client is connected to an application that is running under a nexus, then the client can be sure that the nexus will not let any other applications see the data that the client has passed to the server. Both the Palladium and TCPA methods for ensuring that faulty applications don't have access to sensitive data are good ways of protecting data once it has been transmitted securely.

6.4 Problems that Still exist

6.4.1 Need for a Trustworthy Certifier

Client must initially trust that the certificate from the server has been authenticated. Palladium and TCPA systems will only help to ensure that a server is in the same state that it was when the user first connected to it. There is a need for a reliable certifier that will initially authenticate that a certificate really belongs to the server that the client would like to connect to. This is done by signing the certificate that is on the server that a client connects to. If the client trusts whoever signs the certificate, then the trust can propagate down to the server that the client has connected to. Without the initial identification, there is no way that a Palladium or TCPA system can prevent a client from connecting to a malicious server.

6.4.2 Server Hardware Upgrades

Server hardware upgrades will look like fake servers because of an ID violation when the client connects. This will happen if the hash of the server configuration changes due to upgrade, and the client uses the hash to identify the server.

6.4.3 Hot Swap Backup Servers

Hot swap backup servers will not be easily possible because the keys on each identical system will be different. This is because it would be bad to allow for the keys to be extracted and migrated between machines. If the keys can be extracted and migrated, then the system can be hacked and the keys can be stolen. Often, when a server goes down, often large companies have a backup server that is identical to the running server that they can switch to allowing constant up time. When servers are initially ordered, it may be possible to request that the systems have the same hardware identity. This would allow for a backup server to have the same identity, and allow for a hot swap. This adds additional restrictions to server applications because backup servers will need to be purchased ahead of time and coordinated with the hardware manufacturer.

6.5 Is the New Architecture Practical?

Sealed key storage prevents a malicious individual from using a RSA key pair in order to create an identical server to fake transactions to the client. The Nexus provides additional security by not letting any other applications see the data that the client has passed to the server. Both the Palladium and TCPA methods for ensuring that faulty applications do not have access to sensitive data are good ways of protecting data during transmissions. Trusted computing measures will create difficulty for businesses when planning for backup servers, as the practice of hot swap effectively is impossible unless manufactures set the same hardware keys in conjunction. TCPA and Palladium are beneficial in limiting the threat of spoofing; however, the practicality for replacing one's system for this security is questionable.

7 P2P RESOURCE SHARING

7.1 Threat Model

Peer to peer (P2P) resource sharing over a network is increasingly becoming a common method for transferring and sharing resources between computers. Along with the increased use of P2P networks for network computing, increased security and trust issues that come with the anonymous freedoms available in cyberspace grow as well. Of the prime concerns associated with P2P networks, one of the fundamental causes of network insecurity today can be that an individual can only trust personal data. Data collected from a foreign source cannot be necessarily trusted as its origins might not be known. Additional threats posed from P2P networks involve the trust in one's own data being processed in a remote computer over the network correctly. As it cannot easily be certified that the data was processed correctly, it cannot be openly trusted to be valid. Finally, the storage of data remotely in a P2P network can lead to issues of tampering and misuse. As an individual would have little or no control of the data once it leaves his computer, it is susceptible to unauthorized modification and distribution.

A recent example demonstrating the potential of falsely returned data after processing over a P2P network can be found with the SETI@home program. In this project, millions of volunteers donated their computers to process information to aid the search for extraterrestrial life. However, less than one percent of the participants were able to return falsified results to the program, in a juvenile scheme. Despite being a negligible portion of the population, the act caused a problem that consumed a significant amount of the project's time to repair[15].

7.2 How the Problem is Addressed Today

7.2.1 Beowulf Clusters

Beowulf clusters are massive parallel computing clusters. Each node in the cluster is a system that typically runs a free software operating system such as FreeBSD or Linux. By clustering the machines, applications can run on different nodes in the cluster and the data that is processed can be transferred between nodes over the private network of the cluster. Each node in the cluster assumes that when data is sent to another node, that data is properly received, processed and returned without outside modification. This assumption is often safe to make in a Beowulf cluster where the nodes are connected via a private internal network. The Beowulf architecture is not good for allowing anonymous users to add their computers as nodes to an existing Beowulf cluster. There is no way to ensure that the data that is processed by the new node can be trusted because the owner of the machine and his intent are unknown.

7.2.2 Windows Networking

Distributed storage systems are available today. There are several different varieties of distributed solutions that are in use today. Microsoft Windows comes with Windows networking built into the operating system. Windows networking will detect other Windows computers running on a local network, and allows for folders to be shared over the network to anyone who knows a valid username and password for computer that is sharing the files. This method of distributed storage is not an automatic one, as files must be relocated by the user to balance the storage across a network appropriately. It

[15] http://www.space.com/searchforlife/setihome_cheats_010524.html

would be advantageous to have a distributed storage network that would automatically relocate files to computers on the network that have more storage space than that computer where the file originated. The problem with automatic file distribution is that there is no way to trust the machines that the data is to be stored on. If a user has a sensitive file, or a large data file that he wants to ascertain does not get corrupted, there is currently no method to ensure that if the data is automatically relocated to a remote computer, that the user will be able to get that data back, uncorrupted, in the future.

7.3 How Palladium and TCPA can Help

7.3.1 Machine Identity

RSA signing keys can be stored in the secure storage module. The signing keys can be used to identify the computer on a network where it is important to positively identify other computers on the network. By storing a unique signing key, data can be signed before returning it to a remote computer to further authenticate data by ensuring that it is the same as it was when it came from the remote computer.

7.3.2 TCPA Application Hash

The running applications on a remote computer can be identified to determine if there are any applications that may cause a risk to the data that a user would like to have processed on a remote machine.

7.3.3 Palladium Nexus

When a nexus is running on a Palladium machine, the application that receives the data to be processed can have exclusive rights to the data. Curtained memory will prevent other applications from snooping on the data and modifying it directly in the memory registers. The nexus will work at a higher level to prevent other applications from interacting with the application that is currently processing the data.

7.3.4 Sealed Key Storage

Applications can encrypt sensitive data, store the key in the secure storage module and then store the encrypted data on a remote computer. The only way that the remote computer can manipulate the data is to delete the data. This can be good for a user because the user can be sure that if the data is returned from the remote computer and it is decrypted properly, then the user can be sure that the data has not been modified. There is a chance that the remote computer will delete the data, or modify it so that the data will no longer decrypt properly. In both cases, the user will not be able to get to his data in an unencrypted form either because it was deleted or because it was modified after it was encrypted. In either case, there is no way for the user to be fooled into believing false data. This is an all or none model, where the user will have perfect data or no data at all. In many cases this is an acceptable solution, especially when the data can be distributed to multiple remote computers where the chance of losing all the copies decreases.

7.4 Problems that Still Exist

7.4.1 Data can be Deleted

There is no way to ensure that a malicious application does not delete shared data that is stored on a computer. The application can randomly place the data on the hard drive and encrypt the location so that a malicious program can not determine which file to delete; however, there is nothing preventing an application from modifying the data directly on disk.

7.5 Is the New Architecture Practical?

A reliable machine ID grants the user access to the identity of each node on the network. He may then choose whether or not he/she trusts the network or not. This hinges on a CA however with a list of machine IDs and their corresponding users. Also, trusted applications that are known by the user will be given access to the data; therefore data stored remotely is not modified without the user’s knowledge.

CONCLUSION

The following table gives a short summary of each threat model and how practical it is to solve the threat with a trusted computing platform.

Threat	**TCPA and Palladium Solution Practicality**
Malicious Code	Code signing is strengthened by storing keys in sealed hardware storage. “Curtained Storage” provides a venue to physically protect sensitive data by maintaining a trusted section of storage. TCPA and Palladium provide a practical solution to threat although freedom to manipulate secured data is made more difficult.
Stolen Hardware	The practicality of TCPA and Palladium differs for large corporation and for the personal user. Attestation will prevent a stolen computer from operating on the internet, discouraging theft. Encrypted data is secured on a stolen hard drive with an absence of the original encryption keys provided during installation. However, since corporate intellectual property theft by hardware is largely done by internal agents, TCPA and Palladium cannot prevent such attacks.
E-Mail Authentication	Palladium adds a valid machine ID key using TPM with e-mail, there fore ensuring that an e-mail originated from a specific machine. However, it is not guaranteed that e-mail is submitted by a specific person, and therefore trusted computing is not a practical solution to e-mail authentication.
Spam	Palladium provides no practical solution to eliminate spam as it relies on the spammer to identify his machine signature. Spam senders will opt to turn Palladium’s features off when sending bulk mail. Although Palladium can require all e-mail to be received from Palladium based machines, the scenario that everybody upgrades to Palladium immediately s improbable.
Spoofing	Key storage prevents a malicious individual from extracting the RSA key pair in order to create an identical malicious server to fake transactions to the client. The Nexus provides additional security by not letting any other applications see the data that the client has passed to the server. Both the Palladium and TCPA methods for ensuring that faulty applications don't have access to sensitive data are good ways of protecting data once it has been transmitted securely
P2P Resource Sharing	By storing a unique signing key, data can be signed before returning it to a remote computer to further authenticate data by ensuring that it is the same as it was when it came from the remote computer. However, this will not be sufficient to prevent possible falsified data processing in computing grids using a P2P network.

We analyzed six different threat models and looked at how they are being dealt with today and how they can be dealt with in the trusted computing framework. In each threat model, we compared and contrasted different solutions based on how well they work, their ease of use, and simplicity of implementation. One important lesson one can learn from our analysis is that very few applications require the new features that come with a trusted computer. Therefore, a regular consumer won't be so thrilled about having to upgrade their PCs only to get the features of TCPA or Palladium. Therefore, we suggest that this new wave of trusted computing stay out of the general purpose computing platform, and if some applications demand trusted computing, then they can use a special purpose device to achieve their goals. This way, consumers who don't care don't have to change their lifestyles, but at the same time, the technology is available to whoever needs it.

ACKNOWLEDGEMENTS

We would like to thank Professor Hal Abelson, Joe Pato, Barbara Fox, and Danny Weitzner for their help in researching and writing this paper, especially Joe Pato who was the group advisor for this paper.

This paper was written for Hal Abelson's MIT class 6.805, *Ethics and Law on the Electronic Frontier*, Fall 2002. Omar Bakr wrote the description of TCPA and Palladium. Hareesh Nair examined the threat models. Tony Scelfo served as the editor for the paper, described how the threats are addressed today, and showed how they would be addressed on a trusted computing platform. Aman Narang wrote about how practical a trusted computing platform is for addressing each threat.

APPENDIX A: CSI/FBI Respondent Composition

The respondents for the CSI/FBI survey were comprised from significant portions from high tech (19%), financial services (19%), and manufacturing (11%). Federal, state, and local government agencies combined comprised of a significant portion or respondents (19%) and the remainder of respondents were derived from other large national sectors of business (such as medical institutions - 8%, telecommunications - 5%, etc.).

The organizations surveyed tended to employ large numbers of people, with 24% reporting higher than 10,000 employees. Additionally, these organizations made a significant impact on the economy, with 37% of companies in the private sector attaining income of greater than $1 billion.