A Portable TPM Based on USB Key

Dawei Zhang
School of Computer and Information Technology, Beijing Jiaotong University, China
dwzhang@bjtu.eu.cn

Zhen Han
School of Computer and Information Technology, Beijing Jiaotong University, China
zhan@bjtu.edu.cn

Guangwen Yan
Research Division, Beijing Watchdata System Company, China
guangwen.yan@watchdata.com

ABSTRACT

Trusted computing technology aims to enhance the security of platform by the TPM. But there are some drawbacks of TCG's Trusted Computing architecture for user-based applications. This paper presents a new concept of portable TPM (PTM) based on USB Key to solve those problems. At first, we use PTM to establish a trusted path between the verifier and the user in remote attestation so as to propagate the trust chain to the end user. Secondly, we design the trust model and platform management mechanism of PTM. In this model the single point failure of TPM and frequent sensitive data migrations between different platforms are avoided based on PTM. At last, we implement the PTM on the USB Key with Java Card Runtime Environment. The test results show that the PTM scheme is feasible for user-based application.

Categories and Subject Descriptors

D.4.6 [**Operating Systems**]: Security and Protection

General Terms

Security

Keywords

Trusted Computing, Trusted Platform Module, Java Card

1. INTRODUCTION

Trusted Computing (TC) is a technology developed and promoted by the Trusted Computing Group.With Trusted Computing, the computer will consistently behave in expected ways and those behaviors will be enforced by hardware and software. The basis of Trusted Computing is the **Trusted Platform Module (TPM),** which is special-purpose integrated circuit built into a variety of platforms to enable strong user authentication and machine attestation. Computing and communications products with embedded TPM advance the ability of businesses, institutions, government agencies, and consumers to conduct trustworthy electronic transactions [1].

Remote attestation in TC enables computing devices to verify each other with TPM, but there are some problems to adapt it to user-based attestation in this architecture. There is no reason to trust one computing device any more than another in a networked world. Treating these devices as turtles, the user who seeks a trustworthy system from which to verify others quickly realizes that it's "turtles all the way down" because of the endless loop of trust dependencies [2]. Those problems of user-based attestation with TPM include:

- In a networked world, it is unclear to the user why the device that she uses as the verifier is any more trustworthy than her other devices.
- The trust chain established by remote attestation with TPM does not propagate back to the user. There is no trusted path between the remote verifier and the user because the malware on the user's computing devices can lie about verification results (see Figure 1).

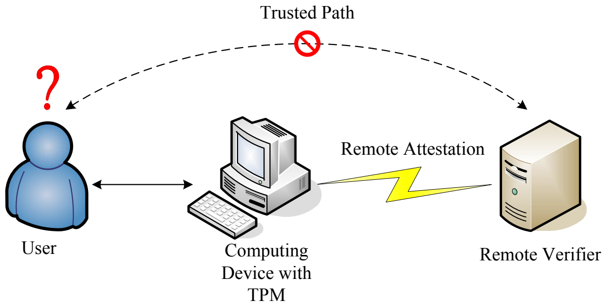

Figure 1. Remote Attestation with TPM

In order to solve two above problems, a customized USB device with a light, which is called *iTurtle* and certified by the trusted party, is used as an axiomatically trustworthy device to perform verification in paper [2]. In other words, it's the "turtle" on which all user trust is built. They proposed a user-observable verification method. If the verification succeeds the light on *iTurtle* is green light, otherwise it is red. Furthermore, they proposed that the software design of *iTurtle* should avoid cryptographic algorithms to eliminate the overhead of maintenance. But we think it's difficult to establish a trusted path for transportation of verification results between the user and the remote verifier without cryptographic operations.

Besides remote attestation, the TPM will also conduct integrity measurement and act as the Root of Trust for Storage (RTS).One computing device has only one TPM until now and all users on this device share it (see Figure 2). There are two problems in this TC architecture:

- A single point failure of TPM. If the only TPM does not work on specific computing device, integrity measurement and remote attestation are not available; the user can not restore their keys and data protected by TPM especially when those BLOBs are bound to a particular TPM. A complex backup mechanism must be introduced to avoid this issue. There is no specific solution about this problem from TCG until now.

CCS'10, October 4-8, 2010, Chicago, Illinois, USA.
ACM 978-1-4503-0244-9/10/10.

- Generally speaking, one user often has several computing devices today. But the sensitive data are bound to one specific device with TPM. If she wants to use protected BLOBs of one platform on another device, complex migration service is needed [4]. It is very inconvenient and dangerous for common users.

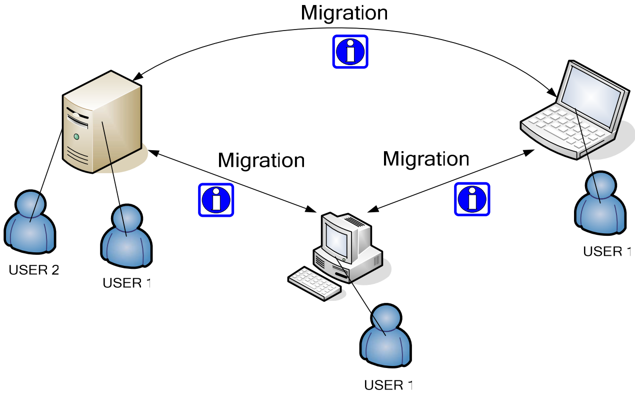

Figure 2. Relationship between Users, TPMs and Platforms

In order to solve four above problems, we propose a portable TPM based on USB Key as an axiomatically trustworthy device in this paper. We call this device **Portable Trusted Module (PTM)**.

2. TRUSTED COMPUTING ARCHITECTURE WITH PTM

The hardware of PTM is a Watchdata USB Token with a little LCD screen and button. The picture of PTM is shown in figure 3.

Figure 3. Picture of PTM

There is a Java Card Runtime Environment [3] in USB Key and we implement some TPM commands [4] and cryptographic operations with Java on Key. PTM communicates with computing device by USB. In other words, it is a TPM-like device with USB bus.

2.1 PTM as an axiomatically trustworthy device

PTM is certified by the trusted party (CA, bank, government etc.) with a public key certificate so that the user can trust it. In fact, it is a trusted user agent in e-transactions [5]. The little LCD screen on Key can not be manipulated outside the USB Key and provides the user-observable verification. Only the software on Key can control it to display the trusted verification results.

We design a simple protocol as a demo to establish the trusted path between PTM and the remote verifier with cryptographic operations on PTM. At first, we use RSA asymmetric encryption algorithm *E(message,key)* and assume that the public key certificates of both sides are exchanged in a secure manner. Secondly, we define:

public key of PTM - PK_{PTM}, private key of PTM - SK_{PTM}

public key of remote verifier - PK_{RV}, private key of remote verifier - SK_{RV}, verification result - *VR*

The demo protocol for trusted path as follows:

(1) PTM generates a random number R_{PTM} and do $ER_{PTM} = E(R_{PTM}, PK_{RV})$

(2) PTM send ER_{PTM} to remote verifier

(3) Remote verifier do $R'_{PTM} = E(ER_{PTM}, SK_{RV})$

(4) Remote verifier generates a random number R_{RV} and do $ER_{RV} = E(VR || R'_{PTM} || R_{RV,} PK_{PTM})$

(5) Remote verifier send ER_{RV} to PTM

(6) PTM do $E(ER_{RV}, SK_{PTM})$ and compare R'_{PTM} and R_{PTM}

(7) If $R'_{PTM} = R_{PTM}$, PTM display *VR* on LCD screen, else display V*erification Fail*.

The trusted path between the user and remote verifier is established because the malware on computing device can not modify verification results and control their display. Therefore, the first two problems discussed in section 1 are solved with PTM.

2.2 Trust model with PTM

PTM is a TPM-like device and can provide some TPM functionalities (PTM can not implement all commands of TPM because of limited resources on USB Key). PTM is a user-based attestation device so that there are some differences between TPM and PTM in application scenarios. In TCG TPM scenario, one TPM is bound to one computing device (platform) and several users use one TPM (See figure 2). But in our PTM scenario, one PTM is bound to one user and several computing devices use one PTM. The relationship between users, PTMs and platforms is depicted in figure 4.

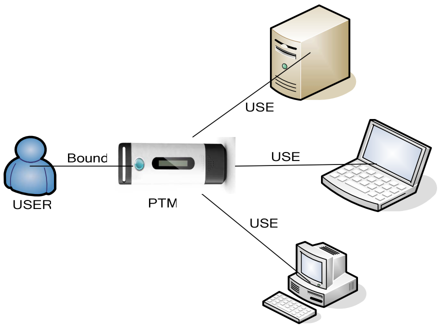

Figure 4. Relationship between Users, PTMs and Platforms

In our design, one user has one PTM and others can not login the PTM without user PIN. We assume that the user's Storage Root Key (SRK) and Attestation Identity Key (AIK) in different platform TPMs can be migrated into this user's PTM with platform management module on computing device and PTM. At this time, PTM become a backup TPM on a platform. The difference is that TPM communicate with the platform by LPC bus but PTM by USB bus. We will explain how we implement USB communications between PTM and platforms by Extensible Firmware Interface (EFI) [6] in section 3.

Now we consider the third and fourth problem discussed in section 1. If the platform TPM failed, PTM can be used as a backup trusted module because PTM can act as the original TPM by AIKs in it. When the user roams among different platforms, she can use PTM to access protected data and keys because different platforms' SRK in PTM can be used to un-seal BLOBs.

3. IMPLEMENTATION OF PTM PROTOTYPE

In computing device platform, we must solve the problem of loading PTM in boot sequence as early as possible. Intel EFI allows the developer to extend platform firmware by loading EFI driver and EFI application images so as to control system boot procedure. In the phase of Drive Execution Environment (DEX) of EFI, we load the USB bus driver that can recognize the USB device and start up it before OS loading. At this point, platform can use PTM just as TPM. The implementation details can be found in our paper [6].

We choose a Watachdata USB Key with Java Card Runtime Environment and implement TPM commands as a Java Card Applet. Generally speaking, we implement a TPM command interpreter on USB Key. All TPM commands are encapsulated in APDU command because APDU is the standard communication protocol in Java Card. Besides TPM commands, we also provide some platform management commands for PTM. The program modules and commands in PTM Applet are shown in figure 5.

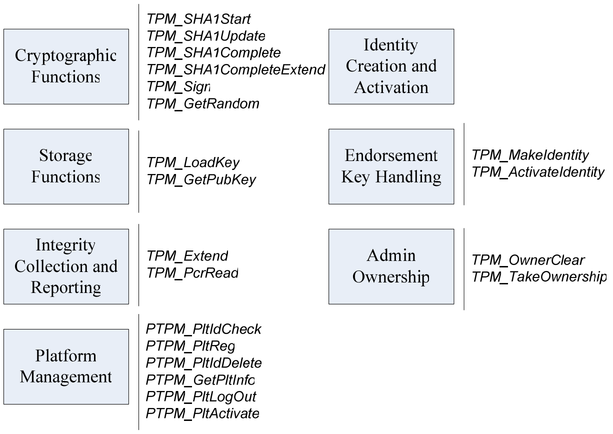

Figure 5. Program Module and Commands

4. TEST RESULTS

The PTM performance will influence its application in information systems. Therefore we give the test results of PTM command response time in our test environment. The hardware platform of PTM is a 32-bits RISC CPU with cryptographic accelerator. Its maximum frequency is 96MHZ. The computing device platform is a laptop. The CPU frequency is 1.5G and operating system is Windows XP. The applications communicate with PTM by PC/SC driver. The response time of commands is shown in table 1.

Table 1. Response time of PTM Commands

PTM Commands	Run time(ms)
PTM_SHA1Start	0.66
PTM_SHA1Update	0.66
PTM_SHA1Complete	0.74
PTM_CreateEndorsementKeyPair	3.40
PTM_ReadPubEK	0.96
PTM_TakeOwnerShip	1.60
PTM_LoadKey2	1.48
PTM_GetPubKey	0.78
PTM_Sign	94.64
PTM_Quote	89.75
PTM_Extend	0.82
PTM_SHA1CompleteExtend	1.01
PTM_GetRandom	0.83
PTM_ReadPCR	0.77

5. CONCLUSION AND FUTURE WORK

The concept of portable TPM (PTM) based on USB Key and trust model of PTM are given in this paper. We use PTM to establish a trusted path between the verifier and the user in remote attestation. In our new trust model the single point failure of TPM and frequent sensitive data migrations are avoided based on PTM. The test results show that the PTM scheme is feasible. In the future, we will further research trusted path and platform management protocols and improve the performance of PTM.

6. REFERENCES

[1] Trusted Computing Group, Trusted Platform Modules Strengthen User and Platform Authenticity, http:// www.trustedcomputinggroup.org , 2005

[2] Jonathan M. McCune, Adrian Perrig, Arvind Seshadri, Leendert van Doorn. Turtles All The Way Down: Research Challenges in User-Based Attestation, *Proceedings of the 2nd USENIX workshop on Hot Topics in Security* ,USENIX, Boston, USA, 2007

[3] Sun Microsystems Inc. Java Card Platform Specification 2.2.2. http://java.sun.com/javacard/specs.html, 2006.

[4] TCG, TPM Main Part 3 Commands, http:// www.trustedcomputinggroup.org , 2006

[5] Dawei Zhang, Peng Hu, Trusted e-commerce user agent based on USB Key, IMECS, Hongkong, China, 2008

[6] Lei Han, Jiqiang Liu, Dawei Zhang, A Portable TPM Scheme for General-purpose Trusted Computing Based on EFI , International Conference on Multimedia Information Networking and Security,IEEE, Wuhan, China, 2009