Trust of User Using U-Key on Trusted Platform

Shuanghe Peng[1,2], Zhen Han[2]
1. Computing Center, Beijing Information Science & Technology University, Beijing 100085, P.R.China
2. Research Center of Information Security Architecture, Beijing Jiaotong University, Beijing 100044, P.R.China
E-mail: shhpeng@sohu.com hz@computer.njtu.edu.cn

Abstract

Trusted Computing based on TPM can be viewed from the following several ways, i.e. trusted on user, trusted on platform, trusted on application and trusted between platforms. Even though trusted on user was mentioned in Trusted Computing architecture, it does not really address security from a user point of view, as the model is centered on the security of platform. User identification and authentication mechanism, are rather rudiment. Andreas Pashalidis and Chris J.Mitchell proposed a Single Sign on scheme using Trusted Platform in 2003, where platform Attestation Identity Key is used as user identity. User identity is bound to his/her trusted platform, which makes it inconvenient to users. Mobility and flexibility are not achieved. Based on the rule of separation of user and platform credentials, trust of user using U-Key technology on Trusted Platform is proposed in this paper. The proposed scheme can simplify the management of user and provide portability and flexibility to user.

1. Introduction

Trusted Computing [1] (TC) emerged widely on the world in 1999. The main goal of TC is to enhance the security of the current PC architecture to ensure the security of the whole information system. TC focuses on the assurance that an entity does behave in the expected manner through mechanisms of integrity measurement, storage, and reporting.

The core of TC is what called Trusted Platform Module (TPM), a built-in hardware chip. The TPM can be thought of as a smart card that is embedded on the system motherboard. It is part of the platform's booting process and is also integrated with the operating system. Although trusted on user is mentioned in TCG, the system authenticates a user by keys or IDs stored in TPM. The mobility and flexibility are not achieved.

So, in this paper, U-Key [2] is introduced here based on the rule of separation of user and platform credentials. Different hardware is used to make user and platform trusted. U-Key is best suited for user credential storage and the TPM is best suited for host credential storage.

2. Background and Related Work

The Trusted Computing Group (TCG) defines a set of specifications aiming to provide hardware-based root of trust and a set of primitive functions to propagate trust to application software as well as across platforms. Within each trusted platform is a trusted sub-system, which contains a TPM, a core root of trust for measurement (CRTM), and support software TSS (trusted platform support service). The TPM is a hardware security solution providing both on boot and operating system security. As defined by TCG, the trusted bootstrap stage is illustrated in Fig 1.Starting with the CRTM, there is a boot-strapping process by which a series of trusted sub-system components (including BIOS, MBR, OS Loader and OS Kernel) measure the next component in the chain (and /or other software components) and record the value in the TPM. By these means, each set of software instructions (binary code) is measured and recorded before it is executed.

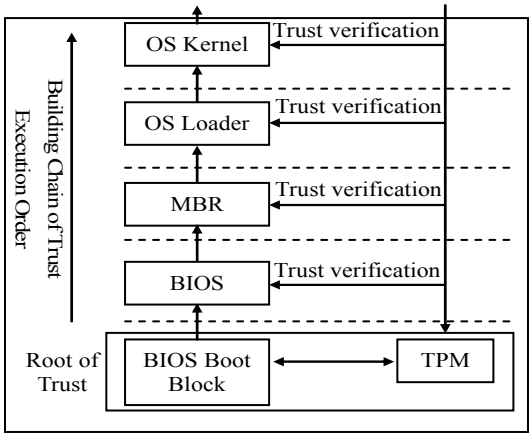

Figure 1. Trusted Bootstrap Defined by TCG

0-7803-9737-1/06/$20.00 ©2006 IEEE

The security of PC is composed of both user and platform. From the bootstrap process illustrated in Figure 1, TCG is only focus on the integrity of the platform. How to achieve trust of user is not mentioned.

In 2003, Andreas Pashalidis and Chris J. Mitchell proposed a TCG-Based SSO scheme [3]. The scheme uses TCG-specified credentials, called `Identity Credentials, as SSO identities. The integrity of the user platform's software state (which includes the AS) is measured by a TCG function called 'integrity Metrics'. Any given SP reliably acquires these software metrics through an 'Integrity Challenge /Response' session, also specified by TCG.

In this scheme, TPM identities acted as SSO identities for users. This will has some undesirable properties from the view of users. First, every TPM identity is bound to its TP. Second, from terms of security, it is critical that residual credentials do not remain on platforms during maintenance in order to protect them from a possible exposure. When user credentials are compromised they must be destroyed to be re-issued. Logical revocation is fine in most cases but very sensitive credentials might require more stringent measures, such as the physical destruction of the device holding them. This can easily be done with a removable device while it becomes more problematic with a device physically attached to the PC motherboard. Last, in order to simplify TPM administration, user and platform credentials can be managed by different authorities, such as Human Resources and MIS depending on company policies.

3. Trust of User using U-Key on Trusted Platform

A U-Key is a USB Token in a shape and size of a USB disk with built-in smart card [4], enabling secure storage and processing of sensitive data. This allows the user information and credentials, including digital certificates and private key to be securely stored only on the U-Key and nowhere else. Whenever a document or key must be signed or decrypted all cryptographic functions requiring the private key are performed by the microprocessor on the smart card. Using this method the private key never leaves the U-Key. This means that no third party can "listen" to the communication between the card and the reader to intercept the private key.

3.1 Trust of Booting User using U-Key

In order to achieve trust on booting user, the booting user must be authenticated. The method to authenticate booting user can be simple as set BIOS password, only legal user with correct password can boot the system. But the weakness of password-based authentication is well known. The highest security level can be achieved by means of a bidirectional authentication using a TPM chip for platform identification as well as user authentication in the form of a U-Key. So to improve the security, dual factor authentication using U-Key is introduced in this paper. Only legal user with correct U-Key and PIN can boot the system.

Since the authentication of booting user is executed at bootstrap stage, when Operating System has not been loaded, only BIOS code is running, so in order to communicate with U-Key at boot stage, a real mode driver of U-Key is needed. This real mode driver must use USB Host Controller Interface provided by BIOS directly. There are two such interfaces, one is Universal HCI (UHCI) specification [5] designed by Intel, the other is Open HCI (OHCI) specification [6]. Since Intel chip is used in most PC motherboard, UHCI 1.1 is selected to design the real mode driver.

After U-Key is introduced in the boot process, the trusted bootstrap is illustrated in Fig 2.

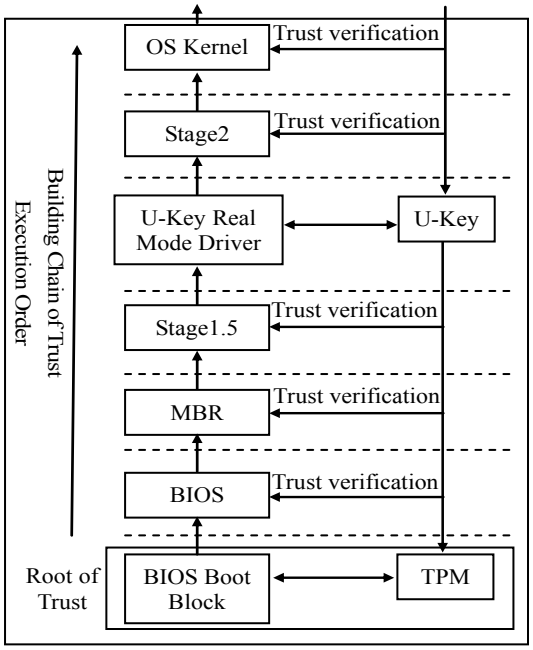

Figure 2. Trusted Bootstrap by Using U-Key

We select Grub [7] as the OS Loader. The real mode U-Key driver [8] is inserted between the code of Grub stage1.5 and Grub stage2. The integrity of the driver is checked by stage1.5, U-Key can be the root of trust for user after verification.

The verification consists of two type of authentication, one is the mutual authentication between the platform and U-Key, the other is user PIN verification.

When U-Key is inserted to the USB port on user platform, mutual authentication between U-Key and

the platform is executed. See Fig.3 for detail description of mutual authentication between U-Key and the platform.

After mutual authentication between the platform and U-Key, the process of user PIN verification starts.

If PIN verification succeeds, user can boot the system as illustrated in Fig. 2, otherwise, the process of bootstrap is halted.

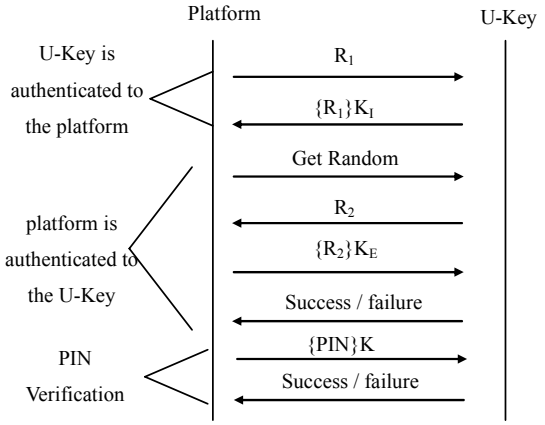

Figure 3. Mutual Authentication and PIN Verification

3.2 A Single Sign on Scheme Using U-Key

The proposed SSO scheme consists of two phases, one is initialization and the other is SSO phase.

The following set of credentials and the corresponding certificate authorities is presumed, if necessary, are available.

(1) TPM AIK pair ($PK_{TPM.AIK}$, $SK_{TPM.AIK}$). Generally the private part of an AIK is protected by a TPM with the storage root key (SRK), and the public key certificate is issued by a trusted third party such as a privacy CA. An AIK is flagged as non-migratable when created, since it must be tightly bound to a single platform.

(2)The AS key pair (PK_{AS}, SK_{AS}). Generally an AS is controlled and owned by a platform administrator. The private key of AS is protected by the TPM in the platform such that only the AS on this platform can use it (by checking the integrity value). The public key is in a certificate format signed by an AIK of the TPM.

(3) User key pair. We assume that each user has at least one identity key pair (PK_u, SKu). The concept of an identity of a user is similar to the AIK of a TPM. An AIK is flagged as non-migratable when created, since it must be tightly bound to a single platform. In contract, a user can generally access a number of platforms, such as a desktop and a laptop in his company, and a home PC. Therefore a user identity key is a migratable key. It can be created by the local TPM and protected by the TPM or it can be created and protected by another trusted device, such as U-Key. We use the latter method in the proposed scheme.

Just like the privacy CA for certificate of a platform AIK, a user can be certified by a trusted third party (an identity CA).

In order to make use of the security storage and cryptography function of the U-Key, a driver is needed. The Software Layer of U-Key is illustrated in Fig 4. It consists of the following modules:

(1) U-Key device driver. A USB driver is based on USB core of Operation System to transfer U-Key specified data between user space and kernel space.

(2) U-Key command service layer. This layer provides commands interface compatible to ISO7816 standard to U-Key service provider.

(3) U-Key service provider layer. This layer provides U-Key service such as initialization, mutual authentication, key pair generation and process of signature to application.

(4) Cryptographic Application Program Interface. The secure and trustworthy functions of the U-Key module are made available to application through cryptographic Application Programming Interface (APIs) compliant either to PKCS#11 standard [9] or to the MS CAPI specification. U-Key can then be used to enhance Operating System security policies or applications security plug-ins which take full advantage of the secure U-Key functions such as sealed storage, key generation, signature and encryption.

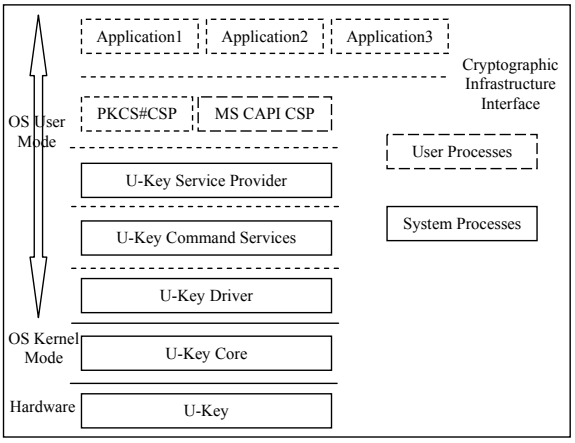

Figure 4. U-Key Software Layers

3.2.1 Initialization Phase

In this phase, initialization of the U-Key and the platform is needed.

(1) U-Key Initialization. Initialization of the U-Key is responsible for identifying the user, personalizing the U-Key personalization and linking the U-Key to the identified user based on confirmed identity of the user. User key pair (PK_u, SK_u) is created. The user is guided through a process to obtain a digital certificate from an identity CA. And the certificate is

stored in the U-Key. Besides, internal and external authentication keys (K_I and K_E) are loaded into the U-Key.

(2) Initialization of the Trusted Platform. As for trusted platform part, it includes the creation of platform AIK pair ($PK_{TPM.AIK}$, $SK_{TPM.AIK}$), platform AS key pair (PK_{AS}, SK_{AS}) and their credentials. AIK certificates are signed by privacy CA. AS certificate is signed by AIK. User authentication state is signed by SK_{AS}.

3.2.2 SSO Phase

Single sign on protocol is illustrated in Fig 5, the steps as follows:

(1) Resource request is sent from the user to the SP.

(2) SP challenges the AS for integrity and the authentication state of user.

(3) The AS checks the authentication state of user, if ok, then ship to step (7), otherwise, a prompt for user to insert U-key is given.

(4) Mutual authentication between U-Key and the Platform is executed. The process is just the same as illustrated in Figure 3.

(5) After the mutual authentication, a prompt for user to enter his/her PIN is given.

(6) User is authenticated to the U-Key by means of PIN, just the same as illustrated in Figure 3.

(7) The platform AS executes *Get User Credential* command to get credential from the U-Key. Whether or not the AS can get user credential from the U-Key is based on the user authentication state in step (6).

(8) SSO protocol between AS and SP starts. The integrity of the AS is guaranteed by TPM. The authentication state of the user signed by SK_{AS} is sent to SP.

(9) The SP verifies the signature of the user authentication assertion against the public key certificate of the PK_{AS}. If verification succeeds, the response to the resource request is return to the user.

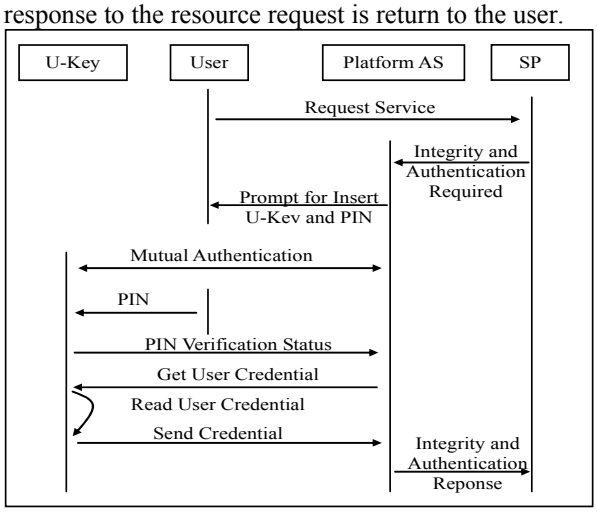

Figure 5. SSO Protocol Using U-Key

4. Conclusions

User security and PC security is two necessary aspects of Trusted Computing architectures. This paper establishes the distinct and complementary roles of TPM and U-Key in TP. Function concerned with user security is separated from TPM based on the strict separation between user and platform credentials. The storage and administration of user credentials is achieved by U-Key. Trust of bootstrap user and operation user is achieved by using U-Key on Trusted Platform. The proposed SSO scheme simplifies the administration of users and makes the user credential portability, which is convenient to users. Further more, the security of the system and application can be improved by the secure storage and processing capabilities of the U-Key.

References

[1]TCG Specification Architecture Overview. http://www.trustedcomputinggroup.org.

[2] Shuanghe Peng Weimin Tang and Yali Mou. The Implementation of T=0 Protocol of Smartcard based on USB Control Chip. Journal of Computer Applications, Vol. 24, October 2004.

[3] Andreas Pashalidis, Chris J. Mitchell, Single Sign-On Using Trusted Platforms. Proc. of Information Security: 6th International Conference, ISC 2003,Bristol, UK, October, 2003.

[4] The ISO 7816 Smart Card Standard Overview. http://www.cardwerk.com/smartcards/smartcard_standard_ISO7816.aspx

[5]Universal Host Controller Interface (UHCI) Design Guide Revision 1.1, March 1996.

[6] Open Host Controller Interface (OHCI) Specification for USB Release 1.0a, September 1999.

[7] GNU GRUB (GRand Unified Bootloader). http://www.gnu.org/software/GRUB/.

[8] Shuanghe Peng, Zhen Han. Enhancing the Security of PC with U-Key. IEEE Security & Privacy, September, 2006.

[9] PKCS#11 standard. http://www.rsasecurity.com/rsalabs/pkcs/pkcs-11/