http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 9 (4): 730-739, 2010
ISSN 1812-5638

A Novel Trusted Terminal Computer Model Based on Embedded System

Xi Qin, Chaowen Chang, Changxiang Shen and Li Gao
Institute of Electronic and Technology, Information Engineering University, 450004,
Zhengzhou, Henan, People's Republic of China

Abstract: In this study, a novel trusted terminal computer model based on embedded system is proposed, which aimed at building secure and trusted computing environments for terminal computers. The model can tolerate untrusted components being on the trusted computer, which doesn't need change hardware structure and operating system on the legacy terminal system. Taking advantage of embedded system and virtual machine, the model assured untrusted components being on the terminal computer can't cause serious information security threats, implement isolation of missions and manage I/O ports and communication resources on the terminal platform so that, the results of actions on it are expected and controlled. Compared with other trusted computing platforms, we can conclude that the trusted terminal model has virtues of permitting untrusted component being on the model, convenience of deploying applications, not needing TPM hardware and supporting legacy system.

Key words: Trusted computing, integrity measurement, TPM, untrusted components, embedded system

INTRODUCTION

With the rapid development of information technology and network, security and efficiency (Chang *et al*., 2010; Asfand-e-Yar and Sher, 2004) of network always act as very important roles. On the open network environment, the security problem became more important. Because, the resources on the terminal computers are open and existing trojans, viruses, system backdoors, hidden channels or malicious events, the security of terminal computers became more and more important. The simplified structure of PC's hardware and software caused the resources on the computer can be used without necessary control, especially the execute codes can be modified and malicious programs can be running. Moreover, the structure doesn't strictly perform access control for validated users accessing resources. These problems can't be solved completely with traditional security technology. The best way is to establish the high level defense mechanism for the platform of terminal computer, not only depending on software to protect the security of the whole information system.

This study proposed a novel trusted terminal computer model based on embedded system tolerating untrusted components, which doesn't need change hardware structure and operating system on the legacy terminal system. Taking advantage of embedded system and virtual machine, the model assures untrusted components being on the terminal computer can't cause serious information security threats, implement isolation of missions and manage I/O ports and communication resources on the terminal platform so that, the results of actions on it are expected and controlled. The model can build secure and trusted computing environments for terminal computers.

The representative structures of existing trusted computing platforms includes the TCG trusted computing platform (Trusted Computing Group, 2007) the Next-Generation Secure Computing Base (NGSCB) (Microsoft Corporation, 2003) etc.

In the TCG platform, the core of trusted computing is a security chipset called TPM (Trusted Platform Module) (Trusted Computing Group, 2006), which is a small SOC (System On Chip) including cryptography engines, non-volatile storage, I/O and platform configuration register, etc. The TPM enable the computer protect the computing platform based on hardware. Based on TPM, the computer can build the root of storage, the root of measurement and the root of report. These roots are the basic to implement transitive trust (trust chain) and build trusted computing platform.

Figure 1 shows the trust transfer processing on the terminal computer which includes measurement flow and execution flow. The root of trust gives a trustworthy description of a second group of functions. Based on this description, an interested entity can determine the trust it is to place in this second group of functions. If the

Corresponding Author: Xi Qin, Institute of Electronic and Technology, Information Engineering University, 450004, Zhengzhou, Henan, People's Republic of China

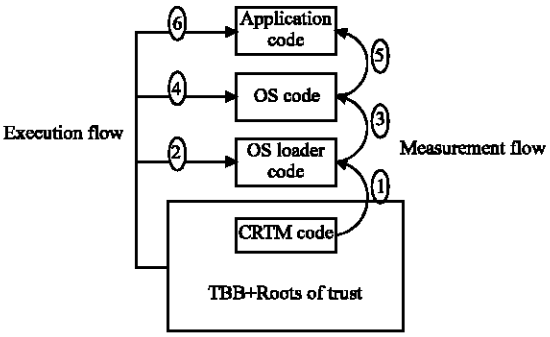

Fig. 1: Transitive trust applied to system boot from a static root of trust

interested entity determines that the trust level of the second group of functions is acceptable, the trust boundary is extended from the root of trust to include the second group of functions. In this case, the process can be iterated. The second group of functions can give a trustworthy description of the third group of functions, etc. Transitive trust is used to provide a trustworthy description of platform characteristics.

Figure 1 shows that trust chain is applied to a system booting from a static root of trust and the trust boundary is extended to include code that didn't natively reside within the roots of trust. In each extension of the trust boundary, the target code is first measured before execution control is transferred.

The TCG trusted computing platform includes TPM, CPU, operating system, application software and network infrastructure, etc. The structure of the PC's trusted computing platform (Trusted Computing Group, 2007) is shown in Fig. 2.

The structure divides into three levels: TPM, TSS (trusted software stack) and application software.

The TPM provides the security supports for the operating system and TSS.

The functions of TSS include providing the only entry for the applications using the function of TPM, permission of synchronization access to TPM, managing the resources of TPM and releasing them in proper time. TSS includes three parts: TDDL (TCG Device Driver Library), TCS (TSS Core Services) and TSP (TCG Service Provider).

The TDDL provides user mode interface. Such an interface has several advantages over a kernel mode driver interface:

- It ensures different implementations of the TCG software stack properly communicate with any TPM
- It provides an OS-independent interface for TPM applications

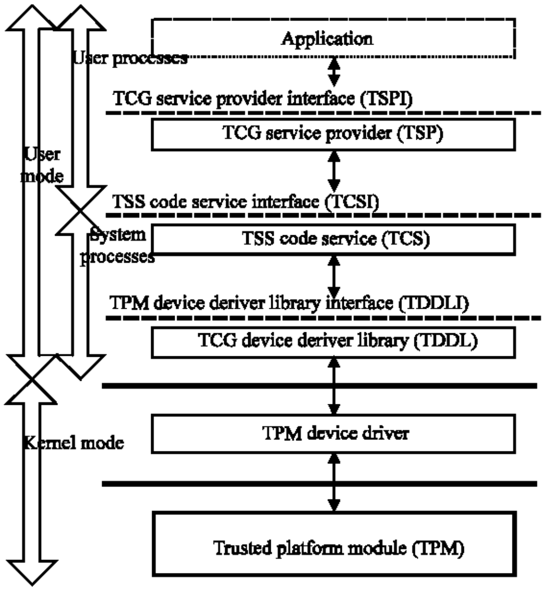
Application
User processes
TCG service provider interface (TSPI)
TCG service provider (TSP)
User mode
TSS code service interface (TCSI)
System processes
TSS code service (TCS)
TPM device deriver library interface (TDDLI)
TCG device deriver library (TDDL)
Kernel mode
TPM device driver
Trusted platform module (TPM)

Fig. 2: The structure of TCG platform

- It allows the TPM vendor to provide a software TPM simulator as a user mode component

The TDDL provides the transition between user mode and kernel mode. It does not manage threaded interactions with the TPM, nor does it perform TPM command serialization. These are applied higher in the stack. Since, the TPM is not multithreaded, there would be a single-instance of the TDDL, per platform and it enforces single threaded access to the TPM.

The TCS provides an interface to a common set of platform services. Though, there may be multiple TCG service providers on a single platform, the TCS ensures they all exhibit common behavior. The TCS provides 5 core services:

- Context management-implements threaded access to the TPM
- Credential and Key Management-Stores credentials and keys associated with the platform
- Measurement Event Management-Manages event log entries and access to associated PC registers
- Parameter Block Generation-responsible for serializing, synchronizing and processing TP commands

The TCS operates as a system process in user mode. It is trusted to manage authorization information supplied to the TPM.

The TSP exposes a C interface to the TPM, based on an object oriented underlying architecture. It resides within the same process address space as the application.

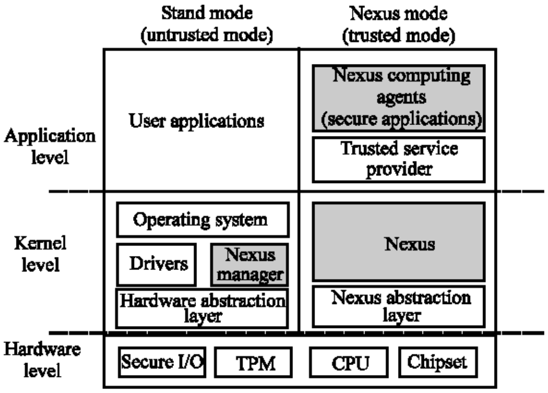

Fig. 3: The structure of NGSCB platform

Authorization takes place at this layer, either using a user interface coded to this layer or via a callback mechanism at the TCS layer (if the caller is remote). In order to provide, a consistent authorization interface to the end user, local applications do not provide authentication services, but rather rely on that inherent in the platform.

The TSP provides two services, context management and cryptography. The context manager provides dynamic handles that allow for efficient usage of application and TSP resources. Each handle provides context for a set of interrelated TCG operations. Different threads within the application may share the same context or may acquire a separate context per thread.

To make full use of TPM protected functions, supporting cryptographic functions must be provided. The TSP does not provide this support except as is necessary to perform operations required by the specification. In particular, bulk data encryption is not exposed by the interface. Example functions include message digesting and byte-stream generation.

The NGSCB (Microsoft Corporation, 2003), formerly known as Palladium, is a software architecture designed by Microsoft which is expected to implement parts of the controversial Trusted Computing concept on future versions of the Microsoft Windows operating system. NGSCB is part of Microsoft's Trustworthy Computing initiative. Microsoft's stated aim for NGSCB is to provide a high security environment in data security, person privacy and system integrity for computer users. The environment can make a security connection among applications, hardware, memory, storage and I/O, form a filter net between software and hardware, any execute codes before submitted to hardware must be checked by NGSCB, protecting the system against malicious attack. The structure of NGSCB is shown in Fig. 3.

The NGSCB relies on hardware technology designed by members of the Trusted Computing Group (TCG). The hardware includes TPM, CPU, secure I/O and related chipset, which provide a number of security-related features to protect the system against retrieve attack, even to the machine's owner. The TPM will provide secure storage of cryptographic keys and a secure cryptographic co-processor and a curtained memory feature in the Central Processing Unit (CPU).

In NGSCB, there are two work modes: stand mode and Nexus mode. The modes split the whole system into two distinct parts. In Nexus mode, there are two software components, the Nexus, a security kernel that is part of the Operating System and Nexus Computing Agents (NCAs), trusted modules within NGSCB-enabled applications. In Nexus, all running software and data be protected strictly in sealing storage, protected executing and protected inputs and outputs, etc. Any code which deals with NGSCB functions must be located within the NCA.

The concept of trusted computing proposed by TCG has provided an effective approach for building trusted computing platform of terminal computer. However, the integrity managements of platform and components (such as these integrity references values publishing, management and updating) have faced huge challenges and troubles. The proof of trustworthy evaluation has much different between in theory and in application. According to TCG, the components running in the terminal computer must be measured and evaluated, the proof of evaluation is that trusted measurement modules (agency) compares the results of integrity measurement values of components with those reference values, if the values are equal, then the result is trusted and the terminal computing environment is trusted and secure.

Based on those viewpoints, if a component is untrusted, which means the component's integrity measurement result is not expected, so the component can't be booted or loaded. However, an untrusted component doesn't mean it is tampered or malicious (such as, the updating version of an old component or adding a new component in system). In this way, an untrusted component may be trustworthy. The way to judge a trusted component of TCG badly limits the applications on trusted computing platform. If an updated component or a new component wants to run on the platform, it must be modified its integrity reference value in TPM by the TPM vendors providing corresponding mechanism. However, the process is long and complex. In other words, loading applications will be limited on the trusted computing platform.

The NGSCB splits the computing platform into two parts: the stand mode and the Nexus mode, providing a high security running environment for users. However, this scheme not only need the support of trusted hardware (such as TPM, CPU), but also the cooperation of trusted operation system, trusted software and trusted I/O. It's impossible to build a trusted computing environment on legacy platform while doesn't change legacy hardware and software.

A TRUSTED TERMINAL MODEL BASED ON EMBEDDED SYSTEM

Considering, the actual applications on open environment, it can't avoid the untrusted components being on terminal computer. The aim of this study was to assured that the untrusted components will not cause serious information security threats and the results are expected and controlled on the trusted terminal model. At the same time, when building the trusted terminal, the key problem is to reduce the limitation to the legacy operating system and applications, support legacy applications without modifying them. Based on the aim, this study proposed the trusted terminal model based on embedded system which can tolerate untrusted components.

Because, information host transmission channels (such as network, bluetooth, infrared and other communication methods) and storage mediums (such as U-disk, hard disk, CD-RAM) are important information transmit ways where the trojans, viruses, system backdoors, hidden channels or malicious events can be hided inside. We should take effective technical approaches to built computing environment where the results are expected and controlled.

The structure of trusted terminal model: The structure of trusted terminal model includes four levels: the hardware level, the trusted virtual machine level, the operating system level and the application level, which is depicted in Fig. 4.

The hardware level: The hardware level consists of the basic hardware of the computer and an embedded system. The embedded system is a SOC (system on chip) which includes embedded CPU, TPM, secure Linux operating system, etc. The embedded system can be treated as the root of trusted to provide the functions of trusted network connection, virtual TPM service and the security management and control for upper computing levels. The embedded system will assure the transmission, distribution and storage of information meet the defined security policies, specifications and protocols. Although, the untrusted components had caused some events and actions which influenced the security information, they should be controlled in the aspects of transmission, distribution and storage, the results of information security will not be changed actually. Finally, we can implement the trusted terminal computer platform tolerating the untrusted components.

Trusted virtual level: The virtual machine level implements the isolation of missions based on virtual memory. This level manages all the I/O resources on the terminal platform, only permits the peripheral devices satisfied security policy communicate to the terminal computer, protect the sensitive information against illegal revealed. This level manages all communication resources during provide the service of virtual Ethernet adapter, guarantees the unique logical export of information communication.

The most important part in virtual level is VM (Virtual Machine). The VM (Barham *et al*., 2003; Rosenblum and Garfinkel, 2005; Dunlap *et al*., 2002; Garfinkel *et al*., 2003) is a system which can support multiple operating system and running on single physical server. It can provide more effective usage to bottom hardware. Take advantage of the VM's function of isolation missions, we can solve two questions: the one is to manage the upper level using hardware resources, the other is to increase the stability

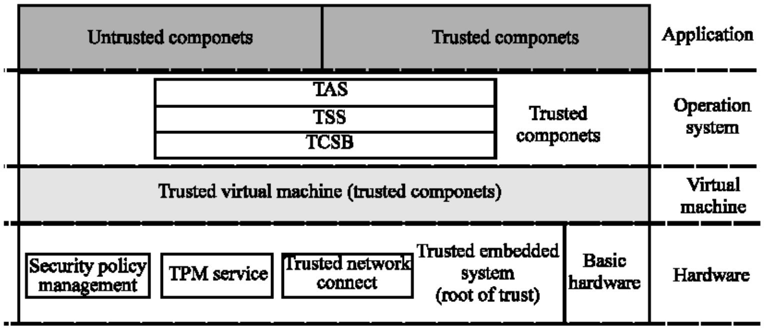

Fig. 4: The structure of trusted terminal model

of applications. Under the control of VM, the operation system and applications access the hardware resources. Based on it, our model can control the actions of the untrusted components.

The trusted embedded system measures the integrity of the VM, to assure the VM is trusted.

Trusted operating system level: This level consists of TCSB (Trusted Computing Software Base) allowing white list mechanism, TSS (Trusted Computing Base Support Service System Service) and TAS (Trusted Computing Base Support Service Application Service). TCSB is loaded after the embedded system has measured its integrity. The security mechanisms of TCSB include the control of processes, the connection of VPN, the filter of packages, security agency and events audit. The TCSB is independence and doesn't be bypassed.

The process control mechanism of TCSB will measure the integrity of the operating system services and applications to prove the processes are integrate and trusted. The TCSB support the white list mechanism, which the services and applications which are on the list can be booted and loaded without integrity measurement. In this way, the untrusted components can be tolerated to loaded and executed.

The VPN connection and filter network packages mechanisms of TCSB can ensure to effectively control and manage the malicious actions of the untrusted components with the help of isolation missions, I/O ports and devices control on the virtual level.

The TSS is up to the TCSB, based on the embedded system, provides the function interfaces for applications, including cryptography keys, certificates of TPM and platform integrity management.

The TAS is up to the TSS, provides applications interface for users, including integrity protection, trusted attestation and information security protection.

Application level: This level consists of all kinds of application programs, including untrusted applications.

Building trusted chain: The trusted terminal model is based on the expanding the trust chain and ensure the trusted components on each level in the model are not be tampered. The process of expanding the trust chain is shown in Fig. 5.

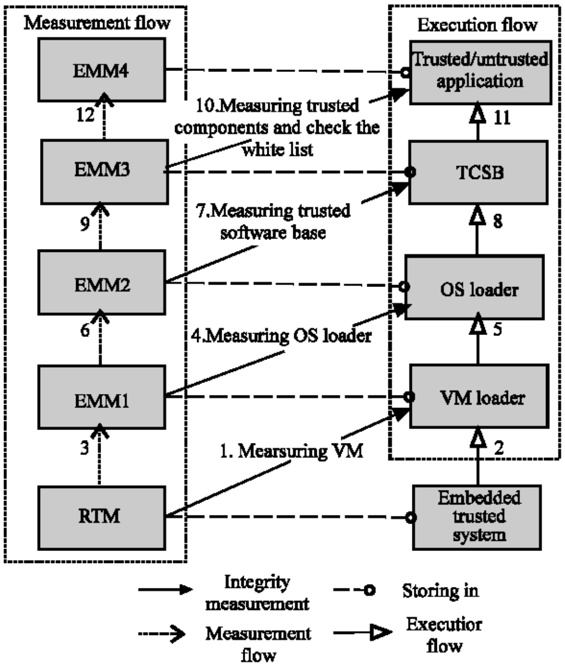

Fig. 5: The trusted chain of trusted terminal model

The virtual TPM service of embedded system is act as the RTM (root of trusted measurement) of the computing platform, which measures the VM loader, if the value of measurement equals the expected value, then the measurement is success. The embedded system loads the VM, at that time, the trust chain expands to the EMM1 (Extended Measurement Module), which is stored in the flash of the embedded system, the flash cannot be modified by users.

The EMM1 measures the OS loader, loads it if success, then trust chain expands to EMM2, which is stored in OS loader.

The EMM2 measures the TCSB, loads it if success, then trust chain expands to EMM3, which is stored in TCSB.

The EMM3 measures the applications, loads it if success. But when the measurement result is failure, if application is on the whist list, it can also be permitted to boot and load normally.

In actually, the process to build the trust chain is the process to measuring the trustworthy of components. The embedded system is act as the RTM, when each platform control right transmitting, the trustworthy is expanding to next component through integrity measurement and verify. In the process of expanding the trustworthy, the expected values of components can get from certificates or references of platform measurement values. The list of expected values extended into the trusted embedded system by a reliable way.

The approach of trusted measurement and loading the components

The analysis of existing integrity measurement models: On the process of building trusted chain, the transmission of control right depending on the integrity measurement of entities. The typical integrity measurement model include: IMA (Integrity Measure Architecture) model (Sailer *et al*., 2004; Jaeger *et al*., 2006), SB (Secure Bus) model (Zhang *et al*., 2007), PBA (Property-based attestation) model (Sadeghi and Stuble, 2004).

The IMA based on LSM (Linux Secure Module) uses hook functions in the kernel data structure and measures the entities during hooks capturing the measurement events and measuring them. The architecture of IMA measures the execute programs, dynamic loader, kernel modules and dynamic libraries in different measurement ways. During inserting into measurement points on the file_mmap LSM hook and process of Load_module£¬the IMA could start the measurement function measure() and trigger measurement processes when the computer system invokes insmod(), execve(), init_module and script interpreter.

The SB model based on the trusted hardware uses the TCG's TPM and Intel's LT or AMD's SEM technology and provides a SK (secure kernel) and a SB (secure bus). With the help of the LT or SEM, the SK could provide isolation function between processes and applications on the computer system. The SB is on the middle of the operation system kernel space and use space, it can allocate isolate memory space for processes. Although the isolation mechanism of SK can protect the running codes against modified, but attackers could modify them before loading these codes. The SB avoids the attack by hash the inputs and outputs of processes. The codes of processes before running is digested by SB, the digested data is signed by SB. The codes of processes, digests and signature data are all sent to verifier. The verifier according to the signature and certificates to judge these data whether trusted or not.

The PBA model is an attestation method based on property which provided by German (Sadeghi and Stuble, 2004) . The PBA attest the platform security based on the property certificates of platform. Because the PBA doesn't verify the configure information of software and hardware of platform, it can avoid leak the information of software and hardware and expands the range of trusted measurement. But the property itself is an abstract concept, the measurement, comparation and verification are difficult.

The measurement of trusted terminal model: On the trusted terminal model, the booting measurement process of system platform of the model has described in earlier so, we describe the loading measurement process of trusted components and application components.

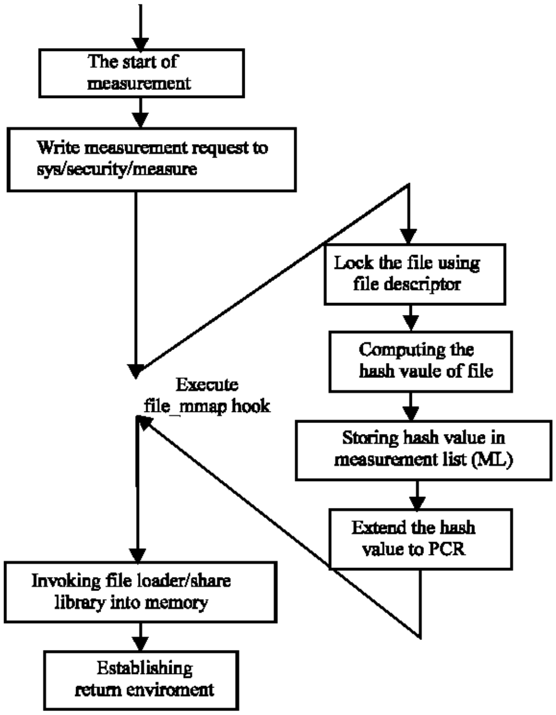
The start of measurement
Write measurement request to sys/security/measure
Lock the file using file descriptor
Computing the hash vaule of file
Execute file_mmap hook
Storing hash value in measurement list (ML)
Extend the hash value to PCR
Invoking file loader/share library into memory
Establishing return enviroment

Fig. 6: The loading measurement process of an executing file

The loading measurement process of trusted components: The trusted components run on the virtual machine, so we use IMA model to measure the loading process of trusted components.

The trusted components includes execute files, kernel modules and dynamic libraries. We insert the measurement points on the file_mmap LSM hooks and load_module process and when the system invokes insmod(), execve(), init_module, the measurement of trusted components begins. Figure 6 shows the loading measurement process of an executing file based on file_mmap LSM hook.

On the loading process, hashes the file and extends the hash value to the PCR (platform configure register) of TPM and maintains a Measure List (ML) in the system kernel.

The loading measurement process of application components: The measurement mission of application components is finished by TCSB. The process control mechanism of TCSB measures the integrity of operation system services and applications, ensures the integrity and trustworthy of components. The trusted terminal model supports white list mechanism on the TCSB, the components on the white list can be booted without integrity measurement, which can tolerate the untrusted components loading and running.

The loading processing of application components can be described as follows:

```
If Hash(Component) = references then //trusted measurement
    Load( Component ); //trusted load
  Else
  If (Component.Type <> OS-kernel ) and (Component IN white-list)
  Then
    Load( Component ); //white list load for untrusted component
  End IF
  END IF
```

Type of components presents OS-kernel, System Service, Application, etc. The important components (such as process control module, VPN client module, network filter module, security agency module and audit module) which relate to the system security all belong to OS-kernel.

That is to say, the system kernels and important security components must be trusted loaded, only defined non-kernel level applications and services can be loaded by white list load.

THE ANALYSIS AND DISCUSSION ON THE TRUSTED TERMINAL PLATFORMS

The TCG computing platform doesn't permit the untrusted components running on the platform. if a component is untrusted, which means the component's integrity measurement result is not expected, so the component can't be booted or loaded. However, an untrusted component doesn't mean it is tampered or malicious (such as the updating version of an old component or adding a new component in system). That is to say, an untrusted component may be trustworthy. The way to judge a trusted component of TCG badly limits the applications on trusted computing platform. If an updated component or a new component wants to run on the platform, it must be modified the integrity reference value in TPM by the TPM vendors through providing corresponding mechanism, the process is long and complex. So, loading untrusted applications will be limited on the trusted computing platform.

The NGSCB computing platform provides a high security running environment for users. However, the users must install signatured hardware and software, such as NGSCB security chipset, security OS and security applications. These signatured hardware and software is the proof of trustworthy. The system controls the actions on the platform according to the relationship of trustworthy. If users don't trust the computing platform or the computing platform itself don't trust the entities on the platform, the usage to the computing system will be limited badly, such as user can't open the files on the disk if the user is untrusted, some devices refused to boot or run if the application is untrusted.

Table 1: The comparing table of different trusted terminal platforms

Contents	NGSCB	TCG	Proposed model
Permitting untrusted component	Yes	No	Yes
Convenience of deploying applications	No	No	Yes
Needing TPM hardware	Yes	Yes	No
Permitting legacy system	No	No	Yes
Types of components on system	Signatured chipset, OS, applications	TPM hardware and TSS software stack	Only embedded system

Compared with the above trusted computing platforms, the trusted terminal model based on embedded system tolerating untrusted components has some advantages as follows:

- The trusted terminal system permits untrusted components being on the system, so deploying the applications will be more convenient and flexible
- The embedded system itself is a independence SOC so that the system won't be influenced and controlled by upper operating system. Even if the operating system has bugs or hidden channels, it won't actually damage the system security
- The embedded system has the function of TPM so that the implementation of the trusted system will be easier
- The virtual machine on the model can implement the isolation of missions and control the hardware resources transparently

The comparing table of different trusted terminal platforms is shown in Table 1.

THE DESIGN AND IMPLEMENTATION OF TRUSTED TERMINAL SYSTEM BASED ON USB EMBEDDED SYSTEM

The structure of the trusted terminal system: Because of the lack of TPM on the legacy system of terminal computer, the trusted computing environment can't be built based on it. Considering the ubiquitous ability of USB interface, our trusted terminal system can be built based on USB embedded system. The terminal system consists of USB embedded system level, virtual box VM level, windows operating system level and applications level. The structure of the system is shown in Fig. 7.

USB embedded system level: The USB embedded system level provides the trusted support functions of trusted

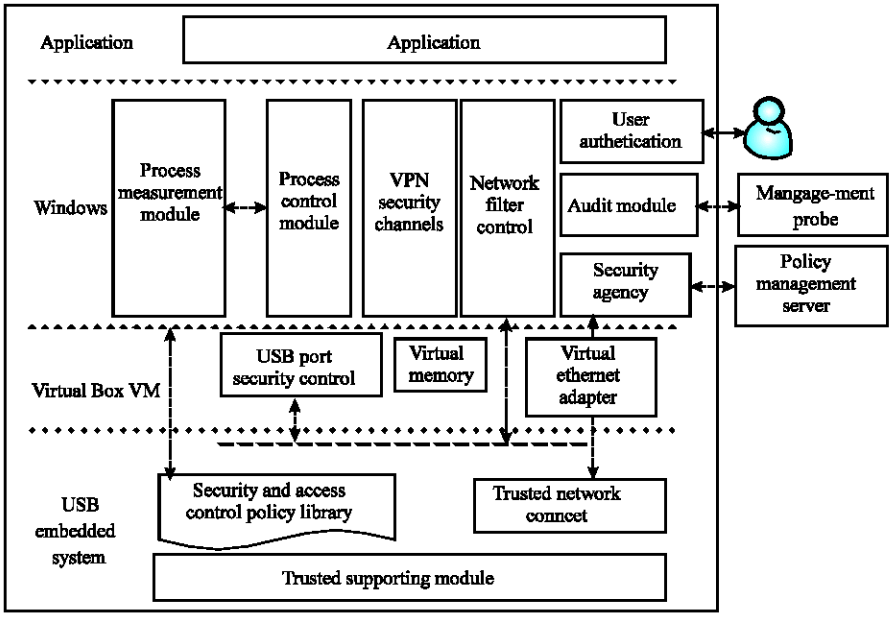

Fig. 7: The structure of trusted terminal system

easurement, trusted storage, trusted report and trusted network connection to upper levels. The embedded system provides the platform attestation for remote network, according to the result of remote attestation the system decide whether the terminal can access the network.

Virtual box VM level: The virtual box VM level can isolate missions through virtual memory. The USB port security control module can control the mobile storage devices based on security policy, assuring sensitive information can't be leaked. The virtual ethernet adapter manages all the communication resources to guarantees the unique logical export of information communication.

Windows level: The process control module in windows level can control the calls of system processes, before the boot of the process the module measures the integrity of the process, according to the measurement result to decide the process whether can be booted.

The network filter control module analyses all network access requests, which meeting security policy can be sent, otherwise forbidden.

Based on virtual ethernet adapter, the VPN security channel module can act as IPsec VPN client, build security channel with IPsec access gateway. Virtual ethernet adapter ensures all information communication on the channel be encrypted, even if the network filter module is bypassed, the sensitive information can't be leaked.

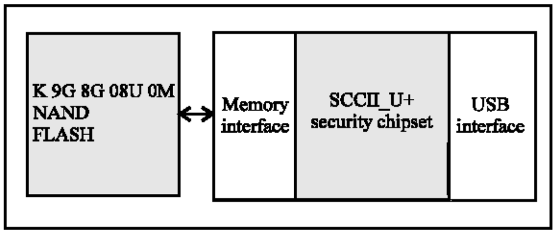

Fig. 8: The physical structure of the USB embedded system

The user authentication module based on the USB embedded system to verify the user identity and right. The audit module audits all actions, operations of terminals, submits audit information to management probe, forbids the users except audit administrator accessing the audit information.

The security agency module communicates with policy management system server, downloads the security polices and submits the audit information.

The USB embedded system: The USB embedded system is a SOC (System On Chip), it mainly consists of TPM security chipset, flash chipset and linux operating system, the physical structure of the embedded system (AisinoChip, 2009) is shown in Fig. 8.

The security chipset is a 32 bit chipset with a USB interface, it is compatible with ARM9 commands collection. There are cryptography algorithm engine, random generator and security protect units on the chipset. The chipset can provide the functions of TPM and cryptography. At the same time, it manages and controls the security policy.

The flash connects the security chipset with storage interface, it provides non-volatile storage under the control of the security chipset.

The USB embedded system communicates to the upper levels with USB interface, provides communication and security services described above. The logical structure of the embedded system is shown in Fig. 9.

The whole system is divided into three districts: policy management and control, TPM and non-volatile storage.

The TPM consists of the core root of trusted measurement, root of trusted storage and root of trusted report, PCR and cryptography algorithm engine.

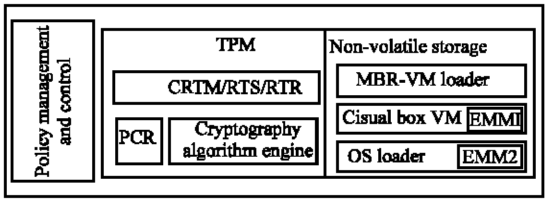

Fig. 9: The logical structure of the USB embedded system

The non-volatile storage includes VM loader, OS loader and Virtual box loader. The storage area can't be modified. VM loader is on the MBR (Main Boot Record) area, EMM1 is on the virtual box and EMM2 is on the OS loader.

The loading process of USB embedded system: The loading process of USB embedded system is shown in Fig. 10.

- Before the computer running, it loads the USB embedded system first and then cleans the MBR on the USB storage area. After the embedded system measured the VM loader, it releases the VM loader to the USB storage area and replaces the legacy MBR in USB
- MBR verifies user PIN password, loads the virtual box and loads the virtual devices, such as virtual encrypt storage, virtual memory, virtual ethernet adapter and port control, at the same time, boots EMM1 (Extended Measurement Module for virtual box)
- After virtual box booted successfully, EMM1 measures OS kernel and related kernel modules, then it releases OS loader to virtual box and replaces the legacy MBR in local hard disk

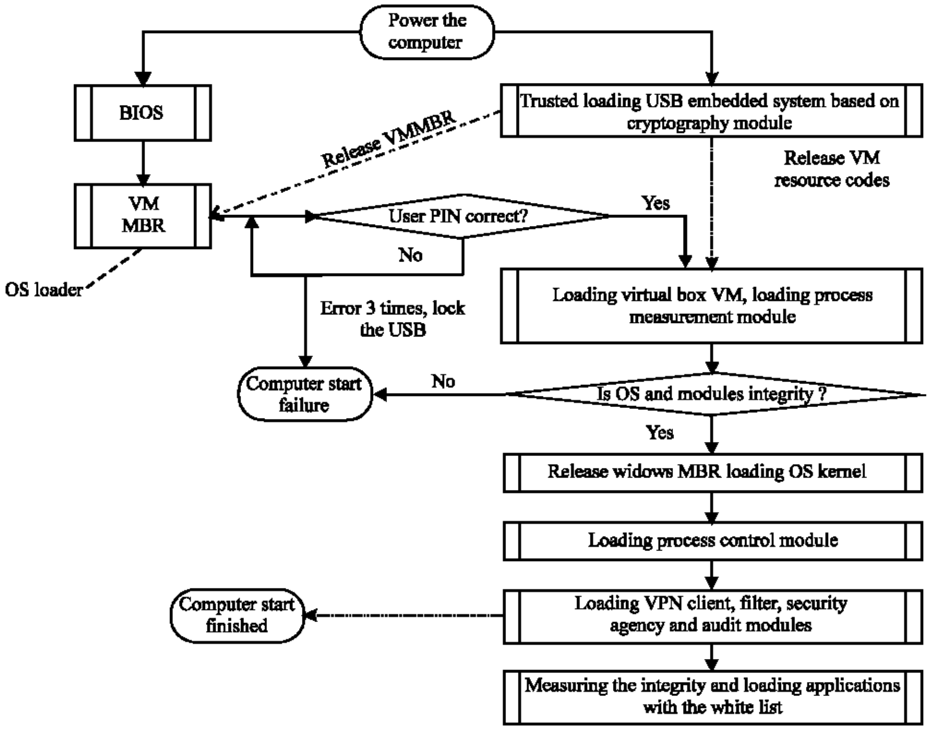
Power the computer
BIOS
Trusted loading USB embedded system based on cryptography module
Release VMMBR
Release VM resource codes
VM MBR
User PIN correct?
Yes
No
OS loader
Error 3 times, lock the USB
Loading virtual box VM, loading process measurement module
Computer start failure
No
Is OS and modules integrity ?
Yes
Release widows MBR loading OS kernel
Loading process control module
Computer start finished
Loading VPN client, filter, security agency and audit modules
Measuring the integrity and loading applications with the white list

Fig. 10: Loading process of USB embedded system

- OS loader loads OS kernel and related kernel modules, then boot EMM2 (Extended Measurement Module for OS loader)
- EMM2 measures TCSB which includes the process control module, VPN client module, network filter module, security agency module and audit module
- TCSB loads its modules and then boot EMM3 (Extended Measurement Module for TCSB)
- EMM3 measures and loads the applications. The process of loading includes the trusted loading and untrusted loading based on white list

CONCLUSION

A trusted terminal computer model based on embedded system which can tolerate untrusted components doesn't need changing hardware structure and operating system on the legacy terminal system. The terminal computing platform permits untrusted components being on system, improves the convenient and flexibility of deploying applications. The embedded system on the model reduces the relying on the legacy OS, not only resolves the problem of absence of TPM hardware on the terminal, but also enhance the security of the system. The virtual machine on the model can manager the hardware resource, implement access control, security storage, isolation missions. This study describes the implementation of trust chain on the model and based on USB card builds a trusted terminal system. Compared with other trusted computing platforms, our trusted terminal model has virtues of permitting untrusted component being on the trusted model, convenience of deploying applications, not needing TPM hardware and supporting legacy system.

ACKNOWLEDGMENTS

This study was supported by the National High Technology Research and Development Program of China (863 Program) under grant No. 2007AA01Z479.

REFERENCES

AisinoChip, 2009. Aisino product datasheet. http://www.aisinochip.com/product/sccii.html.

Asfand-e-Yar, M. and M. Sher, 2004. Secure route path formation in Ad hoc On-demand distance vector routing protocol. Inform. Technol. J., 3: 142-145.

Barham, P., B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho and R. Neugebauer *et al*., 2003. Xen and the art of virtualization. Proceedings of the 19th ACM Symposium on Operating Systems Principle, Oct. 19-22, Bolton Landing, USA., pp: 164-177.

Chang, C., L. Gao, H. Kou, L. Gao, X. Liu and Z. Qin, 2010. An integrity batch report scheme based on the waiting stack. Inform. Technol. J., 9: 79-88.

Dunlap, G.W., S.T. King, S. Cinar, M.A. Basrai and P.M. Chen, 2002. ReVirt: Enabling intrusion analysis through virtual-machine logging and replay. Proceedings of the 5th Symposium on Operating Systems Design and Implementation, Dec. 9-11, Boston, Massachusetts, pp: 211-224.

Garfinkel, T., B. Pfaff, J. Chow, M. Rosenblum and D. Boneh, 2003. Terra: A virtual-machine-based platform for trusted computing. Proceedings of the 19th ACM Symposium on Operating Systems Principles, Oct. 19-22, Bolton Landing, NY, USA., pp: 193-206.

Jaeger, T., R. Sailer and U. Shankar, 2006. PRIMA: policy-reduced integrity measurement architecture. Peoceedings of the 11th ACM Symposium on Access Control Model and Technology, June 7-9, Lake Tahoe, California, USA., pp: 19-28.

Microsoft Corporation, 2003. Next-generation secure computing base. http://www.microsoft.com/resources/ngscb/default.mspx.

Rosenblum, M. and T. Garfinkel, 2005. Virtual machine monitors: Current technology and future trends. IEEE Comput., 38: 39-47.

Sadeghi, A.R. and C. Stuble, 2004. Property-based attestation for computing platforms: Caring about properties, not mechanisms. Proceedings of the 2004 Workshop on New Secruity Paradigms, Sept. 20-23, Nova Scotia, Canada, pp: 66-77.

Sailer, R., X. Zhang, T. Jaeger and L. van Doorn, 2004. Design and implementation of a TCG-based integrity measurement architecture. Proceedings of the 13th USENIX Security Symposium, Aug. 9-13, San Diego, CA, USA., pp: 223-238.

Trusted Computing Group, 2006. TPM main part 2 TPM structures. http://www.trustedcomputinggroup.org/files/resource_files/8D3D6571-1D09-3519-AD22EA2911D4E9D0/mainP2Structrev103.pdf.

Trusted Computing Group, 2007. TCG specification architecture overview. http://www.trustedcomputinggroup.org/files/resource_files/AC652DE1-1D09-3519-ADA026A0C05CFAC2/TCG_1_4_Architecture_Overview.pdf.

Zhang, X., M.J. Covington, S. Chen and R. Sandhu, 2007. SecureBus: Towards application-transparent trusted computing with mandatory access control. Proceeding of the 2nd ACM Symposium on Information, Computer and Communication Security, Mar. 20-22, Singapore, pp: 117-126.