Research on Key Technologies of Trusted Enhancement in Embedded Systems

Xinlai Dang [a], Fei Wang [b] and Jie Qiang [c]
Space Engineering University, Beijing 101400, China.
[a]296869149@qq.com, [b] Wangfei791009@163.com, [c] easyrava@163.com

Abstract. Aiming at the threats faced by embedded systems, using trusted computing technology to solve embedded security equipment problems is a feasible and efficient security solution. Under the premise of not changing the existing mobile device hardware architecture, a new secure boot mechanism of the embedded system is proposed, and the SD card is used as an external trusted platform module. Construct a complete trust chain from the Boot loader to the upper application. The starting point of the trust chain is stored in the SD card. The metric value of each component in the startup process is stored by the key signature in the SD card. And the implementation process of the mechanism is described, and its security is analyzed.

Keywords: Embedded system; trusted enhancement; trusted computing; chain of trust.

1. Introduction

Embedded systems differ from computer systems in that they typically perform predefined tasks with specific requirements. Because the embedded system is only for a specific task, the designer can optimize it. Therefore, embedded systems are widely used in mobile intelligent platforms and sensor devices. However, too much performance is pursued and its security requirements are neglected, resulting in a delay in the security of mobile devices. In recent years, smart terminals have been widely used in mobile payment, medical electronics, smart home and so on. Security issues are slowly emerging.

Using trusted computing to solve embedded system security is a very feasible and efficient solution [1]. Enable the embedded system to start self-test and prevent illegal users from intruding.

2. The Main Idea

The basic idea of trusted computing [2]:

1). First establish a trust root in the computer system. The trustworthiness of the root is ensured by physical security, technical security and management security.

2). Then establish a chain of trust, starting from the root of trust to the hardware platform, to the operating system to the application, level 1 measurement and certification level, level 1 trust level, and extend this trust to the entire computer system. This ensures the integrity of the entire computer system. The chain of trust defined by TCG. see FIG.1

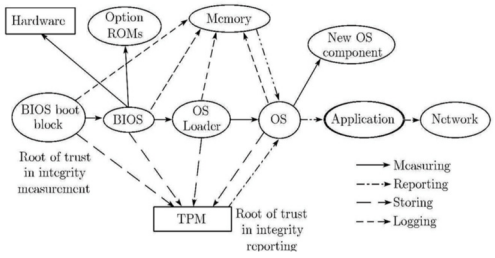

Fig. 1 Trusted PC chain of trust

Our work is to refer the trusted computing theory to the embedded platform, and use the SD card as an external trusted platform module to extend the trust boundary from physical hardware to software level and even the entire device to ensure the credibility of the embedded device. Start and run. Start with the underlying hardware and provide users with a secure software runtime environment.

3. Trust Chain Model for Embedded Platforms

3.1 Embedded Device Specificity

When constructing the trust chain model of an embedded device, it is necessary to consider its particularity.

1). Embedded devices are typically designed for a specific purpose, and they do not have a TPM device for a trusted PC, so they lack the root of trust necessary to boot.

2). Due to its pertinence, embedded devices have high requirements for application scenarios and body types. How to ensure the security of embedded devices without adding hardware and affecting its performance must be considered.

3). TPM device on a trusted PC cannot be completely copied to an embedded device.

3.2 The Startup Process of the Embedded System

1). After the embedded system is powered on, the boot loader is booted to detect device status and script status. The Boot loader is a program used to initialize a hardware device, and its implementation relies mainly on hardware.

2). After the Boot loader starts, copy the boot program to memory, further initialize the embedded device, and boot the Linux kernel.

3). Finally boot the Linux kernel to boot, load the Linux kernel into memory, and launch the set boot parameters to execute the kernel.

4). Jump to the mount file system, boot the init process, and complete the Linux system boot.

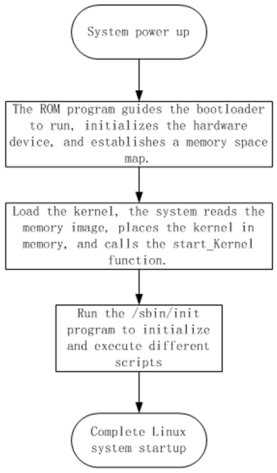

Fig. 2 Embedded system startup process

3.3 Linux-Based Embedded System Trust Chain Model

The trusted embedded system is to add hardware with security protection function on the embedded device, and build a trusted computing environment through the combination of software and hardware.

Through the analysis of the startup process of the embedded system and the idea of trusted computing, we only need to find a way to establish the root of trust, and then construct the chain of trust. From the root of trust, to the hardware platform to the operating system, the first level of measurement, the first level of trust. Extend the entire trust relationship to the entire embedded device.

1). We transform the Bootloader into a SBootloader (Security Bootloader) with metrics [3]. Make it a trusted root, because the startup of the embedded system starts from it. At the same time, the security SD card hardware is introduced and stored in the SD card.

2). The trusted root is unconditionally trusted, and SBootloader is the first starting point of the chain of trust. The SBootloader first measures the kernel and determines whether it is trusted according to the measurement result. If it is trusted, the system control is given to the kernel, otherwise the system is interrupted.

3). After SBootloader gives control of the system to the kernel Kernel, Kernel measures the process (init) and starts other key nodes. According to the measurement result, if the process init is trusted, the system starts the init process and passes the trust to the init process. Otherwise the boot process is interrupted. The init process is a user-level process started by the kernel. After the kernel starts itself, it starts the init to complete the boot process, so init is always the first process.

4). If the init process is secure and trustworthy, measure the Linux OS. If the Linux OS is secure and trusted, the Linux OS starts safely, otherwise the system is started.

The Linux OS measures the upper application.

5). In the trusted computer, the metric information is stored in the platform configuration register PCR inside the TPM chip, and the storage root key is stored in the non-volatile memory, which is protected by the TPM hardware, and the security is very high. Embedded devices do not have a TPM chip, in order to ensure that the metric information is secure and reliable. We use digital signatures to achieve integrity protection of metric evidence information. The system initializes, saves the metric reference value of each startup node in the SD card, and stores it by the storage root key SRK in the SD card. Each time the system extracts the reference value, the reference value is signed and verified to ensure the integrity of the reference information.

In summary, the trust chain of the embedded system is: SBootloader, Kernel, init, Linux OS, Application, and the trust chain model is shown in Figure 3.

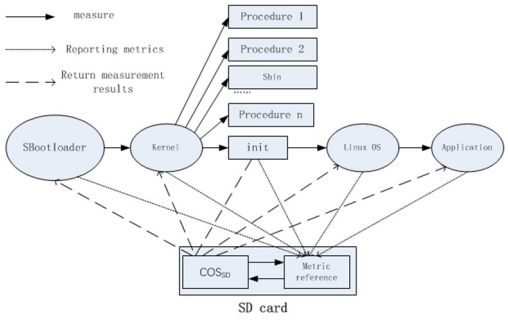

Fig. 3 Trusted PC chain of trust

3.4 Trust Transfer Process

Let the node of the current system control right be SQ, and perform security measurement on the next node TQ, and verify the TQ node is safe and trustworthy through the measurement result. Verification by handing control of the system to TQ, otherwise the system is interrupted.

The chain of trust transfer analysis process is as follows:

SQ: a node that has system control rights;

TQ: The node being measured; the next step is to acquire system control rights;

SRK: The platform stores the root key, which is stored in the SD and cannot be moved to protect its security.

1). SQ: PCRTQ=Hash {TQ}

The node SQ that has the system control right measures the node TQ, and obtains the measurement result PCRTQ.

2). SQ → SD: {SRKPrivacy {PCR}, PCRSD}

The SQ transmits the PCRTQ to the SD card together with the metric reference result SRKPrivacy {PCR} of the TQ signed by the stored root key SRK.

3). COSSD: Result=Verify {PCRTQ, SRKPrivacy {PCR}}

After receiving the message, the secure SD card uses its computing chip COS to determine whether the metric values PCRTQ and SRKPrivacy {PCR} signature values are correct, and finally obtains the measurement result Result.

4). SD→SQ: Result

The SD sends the measurement result to the node SQ, and the node SQ determines whether to hand over the system control right to the node TQ according to the measurement result.

It can be seen from the process of the chain of trust that its metric model is a star metric model. Star type. The star metric model compares the chain metric model, and the metric root RTM is stored in the SD card with high security. The metric is the primary metric level and the trust loss is relatively small. The trust chain star metric model designed in this paper is shown in Fig. 4.

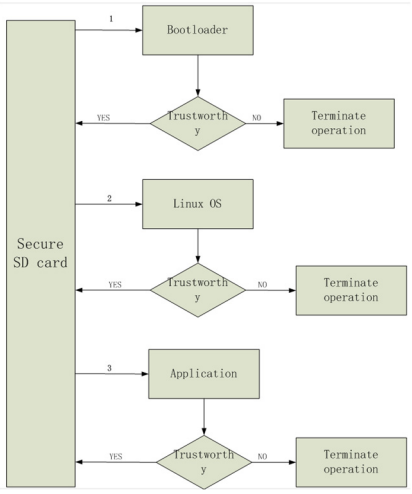

Fig. 4 Star letter metric model

4. Embedded System Trusted Enhancement Simulation Verification

4.1 Building an Embedded System Development Environment

Software, hardware, and other tools should be prepared in advance, and then install certain steps to build an embedded system development environment to meet predetermined requirements.

Before developing an embedded system, the software tools that need to be prepared are:

Linux204 kernel; Linux; VMware, FTP, GCC cross-compilation tool; RSA algorithm; Boot loader code, such as: Uboot or VIVI.

Hardware tools: Secure SD card, a PC with serial port is used as ARM target board, serial line, USB data line, etc.

4.2 Transformation of the Linux Kernel

The kernel adds additional functions on the original basis, and performs integrity measurement on system startup scripts such as init. The init configuration script implements self-starting of the Linux OS system, and measures the integrity of the init to ensure the normal startup of the Linux OS and prevent unauthorized users from acquiring. Item, launching the Linux OS. Thereby ensuring the security and credibility of the kernel.

4.3 Boot Loader Security Transformation

The main functions of the boot loader are: copy the operating system image file to RAM, and then jump to its entry to execute, and the source of the operating system file can be flash, PC, SD Card, PC (can be through the network, USB Even serial port transmissions and so on.

Prepare the boot loader source file. The Boot Loader needs to implement the kernel integrity measurement to ensure its security and credibility. This article uses Uboot to develop Linux and transform the boot loader.

Uboot is used to develop the Boot loader. It is an open source software that supports most ARM systems and includes common peripheral drivers. Has good open source [4].

Uboot is a small program that runs before the operating system kernel runs. Through this program, you can initialize the device and create a map of the memory space.

SBootloader adds a function to the Boot loader. It is to measure the integrity of the Kernel. It compares with the metric reference value in the SD card to ensure that it is trusted. If it is trusted, the system control is given to Kernel.

There are four main steps in Uboot to start the kernel process:

1). Move the kernel to the DDR

2). Verify kernel format, CRC, etc.

3). Prepare the transmission parameters

4). Jump to the execution kernel

Main functions involved: do_bootm and do_bootm_linux

Kernel startup through function: theKernel=(void(*)(int,int,uint))ep; Assign ep to theKernel, this function points to the real entry address of the OS image loaded in memory, which is the first execution code of the operating system, and sets the load address of the kernel.

When Uboot starts the kernel, the machine code is passed to the kernel, and the environment variable machid, and gd->->bi_arch_num are passed to the kernel.

Before the machine code is passed to the kernel, we need to measure its integrity.

The integrity is measured by the Detection_Kernel(ep) function. By passing the digest value of the metric to the SD card, the SD card chip COS determines whether the metric value and the digest value are correct. The result is fed back to the SBootloader, and the SBootloader chooses whether to give the system control to the kernel according to the received measurement result. The trusted boot process of the embedded system based on this trust mechanism is shown in Figure 5.

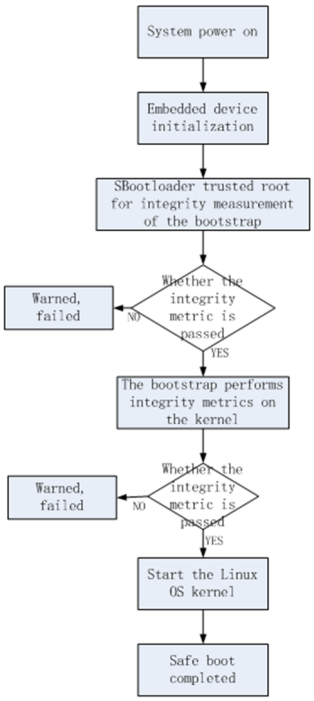

Fig. 5 Trusted PC chain of trust

4.4 Security Analysis

In order to verify whether the embedded device based on the trusted chain mechanism is secure and reliable, we need to analyze its security.

We need to ensure the confidentiality, integrity, reliability, and non-repudiation of embedded systems. We mainly analyzed the witch attack.

Witch attacks are divided into two situations. The first one is to fake the boot node and load the system through illegal means to achieve damage to the embedded system. The second is to tamper with node information and then invade the embedded system through the node.

In the first case, an attacker uses his or her illegal code to replace a legitimate code, thereby forging a node and invading the system for illegal purposes. If this happens, the enhanced embedded system proposed in this paper can achieve a good preventive effect. We perform integrity metrics for each node, so even if this happens, the trusted enhanced embedded system can discover and terminate the system in time.

In the second case, the attacker tampers with the configuration file of the embedded system or the process code to achieve its illegal purpose. At each step of the embedded system startup, we will perform integrity metrics, so if the attacker tampers with it, the secure SD card will feed back the measurement results to the previous node, which will not give control of the system to the next one. Node, the system will stop running.

Therefore, the safe boot mechanism proposed in this paper has a good defense effect on witch attacks.

5. Summary

This paper analyzes the chain of trust proposed by the TCG organization, which is suitable for the PC. We analyzed the specificity of embedded systems and introduced trusted computing technologies into embedded systems. The trust chain was transformed and the effectiveness of the trust mechanism was verified by experiments. Externally externalize the SD card and extend the trust boundary from physical hardware to software level to the entire embedded platform through trusted computing technology. Do not add too much hardware to prevent the specialty and pertinence of embedded devices. Provide security for the user's platform environment from the bottom of the hardware. Finally, the witch attack was analyzed.

References

[1]. Changing Shen, Huangpu Zhang, Haemin Wang, et al. Research and development of trusted computing [J]. China Science: Information Science, 2010, 40(2): 139-166.

[2]. Mantegna R N. Fast, accurate algorithm for numerical simulation of Lévy stable stochastic processes[J]. Physical Review E Statistical Physics Plasmas Fluids & Related Interdisciplinary Topics, 1994, 49(5):4677.

[3]. Bo Zhao, Yongkang Fei, et al. Research on Secure Boot Mechanism of Embedded System [J], Computer Engineering and Applications, 2014, 50(10):72-77.

[4]. Wei Xu. Design of Embedded Image Acquisition System Based on ARM9 [D], Nanjing University of Science and Technology, 2009, 16-17.