doi:10.3969/j.issn.1672-1292.2026.03.002

基于嵌入式系统的可信启动构架设计及实现研究

王修栋[1],施建平[1,2],霍银龙[3],张　华[1],岳彦利[4]

(1.南京师范大学南瑞电气与自动化学院,江苏 南京 210023)
(2.南京师范大学江苏省三维打印装备与制造重点实验室,江苏 南京 210023)
(3.南京国电南自电网自动化有限公司,江苏 南京 211153)
(4.南京工程学院能源与动力工程学院,江苏 南京 211167)

[**摘要**] 分析嵌入式系统的启动流程,提出一种基于树莓派嵌入式系统的可信启动方法. 该方法在硬件上采用 STM32F103RCT6 控制器,实现 SHA-256 加密算法,并提供非易失性存储空间等可信计算模块的基本功能,以此作为硬件可信根. 同时,设计了外围启动控制电路,用于连接树莓派 TF 卡和系统电源,以实现对树莓派启动过程的控制,从而使控制器能够主动读取位 TF 卡中的树莓派启动文件、内核镜像以及操作系统中的其他文件,进行主动度量并验证度量结果. 实验结果表明,系统成功度量了启动文件,完成了树莓派的可信启动. 提出的可信启动方法与国内自主提出的可信计算 3.0 双体系架构相契合,可为其他嵌入式设备的可信启动实施提供参考.

[**关键词**] 嵌入式控制,可信计算 3.0,可信启动,信任链,加密算法
[**中图分类号**]TN492　[**文献标志码**]A　[**文章编号**]1672-1292(2026)03-0014-08

Research on the Design and Realization of Trusted Boot Architecture Based on Embedded System

Wang Xiudong[1], Shi Jianping[1,2], Huo Yinlong[3], Zhang Hua[1], Yue Yanli[4]

(1.School of Electrical and Automation Engineering, Nanjing Normal University, Nanjing 210023, China)
(2.Jiangsu Province Key Laboratory of 3D Printing Equipment and Manufacturing, Nanjing Normal University, Nanjing 210023, China)
(3.Nanjing SAC Power Grid Automation Co., Ltd., Nanjing 211153, China)
(4.School of Energy and Power Engineering, Nanjing Institute of Technology, Nanjing 211167, China)

Abstract: This paper analyzes and researches the boot process of embedded system and proposes a trusted boot method based on Raspberry Pi embedded system. The method adopts STM32F103RCT6 controller in hardware to realize the basic functions of trusted computing module such as SHA-256 encryption algorithm, non-volatile storage space, and so on, and uses this as the hardware trusted root. Meanwhile, a peripheral boot control circuit is designed to connect the Raspberry Pi TF card and the system power supply in order to realize the Raspberry Pi boot process control, so that the controller can actively read the Raspberry Pi boot file, kernel image, and other files in the operating system located in the TF card to realize the active metrics, and to verify the results of the metrics. It is verified that the system successfully measures the boot file and completes the trusted boot of Raspberry Pi. The trusted boot method proposed in this paper can be compatible with the trusted computing 3.0 dual architecture proposed independently in China, and provides a reference for the implementation of trusted boot for other embedded devices.

Key words: embedded control, trusted computing 3.0, trusted boot, chain of trust, encryption algorithm

可信计算技术是保障计算机系统安全的可靠手段,成为当前信息安全领域的研究热点. 该技术旨在建立主动防御的信息安全保障体系[1],其核心思想是在计算机系统内部搭建具有物理安全属性的信任根基,该信任根基一般是具有密码算法的专用芯片,将信任根设为计算机系统信任的源头,从信任根出发对系统启动到运行的关键步骤进行逐层安全校验,将信任从信任根逐级扩展到整个系统,最终建立一种稳定

收稿日期:2024-10-11.
基金项目:国家自然科学基金项目(52005268).
通讯作者:张华,本科,实验师,研究方向:工业电气自动化、电气技术、自动化系统以及弱电系统集成. E-mail:61080@njnu.edu.cn

可靠、可管控的安全运行环境,保证计算机平台各类涉密与敏感操作的的安全性[2-3]. 可信计算组织(trusted computing group,TCG)的信任模型如图 1 所示.

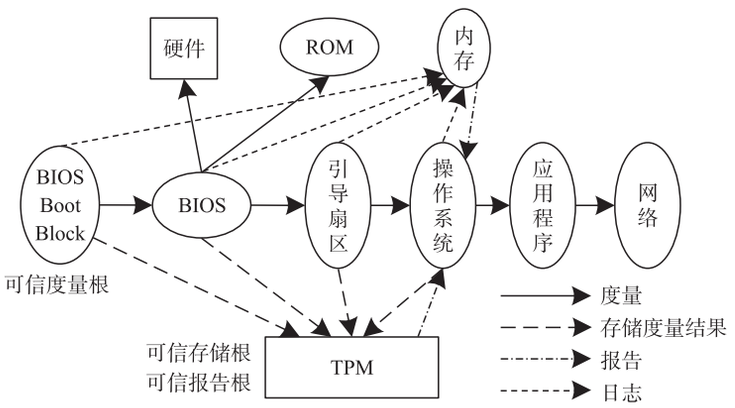


图 1　TCG 信任链

Fig. 1　TCG trust chain

自 20 世纪 90 年代首次提出可信计算至今,经过了 30 年的发展,国际可信计算组织相继建立了 TPM1.2 和 TPM2.0 国际标准规范. TPM1.2 仅支持单一的密码算法,TPM2.0 则增加了的密码算法支持,同时增加了授权层次,得到更广泛的应用. 当前各大芯片厂商都在商用的处理器中集成了可信计算相关功能. 产业内代表性的集成方案有 ARM 公司的 TrustZone 技术、Intel 公司的 SGX 技术和 AMD 公司的 SEV(secure encrypted virtualization)技术等均在处理器中实现了内存的隔离机制,能够为上层应用提供安全的运行环境,保障敏感程序的安全性,并被广泛应用在移动手机和云平台中.

我国可信计算的发展秉承着核心技术自主创新、信息安全自主掌控的理念,经过可信计算 1.0 至可信计算 3.0 三个阶段的发展,在可信平台模块(trusted platform module,TPM)的基础上研制了支持国密算法的可信密码模块(trusted cryptography module,TCM),以及具有主动度量功能和双体系架构的可信平台控制模块(trusted platform control module,TPCM)[4-7]. 目前,我国自主可信技术产业迅速发展,涉及移动端、服务器、工业控制等多个领域,并在国家关键信息基础设施中发挥重要作用.

当前可信计算技术已在服务器、个人计算机等设备中实现规模化落地应用. 现阶段嵌入式设备大量普及,不仅覆盖传统消费电子产品,还延伸至智能家电、安防监控等新兴硬件场景,加之嵌入式系统本身存在诸多安全短板,推动可信计算技术向嵌入式场景延伸拓展已成为行业发展的必然趋势[8-11]. 本文在可信计算 3.0 理念下,从嵌入式设备安全可信启动的角度出发,以树莓派嵌入式系统为例,采用 STM32F103RCT6 微控制器实现密码算法,将其作为系统硬件可信根,并设计外围启动控制电路,形成可信平台控制模块,实现嵌入式系统的主动度量和安全启动.

1　嵌入式系统启动流程分析

嵌入式设备种类繁多,应用场景广泛,因而相对于个人电脑,其具有更多的启动方式. 典型的启动方式有 NAND Flash 启动、MMC/SD(TF)启动、网络启动等[12-13]. 现以树莓派 4B 的 TF 卡启动方式为例,分析其启动流程.

树莓派 4B 启动过程如图 2 所示. 上电启动后,CPU 处于 standby 状态,由 GPU 负责启动系统. SOC 上电后首先检查是否更新 BOOTROM 中的引导加载程序,若无需更新,则由 GPU 开始执行引导加载程序. 这部分代码是芯片出厂时固化在 BOOTROM 中的,通常被称为 first-stage bootloader,代码会初始化时钟和 TF 卡等硬件,并将位于 TF 卡第一分区的 start4.elf 文件读取到 SRAM 中,开始执行 start4.elf 文件. start4.elf 为树莓派的启动文件,负责初始化 SDRAM,并从第一个分区中读取并解析 config.txt,该文件中包含树莓派的配置信息. 之后从 TF 卡的第一分区中读取 cmdline.txt,此文件中包含启动内核的相关参数. start4.elf 继续将 TF 卡第一分区中的 kernel.img、ramdisk.dtb(即将镜像文件和设备树文件,加载到内存的指定地址,然后向 CPU 发出重启信号,随即 CPU 结束 standby 状态,开始从内存的指定地址执行内核代码,正式进入系统启动过程. 综上所述,树莓派的启动由 BOOTROM 中的代码以及 start4.elf 文件控制. BOOTROM 中的代码

为出厂固化,无法更改,因此保证 start4.elf 的可信性非常重要. 后文将基于 TF 卡中的这些启动文件做度量运算,并根据度量结果决定是否给树莓派 4B 上电启动.

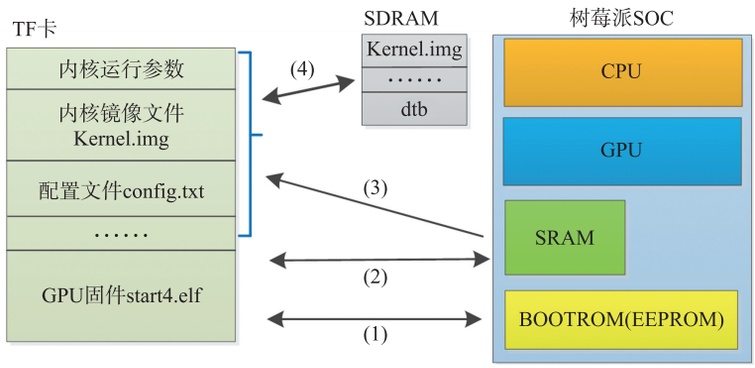


图 2　树莓派启动流程

Fig. 2　Raspberry Pi boot-up process

2　可信系统实现

2.1　可信系统总体设计

本文基于可信计算 3.0 理念[14-15],主要从建立可信实体,设计外围启动控制电路、实现主动度量的角度出发,创新性地提出了一种适用于从 TF 卡启动的嵌入式设备的可信启动方法. 可信系统结构如图 3 所示,可信系统实物如图 4 所示.

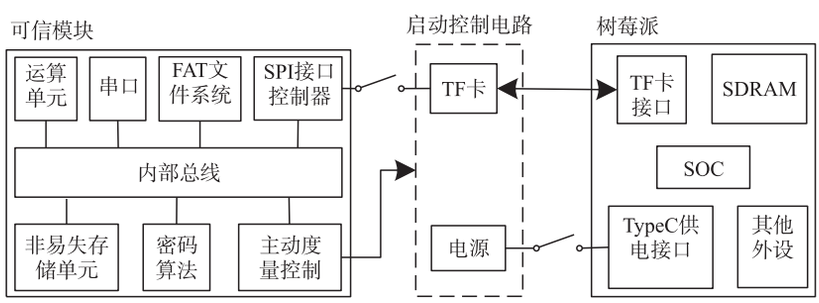


图 3　可信系统结构图

Fig. 3　Structure diagram of trusted system

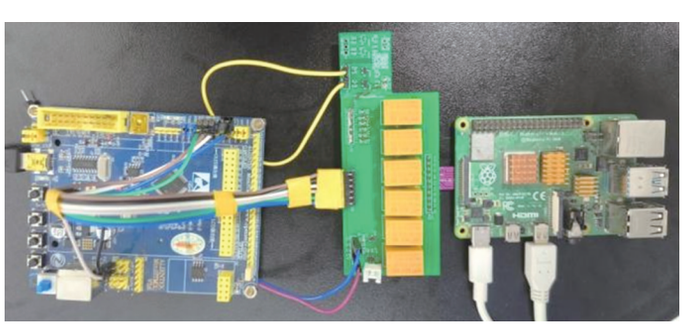


图 4　可信系统实物图

Fig. 4　Physical photo of the trusted system

图 4 中从左到右依次是 STM32F103RCT6 单片机、启动控制电路、树莓派 4B. 系统采用 STM32F103RCT6 微控制器实现 SHA-256 算法,作为系统的可信模块,完成可信度量、可信存储和可信报告等任务. 启动控制电路用于控制上电时序,使可信模块先于树莓派 4B 启动,并对启动文件进行度量. 如果度量结果与保存在可信模块非易失性存储器中的值相同,则输出控制信号,使树莓派 4B 上电启动. 反之,不予上电启动,并通过可信模块的串口输出错误信息. 可信启动具体流程如图 5 所示.

2.2 可信模块算法实现

哈希函数属于密码学体系里的关键技术,常被运用于数据完整性核验、口令身份鉴别、数字签名等多种安全场景. 借助哈希运算可以从原始数据中提取唯一标识数据特征的哈希摘要,该变换属于单向不可逆的数学映射关系,也正是这一特性赋予了哈希算法相应的安全性能[16-18]. 本文采用 STM32F103RCT6 实现 SHA-256 哈希算法,并使用其静态存储空间存储启动文件的标准哈希值,使用串口上报度量信息.

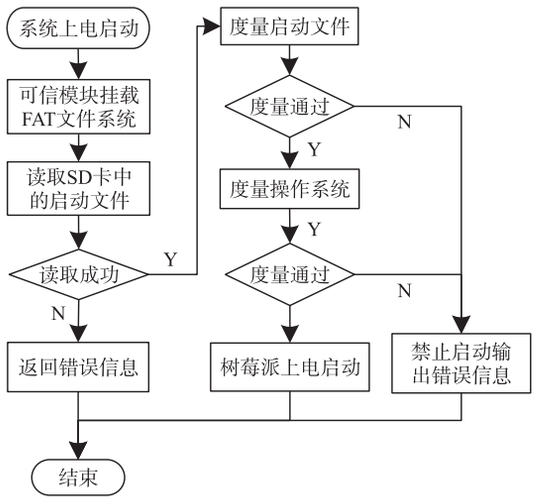


图 5 可信启动流程图

Fig. 5 Flow chart of trusted boot

SHA-256 算法输入报文不定长,输出报文长度固定为 256 比特,即 32 字节. 该算法的实现包含以下步骤[19-20]:

第一步,填充报文,使报文长度除以 512,余数为 448. 填充的第一比特为“1”,其余比特都填充“0”.

第二步,附加长度值,长度为 64 比特,该值为原始数据的长度.

SHA-256 算法涉及的操作全部为逻辑的位运算,包括以下的逻辑函数:

$$\mathrm{Ch}(x,y,z)=(x\ \&\ y)\oplus(\neg\ x\ \&\ z), \tag{1}$$

$$\mathrm{Ma}(x,y,z)=(x\ \&\ y)\oplus(x\ \&\ z)\oplus(y\ \&\ z), \tag{2}$$

$$\sum\nolimits_0(x)=S^2(x)\oplus S^{13}(x)\oplus S^{22}(x), \tag{3}$$

$$\sum\nolimits_1(x)=S^6(x)\oplus S^{11}(x)\oplus S^{25}(x), \tag{4}$$

$$\sigma_0(x)=S^7(x)\oplus S^{18}(x)\oplus R^3(x), \tag{5}$$

$$\sigma_1(x)=S^{17}(x)\oplus S^{19}(x)\oplus R^{10}(x). \tag{6}$$

其中,S^n 表示循环右移 n 个比特,R^n 表示右移 n 个比特.

第三步,将上述两步处理之后的输入报文分成 n 块,每块大小 512 比特,后续运算会进行 n 次迭代,以下步骤为迭代过程.

迭代第一步,对于每一块,将块分解为 16 个 32 比特的字,即 $W[0]\sim W[15]$. 再使用

$$W_t=\sigma_1(W_{t-2})+W_{t-7}+\sigma_0(W_{t-15})+W_{t-16},\quad (16<t<64). \tag{7}$$

依次构造 48 个字,一共组成 64 个字.

迭代第二步,对哈希初值进行 64 次加密循环. 完成 64 次加密循环表示完成一次迭代,每次加密循环如图 6 所示. W_t 为该块的第 t 个字,K_t 为给出的常量. A~H 为初始化哈希值,经过循环加密后得到新的哈希值. A~H 这 8 个字再经过 n 次迭代,即得到最终的结果. 在本设计中,使用 C 语言在 STM32F103RCT6

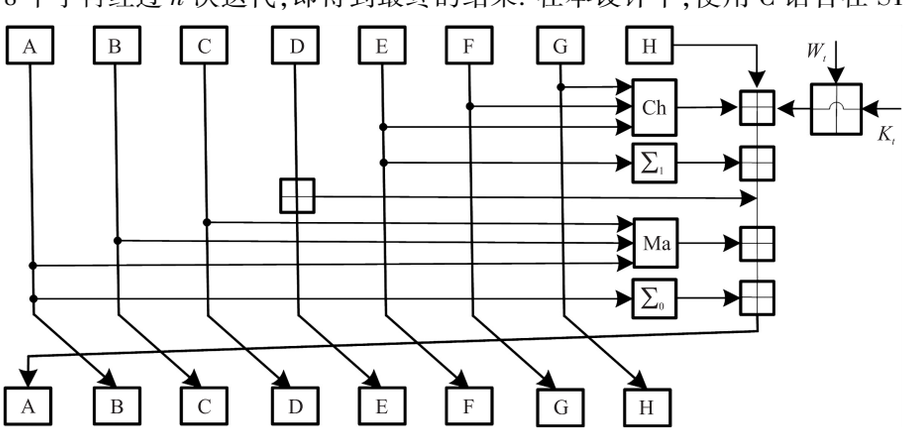


图 6 哈希循环加密步骤

Fig. 6 Steps of Hash-loop encryption

上实现该算法,经过测试,STM32F103RCT6 计算结果与 SHA-256 在线验证工具输出结果一致. 测试结果如表 1 所示.

表 1　可信模块中 SHA-256 算法测试

Table 1　Test of SHA-256 algorithm in trusted module

输入报文	可信模块计算输出报文	SHA-256 在线验证工具
This is a test	e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855	e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
0123456789	84d89877f0d4041efb6bf91a16f0248f2fd573e6af05c19f96bedb9f882f7882	84d89877f0d4041efb6bf91a16f0248f2fd573e6af05c19f96bedb9f882f7882

2.3　可信模块底层驱动实现

树莓派的启动文件位于 TF 卡中的第一个分区,该分区为 FAT32 文件系统,可信模块要度量启动文件,必须先移植 FAT 文件系统,才能读取 TF 卡中的启动文件,完成可信度量. FATFS 是一个文件系统模块,专为小型嵌入式系统设计,该模块结构分为 3 层,分别是应用层、中间层、底层驱动. 应用层位于最上层,提供接口函数,如 f_open、f_close 等,使用者直接调用即可. 中间层实现了 FAT 文件的读写协议. 底层驱动为 SPI 读写 TF 卡,包含 SPI 协议和 TF 卡协议,需由使用者实现. 下载 FATFS 软件包,添加相关文件即可完成移植,移植过程较为简单,重点介绍底层驱动的编写,即使用 STM32F103RCT6 的 SPI 接口读写 TF 卡.

TF 卡的协议较为复杂,针对不同卡类型以及不同的容量有不同的协议,树莓派的 TF 卡为高速 SD2.0 卡,遵循 SD2.0 协议规则. TF 卡支持 SDIO 和 SPI 两种协议,首先拉低 TF 卡片选信号,表示使用 SPI 协议,然后进行 TF 卡的初始化,发送相关命令进入卡识别模式,进而获取 TF 卡的容量等信息,完成卡识别之后,即可进入数据传输模式,包括单块读、多块读、单块写、多块写等操作. TF 卡的初始化操作流程如图 7 所示.

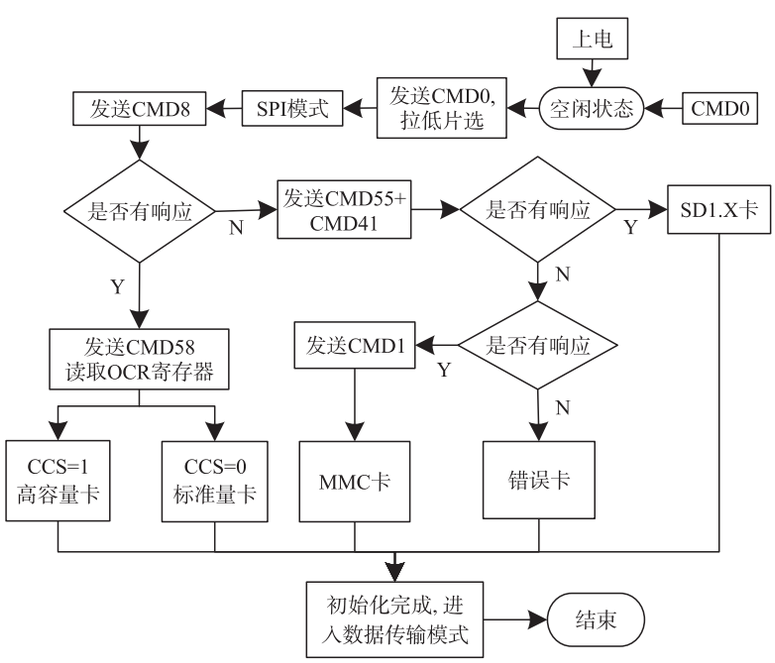


图 7　SD 卡初始化流程

Fig. 7　SD card initialization process

在完成底层驱动的编写与移植后,使用可信模块读取 TF 卡中的启动文件、内核镜像以及设备树文件,并用串口打印文件名和文件大小,文件大小单位为字节,结果如图 8 所示. 由于可信模块是由 STM32F103RCT6 微控制器实现,其内存资源有限,片内 flash 大小为 256 kB,片内 RAM 仅 48 kB,而启动文件较大,需分次读取和度量计算,每次读取大小为 1 024 字节,然后计算哈希值,再循环操作,直至文件结束.

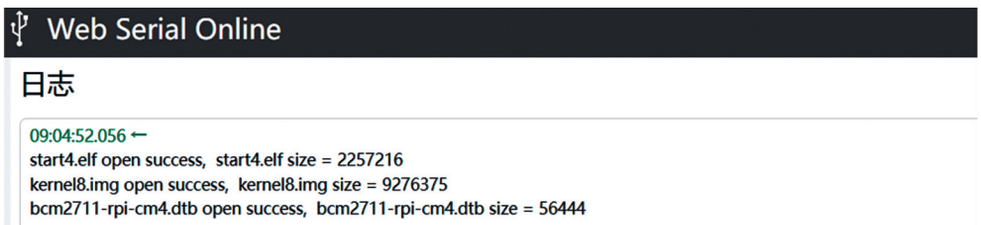
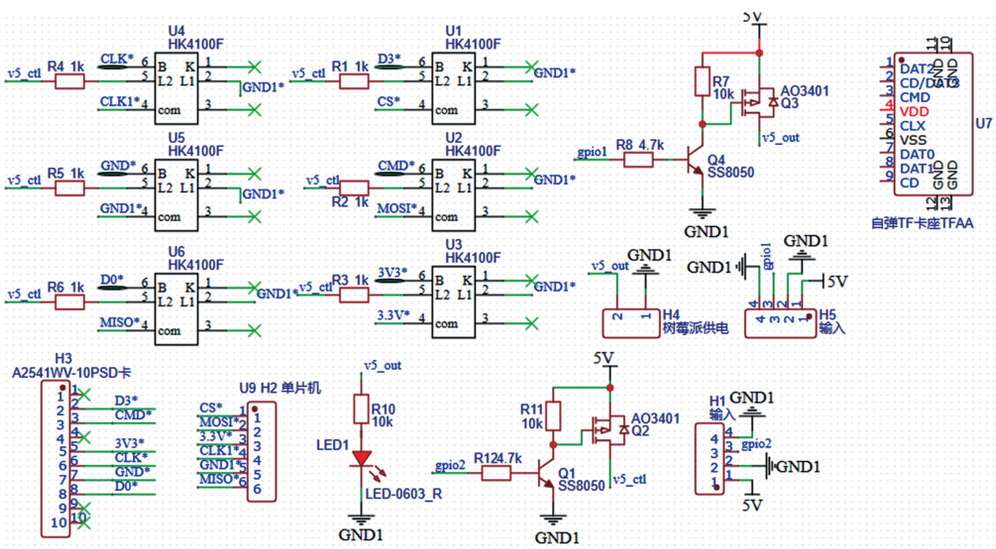


图 8　获取 TF 卡中重要文件和文件大小

Fig. 8　Obtaining important files and file sizes in TF card

2.4　启动控制电路设计

启动控制电路分为三部分,分别为 TF 卡接口、信号通断控制和上电时序控制,启动控制电路如图 9 所示. TF 卡接口部分使用 TF 卡延长器,将树莓派 4B 的 TF 卡接口引出,分为两路. 一路连接原树莓派接口,供树莓派正常使用. 另一路通过信号通断部分连接可信模块,供可信模块进行可信度量. 信号通断部分采用 6 个 HK4100F-DV5V 电磁继电器,用于 TF 卡信号的切换. TF 卡接口采用 SPI 协议,读写速度最高达数十兆赫兹,属于高频信号线. 如果 TF 卡两路信号线并联,则会影响数据的传输,使可信模块无法度量文件,树莓派无法启动,因此采用继电器进行信号隔离,由可信模块输出控制信号,使 TF 卡的两路信号不同时接通. 上电时序控制部分采用 PMOS 通用开关电路,电路前级为三极管,由三极管驱动 PMOS,由可信模块的 GPIO 驱动三极管. 当可信模块度量成功,控制引脚输出高电平使三极管导通,PMOS 导通,树莓派接通 5 V 电源启动. 当可信模块度量失败,控制引脚输出低电平,三极管与 PMOS 关断,树莓派无法接通电源启动.



(a) 启动控制电路原理



(b) 启动控制电路实物

图 9　启动控制电路

Fig. 9　Start-up control circuit

3　可信启动测试

3.1　启动文件度量测试

可信模块上电启动后,从 TF 卡中读取启动相关文件,测试对象选择 TF 卡中的部分文件,即启动文件、镜像文件、设备树文件,将对这些文件进行度量运算,并将度量结果与保存在非易失性存储器中的标准值对比,测试结果通过串口发送至电脑在线串口调试工具 Web Serial Online,并在串口工具上显示计算出的 32 字节哈希值. 最后使用在线验证工具计算的度量值作直观对比,可得可信模块度量启动文件结果与在线验证工具计算结果一致. 可信模块计算结果与在线验证工具计算结果如图 10 所示. 最后进行加电测试,观察系统运行过程. 可信模块先上电,开始度量启动文件. 度量通过后,可信模块控制启动控制电路使树莓派上电,树莓派启动成功. 经过测量系统从上电到树莓派启动一共花费时间约 55 s. 树莓派在无可信度量的情况下,自上电到启动花费 35 s. 系统可信启动多消耗 57.1%的时间,在可接受范围内.



(a) 可信模块度量结果



(b) 在线验证工具度量 start4.elf结果

图 10　文件度量结果

Fig. 10　File measurement results

3.2　非法更改启动文件测试

使用电脑更改 start4.elf 文件,在文件末尾添加任意字符,模拟启动文件被恶意更改,重复上文操作. 观察系统启动过程,以及串口输出内容,发现输出错误信息,树莓派启动失败. 串口输出结果如图 11 所示.

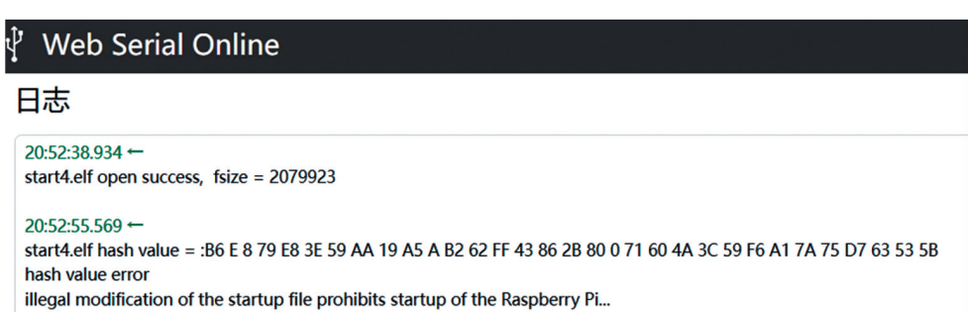


图 11　更改启动文件后测试结果

Fig. 11　Test results after modifying boot files

4　结论

在可信计算技术逐渐向嵌入式领域发展的趋势下,本文结合国内可信计算 3.0 理念,设计了一种用于嵌入式系统的可信启动方法. 该方法以 STM32F103RCT6 作为系统的可信模块,在其上部署 SHA-256 度量算法,度量启动文件,保存标准哈希值,并上报度量结果,同时设计外部的启动控制电路,实现了对树莓

派启动文件的主动度量,完成系统可信启动. 经过测试,度量结果正确,嵌入式系统可安全启动. 可信模块和树莓派分时复用 TF 卡的电路设计十分新颖,解决了部分嵌入式设备难以应用可信启动的问题. 此外,该方法还具有结构简单、成本低廉的优点,且无需更改启动代码层,便于实施. 此外,本方法不仅适用于从 TF 卡启动的嵌入式设备,也可为其他嵌入式设备的可信启动实施提供有效参考.

[参考文献](References)

[1] 连恺. 基于可信链传递的动态可信性系统的设计与实现[D]. 南京:南京航空航天大学,2018.

[2] 刘莉,孟杰,徐宁. 可信计算技术在操作系统发展中的应用综述[J]. 信息安全研究,2018,4(1):45-52.

[3] 郭晨雪,王超,姬胜凯. 可信计算信任链技术研究[J]. 网络安全技术与应用,2023(6):19-20.

[4] 杜君,王杨,郑锟,等. 可信计算 3.0 创新发展与应用实践[C]//第 38 次全国计算机安全学术交流会. 湖南,长沙. 2023.

[5] 冯登国,刘敬彬,秦宇,等. 创新发展中的可信计算理论与技术[J]. 中国科学:信息科学,2020,50(8):1127-1147.

[6] Zheng C M,Li J,Yao X X. Design and implementation of trusted boot based on a new trusted computing dual-architecture[J]. Computers & Security,2023,127.

[7] 张建标,黄浩翔,胡俊. 主动免疫可信计算综述[J]. 中兴通讯技术,2022,28(6):12-16.

[8] 王博,李波,高振铁,等. 基于 TPM 的嵌入式可信计算平台设计[J]. 单片机与嵌入式系统应用,2011,11(1):13-16.

[9] Promila J T,Jain S. TPM based secure boot in embedded systems[C]//2023 Third International Conference on Secure Cyber Computing and Communication(ICSCCC). Jalandhar,India,2023,786-790.

[10] 赵波,倪明涛,石源,等. 嵌入式系统安全综述[J]. 武汉大学学报(理学版),2018,64(2):95-108.

[11] 何欣枫,田俊峰,刘凡鸣. 可信云平台技术综述[J]. 通信学报,2019,40(2):154-163.

[12] 桑赫,林岩,张烽. 嵌入式系统硬件平台的启动方式研究[J]. 单片机与嵌入式系统应用,2015,15(12):5-7.

[13] Lin K J,Wang C Y.Using TPM to improve boot security at BIOS layer[C]//2012 IEEE International Conference on Consumer Electronics(ICCE). Las Vegas,NV,USA,2012:376-377.

[14] 沈昌祥,张大伟,刘吉强,等. 可信 3.0 战略:可信计算的革命性演变[J]. 中国工程科学,2016,18(6):53-57.

[15] 胡俊,沈昌祥,公备. 可信计算 3.0 工程初步[J]. 网络与信息安全学报,2017,3(9):79.

[16] 范明钰. 密码学理论与技术[M]. 北京:清华大学出版社,2008.

[17] 王张宜,李波,张焕国. Hash 函数的安全性研究[J]. 计算机工程与应用,2005,41(12):18-19.

[18] 张大松,姜洪朝,刘智国. 基于 SHA-256 的可信启动完整性校验程序设计与验证[J]. 网络安全技术与应用,2022(3):49-51.

[19] 杨晓辉,戴紫彬. 基于 FPGA 的 SHA-256 算法实现[J]. 微计算机信息,2006(11):146-148.

[20] Aydin Y,Garipcan M A,Özkaynak F. A novel secure s-box design methodology based on FPGA and SHA-256 Hash algorithm for block cipher algorithms[J]. Arabian Journal for Science and Engineering,2024,50:1-14.

[责任编辑:陈　庆]