

基于嵌入式可信平台的运行时监控方法

孙召昌, 马建峰, 孙聪, 卢笛

(西安电子科技大学网络与信息安全学院, 陕西 西安 710071)

摘要: 当前可信计算平台缺乏对自身运行时安全属性的监控, 对此, 提出一种针对嵌入式可信平台的运行时监控方法。通过自动化的代码插入和运行时实时监控, 保证可信平台的运行时安全功能符合设计规范, 并保证系统性能和运行状态符合特定条件约束, 同时对相应的异常进行实时处理。实验结果表明, 随着监控节点数的增长, 监控的准确性和实时性提高, 而监控开销和异常处理开销处于合理范围。

关键词: 可信计算; 运行时监控; 嵌入式系统; 线性时序逻辑

中图分类号: TP393

文献标识码: A

doi: 10.11959/j.issn.2096-109x.2017.00209

Approach on runtime monitoring based on the embedded trusted platforms

SUN Zhao-chang, MA Jian-feng, SUN Cong, LU Di

(School of Cyber Engineering, Xidian University, Xi'an 710071, China)

Abstract: The state-of-art trusted computing platforms fall short of monitoring its own runtime security properties. In order to mitigate these problems, an approach on runtime monitoring for the embedded trusted platforms was proposed. With the automated code instrumentation and the runtime monitoring, that the runtime security functionalities were consistent with the design specifications of trusted computing platforms could be made sure. The system performance and states meet specific constraints could also be ensured. The runtime exceptions on the specific properties were handled in real-time. The experimental results show that with the increase on the number of monitoring nodes, the precision and real-time performance of proposed runtime monitoring increase. Meanwhile, the costs on the monitoring and the exception handling are moderate.

Key words: trusted computing, runtime verification, embedded system, linear temporal logic

1 引言

可信计算技术通过建立硬件可信根和信任链

保证系统的安全性和完整性, 是信息系统安全领域长期以来的研究热点。依据可信计算组织(TCG, Trusted Computing Group)的可信平台架构定义^[1],

收稿日期: 2017-07-01; 修回日期: 2017-09-16。通信作者: 孙召昌, zhchsun1992@gmail.com

基金项目: 国家自然科学基金资助项目(No.61303033, No.U1405255); 国家高技术研究发展计划(“863”计划)基金资助项目(No.2015AA017203); 陕西省自然科学基金基础研究计划基金资助项目(No.2016JM6034); 航空科学基金资助项目(No.20141931001); 工信部某专项科研基金资助项目(No.MJ-2014-S-37)

Foundation Items: The National Natural Science Foundation of China (No.61303033, No.U1405255), The National High Technology Research and Development Program of China (863 Program) (No.2015AA017203), The Natural Science Basis Research Plan of Shaanxi Province (No.2016JM6034), The Aviation Science Foundation of China (No.20141931001), The Special Research Foundation of MIIT (No.MJ-2014-S-37)

TCG 软件栈 (TSS, TCG software stack) 是可信平台的软件核心, 用于保证应用程序对可信平台模块 (TPM, trusted platform module) 硬件功能 (完整性度量、安全存储、对称/非对称加解密等) 的有效使用。在嵌入式系统领域, 可信平台可用于构建安全实时操作系统, 为系统内核提供可信功能支持^[2]。可信系统的构建^[3,4]通常假定可信平台的初始完整性在可信平台运行过程中能够保持, 而对可信平台自身的分析验证^[5-7]则侧重于静态验证, 以上 2 方面研究均无法对部署后 TSS 在实际运行过程中产生的非法输入、服务状态异常等安全问题进行监控, 也无法对嵌入式可信平台的性能和线程状态等关键运行时属性提供监控。

针对以上问题, 本文提出一种嵌入式可信平台的运行时监控方法, 根据 TSS 设计规范, 从功能、性能、系统状态和输入等方面定义监控属性, 根据属性定义, 自动生成并插入监控代码和异常处理代码。得到的 TSS 监控系统保持原有 TSS 功能, 并在运行时验证可信平台是否违反监控属性, 同时能根据不同安全等级对违反监控属性的异常状态进行实时处理。实验结果表明, 随着监控节点数的增加, 监控的准确性和实时性提高, 而监控开销和异常处理开销处于合理范围。

2 嵌入式可信平台运行时监控方法

嵌入式可信平台作为被监控对象, 包括可信平台模块和可信软件栈这 2 个部分。TSS 为上层应用程序提供服务接口, 保证上层应用程序对 TPM 的同步访问和有效应用。TSS 自下而上分为 3 层^[1]。1) TCG 设备驱动库 (TDDL, TCG device driver library): 提供系统无关的 TPM 调用接口, 保证对 TPM 的串行访问; 2) TSS 核心服务 (TCS, TSS core service): 提供密钥管理、TPM 命令同步等核心功能; 3) TCG 服务提供者 (TSP, TCG service provider): 提供面向对象的应用程序服务接口。

由于 TPM 的主要功能由 TSS 调用并反映于 TSS 状态, 本文在 TSS 上构建嵌入式可信平台的运行时监控系统。图 1 为 TSS 监控系统的构建和处理流程。首先, 根据 TSS 的设计规范和系统要求确定监控属性, 并用线性时序逻辑 (LTL, linear temporal logic) 对监控属性进行形式化描述, 然后根据属性的形式化描述自动生成监控代码并插入 TSS 原系统, 得到 TSS 监控系统, 将该系统部署在嵌入式操作系统 (如 VxWorks) 平台上运行。

TSS 监控系统运行时, 被监控的 TSS 程序线程中与监控属性相关的系统状态、功能和性能变量状态值被发送至一个事件队列, 一个独立的监

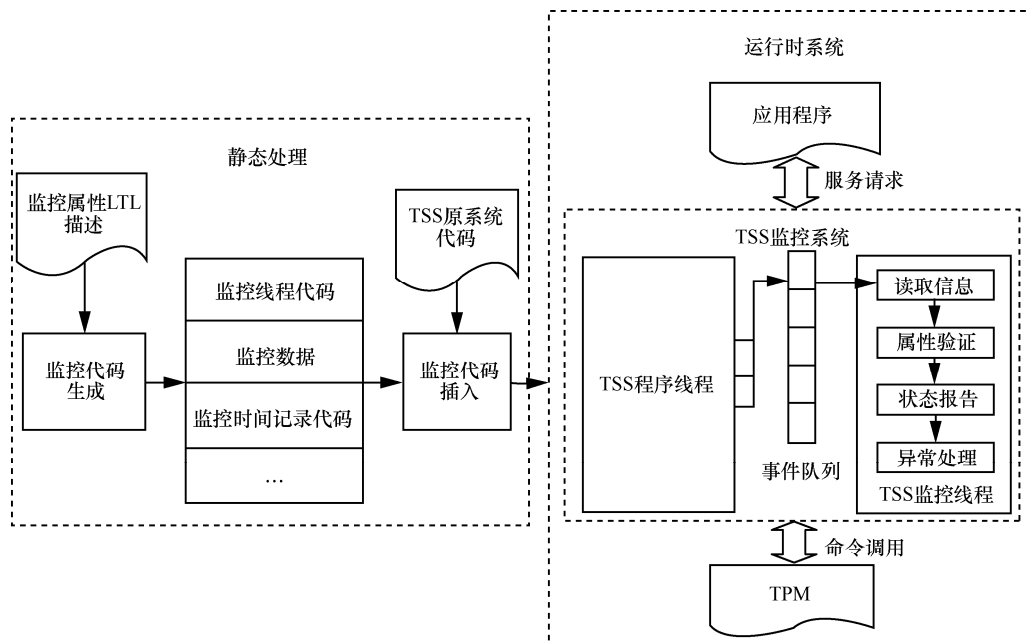


图 1 TSS 监控系统的构建和处理流程

控线程通过动态周期轮询的方式对事件队列中的变量信息进行采样,并依据采样结果验证系统状态是否满足属性要求。当发现违反监控属性的异常状态时,根据属性类型采取相应的异常处理机制。监控过程不影响原有 TSS 功能的正常调用。

2.1 属性规约

根据可信软件栈 TSS 的设计规范^[1]和系统安全需求,将监控的目标属性分为功能、性能、系统状态、合法输入 4 类,具体描述如下。

1) 功能属性:主要包括密钥合法性、度量权限和认证顺序 3 种属性。其中,密钥合法性指所生成(使用)密钥的长度与系统规定的长度一致,且密钥的使用时长在更新周期范围内;度量权限指度量信息的读取操作需要普通用户权限,写入操作必须具有高级用户权限;认证顺序指远程认证只能在自身认证完成后才能调用,由所使用的 TSS 中认证相关接口的具体设计决定。

2) 性能属性:主要包括队列长度、优先级设置和响应时间 3 种属性。其中,队列长度属性表示 TCS 层命令队列的实时长度不能大于系统规定的长度上限;优先级设置属性指从命令队列取出的命令始终是该队列中优先级最高的命令;响应时间属性指 TDDL 向 TPM 发起请求与从 TPM 接收数据之间的时间间隔在系统规定的时间间隔范围内。

3) 系统状态属性:主要指对线程状态的收集和分析,系统运行过程中,线程状态必须与系统规定的合法状态保持一致,合法的系统状态包括就绪态、阻塞态、延迟态和挂起态。监控的目标

线程主要包括 TCS 层命令接收线程、命令分发线程、远程服务线程等。

4) 合法输入属性:主要指上层应用程序提供的接口输入参数个数、类型和数据长度必须满足所调用接口的特定要求。

使用线性时序逻辑 LTL^[8]对以上各类目标属性进行描述,具体描述如表 1 所示,其中的属性公式表示一类形式近似的属性公式的共同模式。针对 TSS 的分层设计,为不同的目标属性选择适合的监控层次,从而保证运行时监控的正确性和及时性。如对度量权限属性选择在 TSP 层监控,可以对属性状态做出及时有效的判断,减少系统开销。

2.2 属性异常处理机制

不同属性的异常处理机制与所处的监控层次有关,表 2 为不同监控层次的属性设定对应的安全等级和异常处理机制,安全等级越高,对应的异常处理机制越复杂。其中,TCS 层属性与可信平台核心服务的正常运行紧密相关,安全等级最高,属性异常时挂起系统服务;TDDL 层属性与系统功能调用的正确性有关,属性异常时系统自动完成对当前接口的重新调用;TSP 层属性与用户调用可信服务的合法性有关,安全等级最低,属性异常时终止当前接口调用。

为提高监控系统的可用性,由 TCS 层的属性处理线程对各属性监控结果进行统一管理。算法 1 描述了对监控属性的异常处理,监控线程(MonitorThread)获取系统运行过程中的目标属性状态(第 3 行),如果发现属性异常,及时将该异常属性的类别信息发送至属性消息队

表 1 监控属性描述

分类	描述	属性公式的模式	属性层次
功能	密钥合法性	$\forall \text{key}: \square((\text{result_length} == \text{specify_length}) \wedge (\text{use_duration} < \text{specify_duration}))$	TDDL
	度量权限	$\square((\text{read_measure} \rightarrow \text{level}_{\text{NORMAL}}) \wedge (\text{write_measure} \rightarrow \text{level}_{\text{HIGH}}))$	TSP
	认证顺序	$\square(\text{remote_verify} \cup \text{self_verify}_{\text{FINISHED}})$	TSP
性能	队列长度	$\square((\text{queue_length} \leq \text{max_length}) \wedge (\text{queue_length} \geq 0))$	TCS
	优先级设置	$\square(\bigwedge_{i \in [1, \text{queue_length}]} (\text{priority_cmd_get} \geq \text{priority_cmd}_i))$	TCS
	响应时间	$\square(\text{call_time} \leq \text{specify_max_time})$	TDDL
系统状态	线程状态	$\square(\bigvee_{i \in [1, \text{state_num}]} (\text{task_state} == \text{specify_state}_i))$	TCS
合法输入	输入参数	$\square((\text{param_amount} == \text{specify_amount}) \wedge (\bigwedge_{i \in [1, \text{specify_amount}]} (\text{param_type}_i == \text{specify_type}_i)) \wedge (\bigwedge_{i \in [1, \text{specify_amount}]} (\text{param_length}_i == \text{specify_length}_i)))$	TSP

列（第 5 行~第 6 行），否则，直接完成属性状态的报告。属性处理线程（PropertyHandlerThread）接收属性消息队列的属性异常信息（第 13 行）并获取该属性的安全等级（第 14 行），根据安全等级采取适合的异常处理机制（第 15 行~第 29 行）。

表 2 属性安全等级分类

属性层次	安全等级	异常处理机制
TCS	0	挂起系统服务
TDDL	1	重新调用当前接口
TSP	2	终止当前接口调用

算法 1 监控属性的异常处理

```

1) Procedure MonitorThread()
2)  ....
3)  status ← getPropertyStatus();
4)  If status == violated Then
5)    property ← getCurrentProperty();
6)    sendProInfoToQ(property);
7)  Else
8)    完成属性状态报告;
9)  End If
10) End Procedure
11) Procedure PropertyHandlerThread()
12)  While true Do
13)    property ← recvProInfoFromQ();
14)    level ← getSecurityLevel(property);
15)    If level == 0 Then
16)      For each taskId created in TCS Do
17)        taskSuspend(taskId);
18)      End For
19)    Else If level == 1 Then
20)      终止当前接口调用;
21)      newResult ← funcInterfaceCall();
22)      If newResult == false Then
23)        输出错误信息;
24)      Else If newResult == true Then

```

```

25)        result ← newResult;
26)      End If
27)    Else If level == 2 Then
28)      终止当前接口调用;
29)    End If
30)  End While
31) End Procedure

```

2.3 监控代码的生成和插入

本文通过对 RiTHM 工具^[9]进行扩展完成监控代码的生成工作。该工具的输入为原始系统代码和由 LTL 描述的监控属性（表 1 中监控属性均对应于独立的配置文件），输出为用于实时监控的系统监控程序代码。该程序在运行时生成独立的监控线程，不影响原有程序线程的正常执行。

针对嵌入式系统环境和可信平台的具体异常处理要求，本文实现的扩展工具能够自动化完成以下代码转换操作。

1) 为使监控系统能够运行于嵌入式系统环境，自动化以下过程：① 用嵌入式系统（VxWorks）库函数替换通用系统（Linux）库函数；② 去除监控系统对于嵌入式环境无法支持的 OpenCL 函数库的依赖；③ 执行必要的语法规则转换，以使 C++ 代码符合嵌入式操作系统要求。

2) 自动生成第 2.2 节算法 1 所述的属性异常处理机制的实现代码，并将异常处理代码自动插入原代码。

3) 增加对多配置文件输入和多属性监控功能的支持，完成不同属性监控代码的复用。

工具扩展前后的功能对比如表 3 所示。

3 实现和评价

本节首先分析监控系统的生成开销，然后对监控系统的监控准确性、实时性和监控过程开销进行验证和评价。

3.1 RiTHM 工具时间开销分析

图 2 表示 RiTHM 工具扩展前后自动生成不

表 3 RiTHM 工具扩展前后的功能对比

工具	监控代码生成	嵌入式系统环境	异常处理代码生成	多配置文件的输入	多属性监控
RiTHM ^[9]	支持	不支持	不支持	不支持	不支持
RiTHM extended	支持	支持	支持	支持	支持

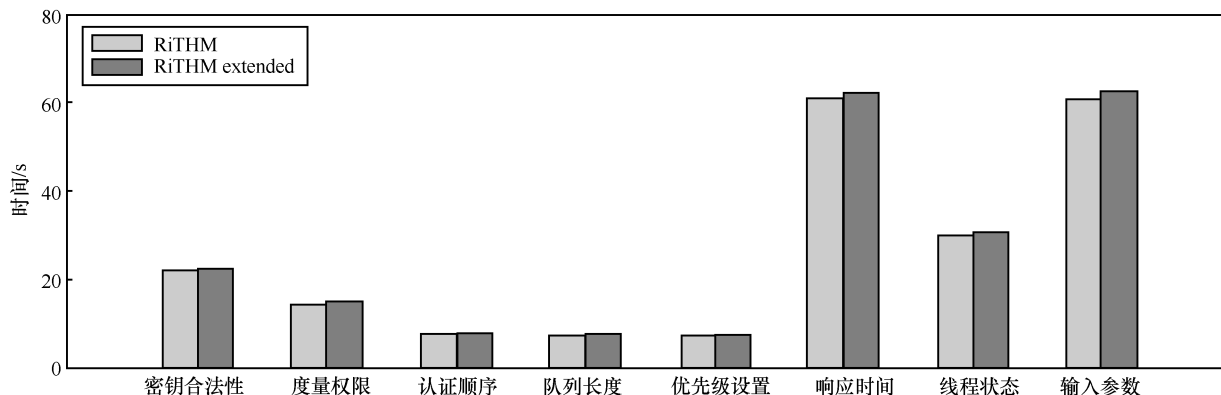


图2 RiTHM 工具扩展前后的时间开销对比

同目标属性对应监控代码的时间开销对比,图2表明,对工具扩展后,其时间开销比原工具高2.4%。但由于监控代码的生成过程是一个离线处理过程,可在通用计算环境中预先完成,因而时间开销的增加不会影响嵌入式环境下监控系统的运行开销。

3.2 实验环境与错误注入

本节实验用于评价嵌入式可信平台运行时监控方法的准确性、实时性和监控开销,在笔者团队已实现的嵌入式可信软件栈上进行。实验环境由POWERPC开发板和iTOP-4412开发板组成,如图3所示。其中,POWERPC开发板(MPC8247-400 MHz CPU/128MB SDRAM)运行VxWorks 5.5嵌入式实时系统,用于部署和运行TSS监控系统,并向VxWorks上层应用程序提供加解密、度量、认证等可信服务;iTOP-4412开发板(Exynos4412 Cortex-A9 1.4~1.6 GHz CPU/2GB DDR3 RAM)运行Linux操作系统,其上部部署TPM Emulator,用于模拟TPM的度量、加解密等功能。开发板之间通过串口通信完成命令和数据的传输。

结合表1和表2对监控属性的描述和分类,选取不同监控层次和安全等级上的3种属性(队列长度、密钥合法性和度量权限)进行实验。针对各属性的不同要求,分别自动化随机生成2000条数据(含非法数据)作为软件错误注入^[10]的测试用例,其中,与队列长度相关的非法数据指超出系统规定范围的队列长度值,与密钥合法性相关的非法数据指不满足获取密钥接口指定的密钥长度值,与度量权限相关的非法数据指不满足度量相关接口要求的用户权限等级值。在被注入数据

中,统计不满足属性要求的非法数据的比率分别为73.7%、87.4%、81.3%。

使用上述测试用例进行软件错误注入测试,由上层应用程序对TSP层的可信服务接口进行多次调用,每次调用过程中,在目标属性所处的监控层次创建错误注入线程,读取一组测试用例并修改目标属性相关的程序变量,完成TSS监控系统非法数据的随机注入。同时,在该监控层次的不同代码位置插入监控程序,验证监控属性相关的程序变量是否符合属性要求,每个位置插入一条监控程序,不同位置插入的监控程序数量称为监控节点数(number of monitor node)。

3.3 实验结果

实验从准确性、实时性、监控开销3方面进行评价。其中,准确性指实际监控到的错误数占注入错误总数的比例。实时性指错误发生时刻与该错误被监控到的时刻之间的时间间隔,时间间隔越短,实时性越强。监控开销分为时间开销和内存开销,时间开销指从接口调用到返回结果所用的时间,内存开销指软件系统运行过程中的平均内存分配。

对实验结果数据进行收集和分析,得到监控3种属性时,监控系统的准确性和实时性、时间开销及内存开销对比分别如图4~图6所示。

图4(a)说明,对于3种不同属性(度量权限、队列长度、密钥合法性)的监控准确性均随着监控节点数的增加而以近似线性趋势上升,当监控节点数足够多时,准确率可达100%。图4(b)说明,对3种属性的监控实时性具有相似的变化趋势,

随着监控节点数的增加, 错误发生时刻与该错误被监控到的时刻之间的时间间隔不断减小, 监控系统的实时性变强, 同时, 时间间隔的变化趋势随监控节点数的增加而逐渐减缓。

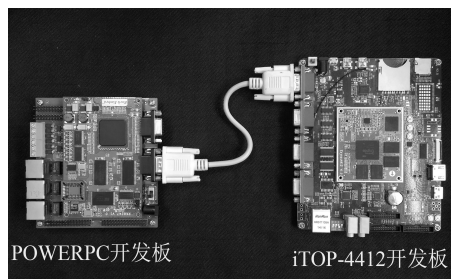


图 3 实验硬件环境

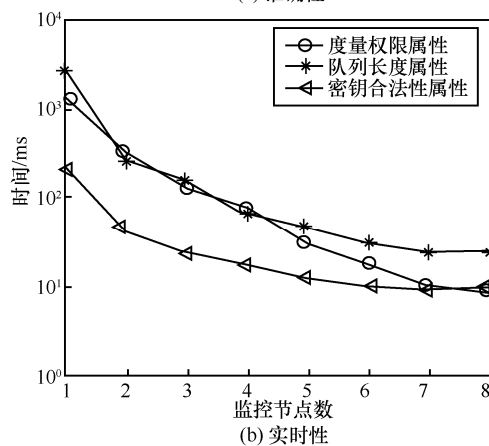
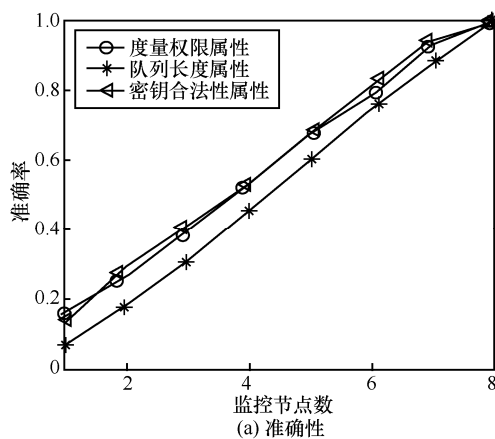


图 4 监控系统的准确性和实时性

在图 5 和图 6 中, TSS system 为 TSS 原系统对应的开销; TSS monitor system 指无异常处理的监控系统对应的开销; TSS exception handling system 指加入异常处理的监控系统对应的开销。图 5 和图 6 说明, 3 种属性对应的监控系统, 无论是否进行异常处理, 其监控时间开销和内存开销均随监控节点数的增加而增加, 且始终高于 TSS 原系统。

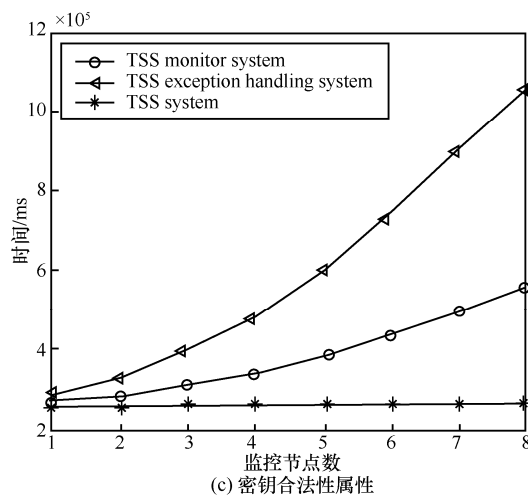
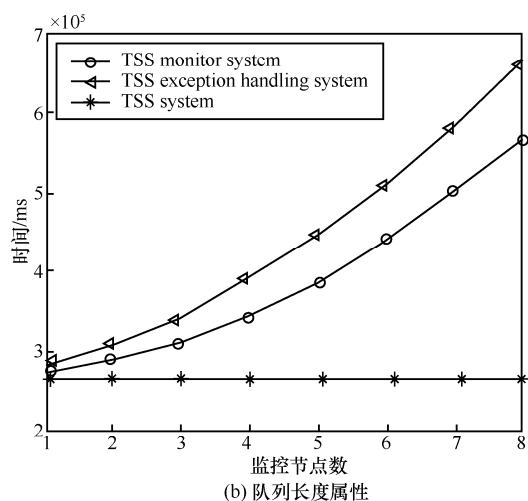
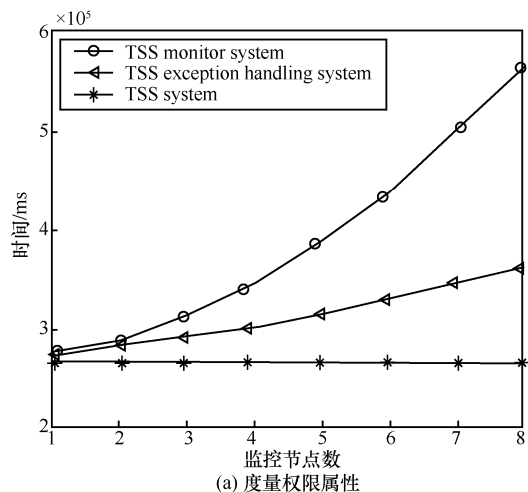


图 5 监控系统的时间开销

由于各属性的异常处理方式不同, 异常处理导致的开销增长也各不相同。对于度量权限属性, 发生异常时, 根据表 2 规则终止服务调用并返回错误信息, 服务调用次数减少, 因此添加异常处理功能后, 监控系统的时间开销和内存开销总体

上明显降低,如图 5(a)和图 6(a)所示。对于队列长度属性,发生异常时,依据表 2 规则挂起系统服务,由于异常属性信息的传输以及对各服务的挂起操作,导致添加异常处理功能后,监控系统

的时间开销和内存开销均小幅增加,如图 5(b)和图 6(b)所示。对于密钥合法性属性,发生异常时,系统根据表 2 规则自动完成对当前接口的重新调用,导致接口调用次数增多,因此添加异常处理功能后,监控系统的时间开销和内存开销均明显增加,如图 5(c)和图 6(c)所示。

虽然实验表明,监控开销随监控节点数增加持续增长,但对于实际系统,监控节点数的选取应以满足系统对监控准确性和实时性的具体要求为依据,满足要求后,监控节点数不用继续增加,此时的监控开销为实际监控开销。针对当前嵌入式软件栈,实际监控开销能够保证监控系统在嵌入式环境下可用。

4 结束语

本文提出的针对嵌入式可信平台的运行时监控方法,通过扩展运行时验证工具 RiTHM 实现对嵌入式可信平台的功能属性、性能属性、系统状态属性及合法输入属性的运行时监控和自动化异常处理。本文方法保证了运行无异常的嵌入式可信平台满足预定义的系统属性,从而保证系统整体持续可信。当前方法提供的监控属性类别仍不完备,监控开销与监控节点数相关性明显,未来研究将围绕增加属性类别、减少监控开销、完善异常处理机制等方面展开。

参考文献:

- [1] Trusted computing group. TCG specification architecture overview[EB/OL]. http://www.trustedcomputinggroup.org/resources/tcg_architecture_overview_version_14.
- [2] WANG X, MIZUNO M, NEILSEN M, et al. Secure RTOS architecture for building automation[C]//ACM Workshop on Cyber-Physical Systems-Security and/or Privacy. 2015:79-90.
- [3] YU Z, WANG Q, ZHANG W, et al. A cloud certificate authority architecture for virtual machines with trusted platform module[C]//IEEE International Conference on High Performance Computing and Communications. 2015:1377-1380.
- [4] TIAN J, BUTLER K, MCDANIEL P, et al. Securing ARP from the ground up[C]//The 5th ACM Conference on Data and Application Security and Privacy. 2015:305-312.
- [5] SHAO J, QIN Y, FENG D, et al. Formal analysis of enhanced authorization in the TPM 2.0[C]//The 10th ACM Symposium on Information, Computer and Communications Security. 2015: 273-284.

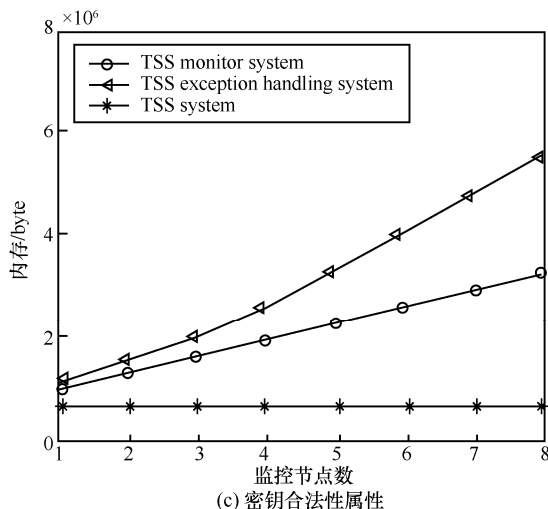
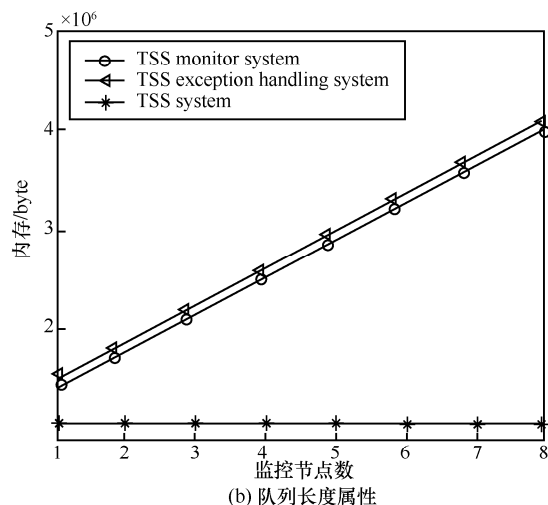
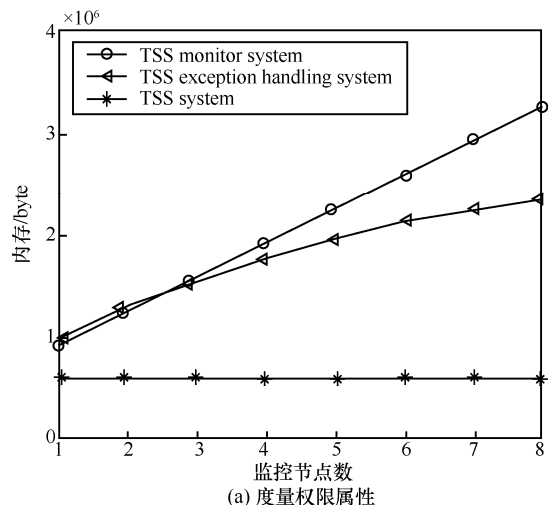
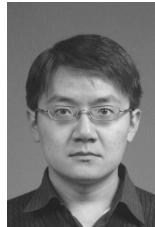


图6 监控系统的内存开销

- [6] WANG W, QIN Y, FENG D. Automated proof for authorization protocols of TPM 2.0 in computational model information security practice and experience[M]. Berlin: Springer. 2014: 144-158.
- [7] DELAUNE S, KREMER S, RYAN M D, et al. A formal analysis of authentication in the TPM[C]//The International Conference on Formal Aspects of Security and Trust. 2010: 111-125.
- [8] BAIER C, KATOEN J. Principles of model checking[M]. London: The MIT Press, 2008: 595-663.
- [9] NAVABPOUR S, JOSHI Y, WU W, et al. RiTHM: a tool for enabling time-triggered runtime verification for C programs[C]//Joint Meeting on Foundations of Software Engineering. 2013: 603-606.
- [10] BIEMAN J M, DREILINGER D, LIN L. Using fault injection to increase software test coverage[C]//The Seventh International Symposium on Software Reliability Engineering, IEEE Computer Society. 1996:166.



马建峰（1963-），男，陕西西安人，西安电子科技大学教授、博士生导师，主要研究方向为密码学、无线和移动安全。



孙聪（1982-），男，陕西兴平人，博士，西安电子科技大学副教授，主要研究方向为信息流安全、可信软件。

作者简介：



孙召昌（1992-），男，河北邢台人，西安电子科技大学硕士生，主要研究方向为可信计算、运行时监控。



卢笛（1983-），男，陕西安康人，博士，西安电子科技大学讲师，主要研究方向为云计算、网络与系统安全。