OpenAnolis
龙 蜥 社 区

OpenAnolis RISC-V Trusted Computing Technology Practices White-Paper

龙蜥社区RISC-V可信计算技术

实践白皮书

浪潮信息龙蜥联合实验室

版权声明

编写者

（排名不分先后）

参编单位：浪潮电子信息产业股份有限公司、睿思芯科（深圳）技术有限公司、山东博算智新信息科技有限公司、中科方德软件有限公司

参编组织：龙蜥社区

参编人员：苏志远、何伟、徐峥、尹奋强、刘雁鸣、曹柱、徐潇、张百林、许荣飞、任清源、张刚诚、张伟、李拓、邹晓峰、王长红、曾廷、冯建茹、冯倩倩

目录

认识龙蜥

龙蜥社区（OpenAnolis）是立足中国面向国际的 Linux 服务器操作系统开源根社区，引领云智融合技术浪潮下国产操作系统的创新发展。

秉承“平等、开放、协作、创新”的原则，社区理事会由阿里云、中兴通讯、海光信息、Intel、浪潮信息、统信软件等 25 家国内外头部企业共同组成。社区生态伙伴超 1000 家，来自芯片厂商、软件厂商、整机厂商、操作系统厂商等覆盖操作系统全产业链参与生态共建。龙蜥操作系统装机量达 1000 万，服务了金融、通信、政务、能源、交通等众多行业 243 万用户。

龙蜥项目运作模式

龙蜥社区已成立 60+ 个 SIG 工作组，围绕芯片、内核、编译器、安全、虚拟化及云原生等操作系统核心领域进行技术创新，已发布 Anolis OS 23、Anolis OS 23.1 GA 版、Anolis OS 8.8、LoongArch GA 等多个社区版本，超 2 万名开发者参

与贡献。为应对 CentOS 停服，官网已上线「CentOS 停服专区」为用户提供迁移方案及长期稳定支持，为广大 CentOS 迁移用户保驾护航。

龙蜥运营管理

为更好地运营和治理社区，龙蜥社区定期召开理事大会，月度运营委员会会议、技术委员会会议。关于龙蜥社区第六届理事大会：会上全票通过了龙蜥技术委员会（TC）附属章程修订提议，理事长和副理事长分别做了 2024 年工作汇报和 2025 年龙蜥社区的规划 。

龙蜥开放的生态

为了鼓励合作伙伴在社区探索出更多的商业合作方式，真正牵引企业在龙蜥社区的合作落地，社区推出「龙腾计划 3.0」——「AI 引擎生态加速合作计划正式发布」，标志着龙蜥社区在推动操作系统与 AI 技术创新融合、以及利用 AI 驱动上下游产业合作变革方面迈出了重要的一步。

关于龙蜥操作系统 （ Anolis OS ）

龙蜥操作系统（Anolis OS）搭载了 ANCK 版本的内核，性能和稳定性经过历年“双 11”历练，能为云上典型用户场景带来 40% 的综合性能提升，故障率降低 50%，兼容 Linux 生态，提供平滑的 CentOS 迁移方案，并提供全栈国密能力。

最新 Anolis OS 23 版本已发布，新增对智能计算的全面支持，内置 rpm 格式的 AI 组件、主流 AI 框架 tensorflow2、pytorch ，支持一键安装 nvidia GPU 驱动、CUDA 库等。龙蜥社区将基于 Anolis OS 23 构建 AI 容器镜像生态，提供主流

的 AI 训练/推理镜像，并发布开箱即用的 modelscope / huggingface AI 大模型实践镜像，稳步提升 AI 的支持蓝图。

下载体验链接：https://openanolis.cn/download

2021 年 12 月 31 日，龙蜥开源社区（OpenAnolis）上线「CentOS 停服专区」，为受 CentOS 停服影响的用户提供迁移方案及长期稳定支持。此次停服，龙蜥操作系统（Anolis OS）产品优势包括：打造系统化解决方案 AOMS、提供多款配套工具、承诺 10 年技术支持、兼容 CentOS 生态、具备差异化核心技术优势、历经丰富场景验证、沉淀用户迁移案例实践。

反馈与共创

OpenAnolis 是一个开放包容的社区，因此我们也欢迎志同道合之士参与我们的文档修订。

对于文档中您认为不足之处，欢迎到我们的官方仓库 RISC-V 可信计算技术实践白皮书新开 issue，我们会第一时间进行响应。

另外，若您想更新文档，也同样欢迎在 RISC-V 可信计算技术实践白皮书提 PR。

1. 可信计算概述

可信计算是通过检测和强化实体行为的预期性来保障实体信任的技术，是一种从体系结构着手解决信息系统安全问题的技术理念；其基本思想是先在计算机系统中建立一个信任根（基），信任根的可信性由物理安全、技术安全与管理安全共同确保；再建立一条信任链，从信任根开始，到硬件平台，到操作系统，再到应用，一级度量认证一级，一级信任一级，把这种信任扩展到整个计算机系统。 可信并不等同于安全，但可信是安全的基础，因为安全组件、策略只有运行在可信的环境下才能进一步达到安全目的。通过系统和安全组件的完整性保障，可以确保业务应用使用正确的软件栈，并在软件栈受到攻击发生改变后能及时发现。总之，在系统和应用中引入可信计算能够极大地降低由于使用未知或遭到篡改的系统/软件遭到攻击的可能性。

1.1 可信计算应用发展现状

可信计算概念最早可追溯到 1983 美国国防部的 TCSEC 准则及之后出现的彩虹系列信息系统安全文件。1999 年，IBM、微软、Intel 等企业成了 TCPA（2003 年改名为 TCG），主要致力于形成可信计算的工业标准。目前 TCG 已经制定了包括 TPM、TSS、TNC 等一系列技术规范，并形成了针对 IoT、云计算、个人终端、移动终端、存储等不同场景的工作组，致力于可信计算在相关场景的应用标准编制与解决方案推动。

国际上已经形成以 Trusted Computing Group（TCG）为代表的可信计算组织，推动 TPM 规范在 PC、服务器、移动终端、网络、云计算、物联网等领域的应用。

在服务器及云计算领域，国际 IT 巨头已将可信计算技术作为其产品的重要支撑。Intel 服务器 CPU 已经全面支持 TPM2.0；微软 Windows Server 2012+ 已支持 TPM2.0，支持可信云计算环境的构建；Linux Kernel4.0 已经集成 TPM2.0，以及主流虚拟化软件 Xen、KVM、Openstack、VMware 等都提供了对 TPM 和 vTPM 的支持；IBM 收购的 Softlayer 公司为全球 60 多个重要客户提供可信云主机服务；同时，许多芯片公司都将部分可信计算功能集成到商用的处理器中，如 ARM 的 TrustZone 技术、 Intel 的 SGX 技术和 AMD 公司的 SEV（secure encrypted virtualization）技术等, 都在处理器中实现了内存隔离, 可以为上层应用提供安全的执行环境, 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。

1.2 可信计算关键技术

1.2.1 信任根概述

信任根是可信计算机系统可信的基础，可信计算平台包含三个信任根：可信度量根(Root of Trust for Measurement, RTM)、可信报告根(Root of Trust for Report, RTR)、可信存储根(Root of Trust for Storage, RTS)。如下图所示：

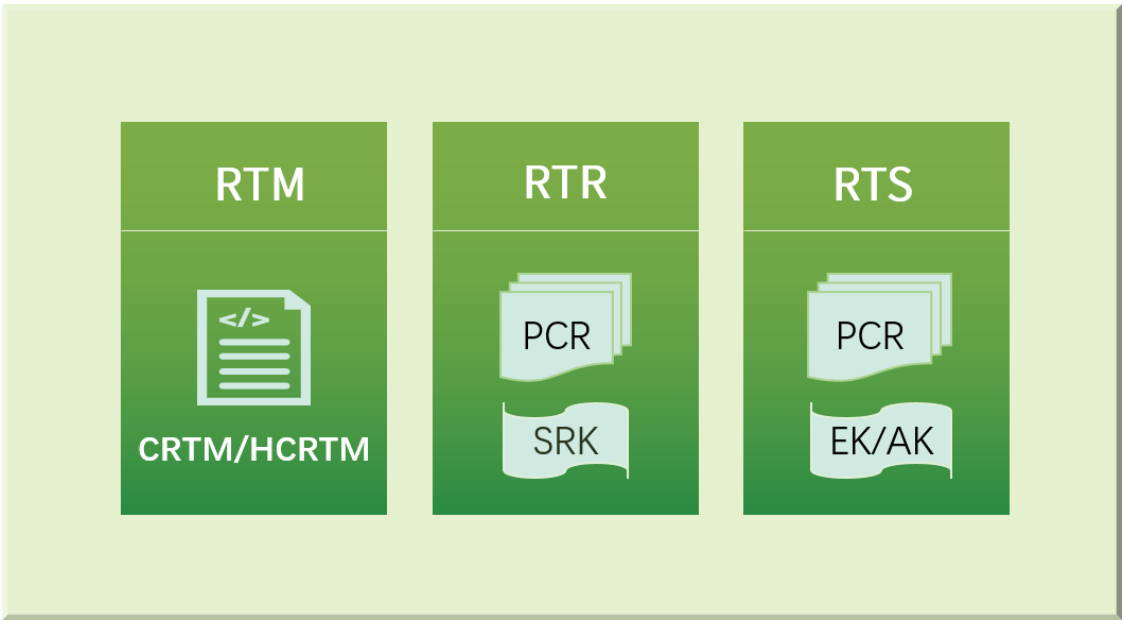

图 1-2-1 可信计算平台信任根构成

度量存储报告机制是可信计算平台重要工作机制，这一机制是指“对可信计算平台进行度量并将度量产生的可信度量值进行存储，当访问客体询问时提供可信报告”，这一机制是确保可信计算平台自身可信并向外提供可信服务的基础。RTM 是对可信计算平台进行度量的信任基点，它的本质是一段代码；RTS 是平台可度量值可信存储的基点，它是由信任根（如 TPM/TCM）中的平台状态寄存器（PCR）和信任根中的存储根密钥（SRK）共同构成；RTR 是平台向访问客体提供可信报告的信任基点，它是由 PCR 和信任根中的背书密钥（EK）共同构成。它们（RTR/RTM/RTS）共同构成了可信计算平台的信任基础。

1.2.2 可信度量与信任链

可信计算中采用对客体哈希的方式实现客体完整性值的采集，通过实时采集到的完整性值与客体完整性基准值比对的方式验证客体的可信状态。对客体进行哈希、

将哈希结果扩展到可信根、并记录事件日志的过程称为可信度量或度量事件。 如下图所示。

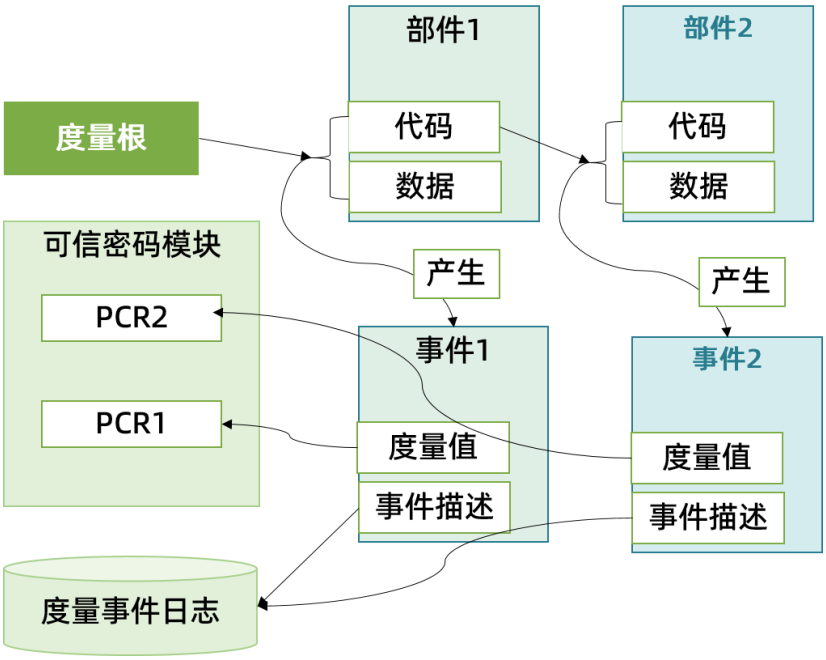

图 1-2-2 可信度量事件示意图

1.2.3 可信计算平台

可信计算平台是指具有可信计算安全机制并能够提供可信服务的计算平台.其主要特征是：有信任根，并基于此构建信任链机制、具有度量存储报告机制、能够提供确保系统数据完整性、数据安全存储和平台远程证明等可信功能服务[4]。典型的可信计算平台包括:可信 PC、可信服务器、可信 PDA 等。

可信计算平台以 CRTM（可信度量根核）为起点，以信任链的方式来度量整个平台资源的完整性，将度量值扩展到可信根的平台配置寄存器中，并通过可信芯片向询问平台可信状态的实体提供报告，供访问者判断平台是否可信，以决定是否可

以与其交互。这种工作机制被称为可信度量、可信报告和可信存储机制，是可信计算机和普通计算机在安全机制上的最大区别。

1.2.4 可信软件栈

可信软件栈是为了方便操作系统层面的安全应用访问可信芯片提供的服务，从而实现为用户提供可信服务。

- 位于用户应用软件与可信芯片之间，作为使用可信芯片的入口。
- 提供安全芯片访问、安全认证、密码服务和资源管理等功能。
- 为应用程序提供一套函数接口来同步访问可信芯片的功能 。
- 对安全芯片自身有限的资源进行管理；管理多个应用程序访问可信芯片资源的请求，提供可信全芯片的并发访问。
- 以合理的字节流顺序及赋值对上层应用隐藏内部命令处理过程 。
- 解决可信芯片自身接口的复杂性和对外服务的不便性 。
- TPM 对应可信软件栈 TSS（TCG Software Stack）。
- TCM 对应 TCM 服务模块 TSM（TCM Service Module）。

1.2.5 远程证明

远程证明是指可信计算平台向外部实体证明平台身份及可信状态的过程。相对于基于身份的认证机制，远程证明进一步扩展和丰富了认证的内容，使得认证的实体能够对认证客体进行更深层次更细粒度的认证。有效的避免了基于身份认证的过程中对实体安全状态一无所知的窘境。远程证明不仅对用户和平台身份进行了认证、

还进一步的对平台的安全状态、软硬件配置等信息进行了验证，保证了平台的状态符合预期的安全策略，从源头上消除了大量潜在的安全攻击。

基于 TPM/TCM 的远程证明与上述目标一致，是一种验证计算机系统软硬件配置正确性的方法。在这个方法中，验证者向系统发送挑战查询，选择要验证的范围。证明者收集描述系统的引导状态、当前配置和身份的证据。背书密钥用于签署证明证据，其公钥会传递给验证者。验证者使用密码学方式检查证据是否满足其信任评估策略，并确定设备是否来自期望的制造商。

2. 龙蜥操作系统可信计算现状

龙蜥操作系统以开源社区优秀成果为上游集成了模拟可信根、可信软件栈、可信工具集，满足用户对可信计算基础服务的需求。同时，可信计算 SIG 对可信根增强引擎等可信计算基础软件进行了探索分析，形成了龙蜥操作系统可信计算实践指南相关内容。未来，针对可信计算基础软件全栈国密支持与实践探索也将是可信计算 SIG 的重点工作。

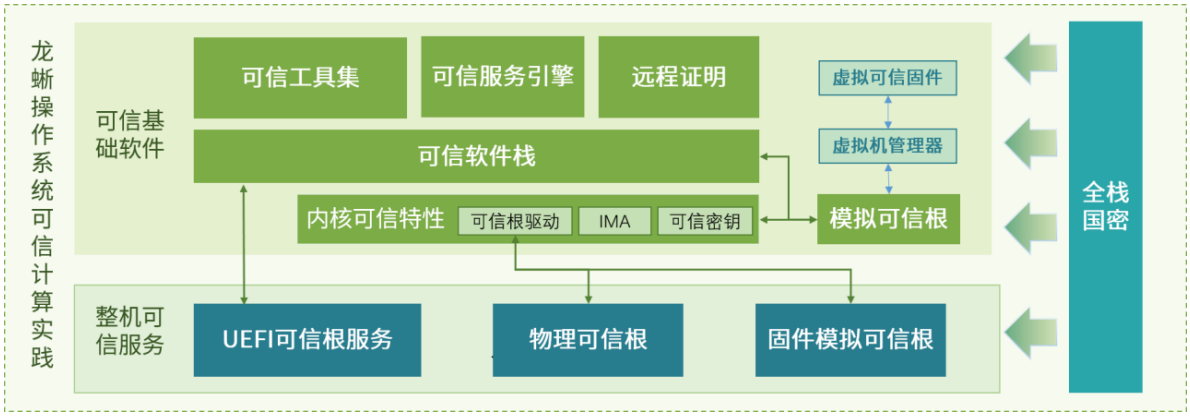

图 2-1 龙蜥操作系统可信计算架构图

2.1 内核可信特性

2.1.1 概述

龙蜥操作系统内核继承了 linux 内核成熟的可信计算特性，包括可信根设备驱动、内核完整性架构（Integrity Measurement Archtecture,简称 IMA）、内核可信密钥与加密密钥等。

2.1.2 可信根设备驱动

- **TPM Eventlog**

本节简要描述了什么是 TPM Eventlog（即度量事件日志），以及如何将 TPM Eventlog 从预引导固件传递到操作系统。

1） TPM Eventlog 概述

固件（如 BIOS/UEFI 等）维护了一个 TPM Eventlog，每当固件度量新的组件并将度量结果扩展到任何 PCR 寄存器时，该日志都会新增一条记录。按事件类型分隔，并包含了扩展后的新 PCR 值。TPM Eventlog 主要用于远程证明，用于向挑战者证明平台的可信状态。TPM Eventlog 的主要作用是提供详细的度量对象和度量结果，远程证明过程中通过包含 PCR 内容的可信报告验证度量日志的可信性。

2） UEFI TPM Eventlog

UEFI 提供了查询 TPM Eventlog 的 protocol 服务。在调用 ExitBootServices() 之前，Linux EFI 会将事件日志复制到由内核自定义的配置表中。 需要说明的是，ExitBootServices()生成的度量结果不会出现在最终的度量日志表中。固件提供了最终事件配置表来解决这个问题。在第一次调用 EFI_TCG2_PROTOCOL.GetEventLog()之后，事件被镜像到最终时间配置表中。

- **vTPM Proxy Driver for linux Container**

1） 面向容器的虚拟 TPM 驱动概述

该特性是为每个 Linux 容器提供 TPM 功能，该特性允许程序以与物理系统上 TPM 交互相同的方式与容器中的 TPM 交互。每个容器都有自己独特的、模拟的软件 TPM。

2） 面向容器的虚拟 TPM 驱动的设计

为了使 swTPM 可用于每个容器，容器管理堆栈需要创建一个设备对，该设备对由一个客户端 TPM 字符设备/dev/tpmX (X=0,1,2...)和一个“服务器端”文件描述符组成。在将文件描述符传递给 TPM 仿真器的同时，通过创建具有适当主编号和副编号的字符设备将前者移动到容器中。然后，容器内的软件可以使用字符设备发送 TPM 命令，仿真器将通过文件描述符接收命令，并使用此文件描述符返回响应。

虚拟 TPM 代理驱动程序提供了一个设备/dev/vtpmx，用于使用 ioctl 创建设备对。ioctl 将其作为配置设备的输入标志。例如，这些标志指示 TPM 模拟器是否支持 TPM 1.2 或 TPM 2 功能。ioctl 的结果是生成“服务器端”的文件描述符以及所创建的字符设备的主次编号，以及返回 TPM 字符设备的编号。例如，如果创建了/dev/tpm10，则返回数字(dev_num) 10。 一旦设备被创建，驱动程序将立即尝试与 TPM 通信。驱动程序的所有命令都可以从 ioctl 返回的文件描述符中读取并立即得到响应。

- **Firmware TPM Driver**

fTPM(Firmware TPM)驱动程序用于支持在 ARM 的 TrustZone 环境中实现的 TPM 设备。驱动程序允许程序以与硬件 TPM 交互相同的方式与 fTPM 交互。

2.1.3 IMA

IMA 是目前使用最为广泛的完整性度量架构。它通过在内核中增加模块，当运行程序运行、内核模块被挂载和动态链接库被加载时，对用到的代码和关键数据(如配置文件和结构化数据) 进行一次度量，并将度量结果扩展到 TPM 的 PCR10(默认) 中，同时创建并维护一个度量列表(Measurement List，ML) 。当远程挑战者发起

挑战时，将度量列表 ML 和用 TPM 签名的 PCR10 中的度量值发送给挑战者，挑战者通过对比度量值和基准值来判断平台是否可信[6]。

2.1.4 内核可信密钥与加密密钥支持

可信密钥和加密密钥是添加到现有内核密钥环服务中的两种新密钥类型。这两种新类型都是可变长度对称密钥，所有密钥都是在内核中创建，用户空间只能看到、存储和加载加密的 blob。可信密钥需要可信源的可用性以获得更高的安全性，而加密密钥可以在任何系统上使用。为了方便起见，所有用户级 blob 都以十六进制 ASCII 格式显示和加载，并经过完整性验证。

- **信任源**

可信源为可信密钥提供安全来源。信任源是否足够安全取决于其实现的强度和正确性，以及特定用例的威胁环境。由于内核不知道环境是什么，也没有信任度量，因此它依赖于可信密钥的使用者来确定信任源是否足够安全。 内核支持硬件可信根（如 TPM/TCM）作为内核可信密钥的可信源。由于 TPM/TCM 提供的 SRK 机制确保了根密钥不会离开可信根、且提供芯片级安全保障机制和高安全等级的密钥熵源。

- **将密钥使用与平台完整性状态绑定**

基于 TPM/TCM 提供的密钥服务，密钥可以选择性地密封到指定的 PCR 值，并且只有在 PCR 和 blob 可信验证通过的情况下，TPM 才会对密钥进行解封。加载的可信密钥可以使用新的(未来的)PCR 值更新，因此密钥很容易迁移到新的 PCR 值，例如当内核和 initramfs 更新时。同一个密钥在不同的 PCR 值下可以保存多个 blob，因此很容易支持多个 boot。

- **接口和 API**

  内核支持 TPM/TCM 访问接口和 API。

2.1.5 内核可信配置参数(KConfig)

内核中有需要可信计算相关的配置参数，下表中列举出部分常用的参数以及解释其含义。

表 2-1-5 内核可信功能配置参数

内核可信配置名称	用途	依赖的 Config(以下省略 CONFIG_开头)
CONFIG_SYSTEM_TRUSTED_KEYS	系统可信 keys, 可以级联	CRYPTO [=y] && SYSTEM_TRUSTED_KEYRING [=y]
CONFIG_SECONDARY_TRUSTED_KEYRING	允许用户从用户空间加入 system trusted key，要求 key 必须是从用户空间加载，且必须由其中一把 key 签过	CRYPTO [=y] && SYSTEM_TRUSTED_KEYRING [=y]
CONFIG_SYSTEM_EXTRA_CERTIFICATE	设置预留区域允许用户手动插入证书而不需要重编内核	CRYPTO [=y] && SYSTEM_TRUSTED_KEYRING [=y]
CONFIG_SYSTEM_EXTRA_CERTIFICATE_SIZE	证书预留区域的大小	CRYPTO [=y] && SYSTEM_EXTRA_CERTIFICATE [=y]
CONFIG_MODULE_SIG	开启后对内核模块的签名进行检查	MODULES [=y]
CONFIG_MODULE_SIG_FORCE	开启后内核会直接拒绝加载签名有问题的内核模块	MODULES [=y] && MODULE_SIG [=y]
CONFIG_INTEGRITY_AUDIT	使能完整性审计支持	INTEGRITY [=y] && AUDIT [=y]

内核可信配置名称	用途	依赖的 Config(以下省略 CONFIG_开头)
CONFIG_IMA	使能 IMA	INTEGRITY [=y]
CONFIG_IMA_WRITE_POLICY	允许对 IMA 策略的多次写入	INTEGRITY [=y] && IMA [=y]
CONFIG_IMA_READ_POLICY	允许读当前 IMA 策略	INTEGRITY [=y] && IMA [=y]
CONFIG_IMA_X509_PATH	IMA x509 证书路径	INTEGRITY [=y] && IMA_LOAD_X509 [=y]
CONFIG_IMA_DEFAULT_HASH	IMA 缺省的 HASH 算法	INTEGRITY [=y] && IMA [=y]
CONFIG_IMA_DEFAULT_TEMPLATE	IMA 缺省的模板	INTEGRITY [=y] && IMA [=y]
CONFIG_IMA_APPRAISE	使能本地度量完整性评估	INTEGRITY [=y] && IMA [=y]

2.2 可信软件栈

2.2.1 可信软件栈概述

TCM Service Module 提供用于访问 TCM 的标准 API。应用程序开发人员可以使用此软件规范来更好地使用 TCM 和开发基于 TCM 的应用程序。

TCG 软件栈 (TSS) 是一种软件规范，提供用于访问 TPM 的标准 API。 应用程序开发人员可以使用此软件规范来更好地使用 TPM 和开发基于 TPM 的应用程序。

2.2.2 各种语言的开源 TPM TSS 软件栈现状

TPM 的软件栈非常繁荣，涉及到多个语言以及多个开源项目。以下表格仅列出部分主流开发语言（C/Go/Java/Python/Rust）的主流 TPM TSS 软件栈现状。

表 2-2-1 各语言 TPM TSS 软件栈现状

TSS 项目	主要贡献者/企业	开发语言	对 TPM1.2/TPM 2.0 的支持情况
tpm2-tss	Intel	C	支持 TPM 2.0
IBM’s TPM 2.0 TSS	IBM	C	支持 TPM 1.2 和 TPM 2.0
go-tpm	Google	Go	支持 TPM 1.2 和 TPM 2.0
go-tpm2	canonical	Go	支持 TPM 2.0
JSR 321	MIT	Java	支持 TPM 1.2
tpm2-pytss	Intel	Python	支持 TPM 2.0
rust-tss-esapi	Arm	Rust	支持 TPM 2.0

2.3 可信工具集 tpm2-tools

tpm2-tools 是一套基于 TSS2.0（Trusted Software Stack，可信软件栈）接口开发的 TPM2 管理工具集合，可用于操作 TPM2.0 芯片实现密码学、可信存储、完整性验证等功能。

- **TPM2.0 基本功能**

  tpm2_startup 工具可执行 TPM2_CC_Startup 命令使能 TPM2.0 芯片。

  tpm2_getcap 工具可执行 TPM2_CC_GetCapability 命令获取 TPM2.0 芯片信息。

- **TPM2.0 密码学功能**

TPM2.0 密钥管理采用加密存储的方式，每一密钥都有父密钥，密钥导出 TPM2.0 芯片时，都被父密钥加密保护，导入 TPM2.0 芯片时又父密钥解密恢复。在加密存储体系中，存在一个根密钥（又称为 PrimaryObject），该密钥无法导出到 TPM2.0 芯片外。TPM2.0 中有三个独立的特权域（Hierarchy），每一特权域都可创建根密钥。

- **TPM2.0 存储功能**

TPM2.0 芯片内置了 NVRAM（Non-Volatile Random Access Memory，非易失性随机访问存储器），用于存放数据。TPM2.0 芯片 NVRAM 读写需要授权，因此可用于存放敏感数据。TPM2.0 NV 空间需要要先定义才能进行读写操作，使用完毕后要释放已定义的空间。

- **TPM2.0 PCR 功能**

TPM2.0 PCR(Platform Configuration Register, 平台配置寄存器)是 TPM2.0 中与完整性相关的信息存储空间。PCR 的更新方法叫做扩展（Extend），扩展是一种单向的加密操作，保证度量值不被篡改。

- **TPM2.0 死锁功能**

TPM2.0 中对象都需要授权访问，使用错误授权访问具有 DA 保护属性的对象（如密钥、没有设置 noDA 属性的 NV 等）会导致死锁计数器加 1，当死锁计数器达到一定数值后，TPM2.0 便拒绝授权访问。 TPM2.0 中与死锁相关的属性有 maxTries(最大允许授权失败次数)、lockoutRecovery（Lockout Hierarchy 死锁恢复时间），recoveryTime(死锁计数器自减一时间间隔)。

2.4 可信服务引擎 tpm2-tss-engine

tpm2-tss-engine 利用遵循可信计算组织（Trusted Computing Group，TCG）的软件栈——TSS2.0,实现了基于 TPM2 设备的 OpenSSL 密码引擎。tpm2-tss-engine 利用 TSS2.0 中的增强型系统 API（Enhanced System Applicaion Service Interface， ESAPI）与 TPM2 设备通信。tpm2-tss-engine 支持 RSA 加解密、签名以及 ECDSA 签名功能。

3. RISC-V 可信计算实践及解决方案

可信计算技术作为一种系统级安全增强技术，例如系统可信验证（动态、静态）、轻量级可信集群构建等，应用得当可以极大的提升操作系统安全能力，这些能力的应用也是 RISC-V SIG 探索可信计算最佳实践的重要方向。本章将分别介绍 RISC-V 可信计算相关知识以及如何在 RISC-V 架构芯片上编译测试来验证可信计算的基础功能，并通过 Ktrusted 来构建轻量级的 RISC-V 高性能场景可信计算实践解决方案。

3.1 RISC-V 可信计算

3.1.1 什么是 RISC-V

RISC-V 是由加州大学伯克利分校开发发布的一种免费、开放的指令集架构标准。其规范由 RISC-V 国际基金会下的技术社区协作开发，并以宽松的开源许可证发布，允许任何人无偿地设计、制造和销售 RISC-V 芯片。

这种开放性本身就是一项关键的安全特性。它允许整个架构设计，从指令集手册到具体的硬件实现，都可以被公众，包括学术界、行业专家和安全研究人员，进行彻底的审查。这种透明度从根本上消除了硬件设计中存在未公开指令、隐藏后门或恶意逻辑的可能性。在专有架构中，用户必须信任供应商提供的“黑盒”，其内部运作和安全保证无法被独立验证。而在 RISC-V 生态系统中，信任建立在可验证的、公开的工件之上。这种从“因信而信”到“因证而信”的转变，是 RISC-V 安全理念的核心，它将信任的基础从对单一供应商的依赖转移到了一个可由全球社区共同审计和验证的开放标准之上。

近年来，随着 RISC-V 指令集架构处理器在云计算、高性能、AI 等场景的应用落地，针对安全、可信等功能实现的扩展指令和硬件模块被开发和设计出来。并通过标准的建立推动 RISC-V 安全、可信生态系统向成熟发展。

3.1.2 安全启动

在安全启动链之上，标准化的固件接口和组件间认证协议对于构建一个可互操作且安全的生态系统至关重要。

- **现代固件取代传统 BIOS：**RISC-V 平台不使用传统的 BIOS。取而代之的是一个更现代、模块化的固件栈，其核心组件通常包括 OpenSBI 和 U-Boot。OpenSBI（RISC-V Supervisor Binary Interface）项目为运行在监管者模式（S-mode）的操作系统提供了一个标准接口，用以访问由机器模式（M-mode）固件提供的底层平台服务。在 OpenSBI 完成底层初始化后，通常会转交控制权给像 U-Boot 这样的通用引导加载程序，由后者负责加载操作系统内核。
- **UEFI 的引入与安全启动：**为了与主流服务器和桌面生态系统对齐并解决平台碎片化问题，RISC-V 社区正在积极引入 UEFI（统一可扩展固件接口）。RISC-V 国际基金会正在制定的平台规范已将 UEFI 作为服务器平台的一项要求。社区正在将 UEFI 的开源参考实现（EDK2）移植到 RISC-V 平台。UEFI 的一个核心安全功能是安全启动（Secure Boot），它通过验证引导加载程序和操作系统的数字签名来确保只有受信任的代码才能在启动过程中执行，从而将信任链从硬件 RoT 扩展到整个软件栈。
- **SPDM 组件认证：**安全协议和数据模型（SPDM）是由 DMTF 制定的一个行业标准，它独立于任何特定的处理器架构。SPDM 为平台上的不同组件（如主 CPU、智能网卡、基板管理控制器等）之间提供了一种标准化的认证和安全通

信协议。通过 SPDM，平台可以在启动或运行时验证各个组件的身份和固件完整性，确保它们未经篡改，从而将这些组件安全地纳入系统的信任边界。RISC-V 生态系统正在采纳该技术，并已有相关的实现项目。

3.1.3 可信启动

目前，RISC-V 架构对基于 TPM 的可信计算与度量机制的支持呈现出多元化的现状。

在具体实现上，存在三种并行模式。首先是 dTPM，尽管已标准化，但商用硬件的普及尚处于早期阶段，当前软件生态的开发主要依赖 QEMU 等仿真环境推进 。其次是固件 TPM（fTPM），以 RfTPM 等研究项目为代表，它利用 RISC-V 原生的机器模式（M-mode）和物理内存保护（PMP）等标准特性，在无需额外硬件的条件下提供了功能完整的 TPM 实现，并在部分密码学运算上表现出性能优势 。最后，针对资源极度受限的物联网设备，已有利用 PMP 构建的轻量级证明机制（如 LIRA-V），实现了核心的远程证明功能 。

软件生态方面，U-Boot 等主流引导加载程序已支持度量启动流程 。同时，Linux 内核拥有成熟且独立于硬件架构的 TPM 子系统，使得 tpm2-tools 和 systemd-cryptenroll 等标准工具链可直接应用于 RISC-V 平台，与底层硬件或固件实现对接 。

3.1.4 加密扩展

为了高效地执行现代密码学算法，RISC-V 标准化了一套标量加密扩展。该扩展集于 2021 年 11 月被正式批准，并已并入非特权 ISA 规范中。这些指令被设计为轻量级，适用于从嵌入式微控制器到高性能应用处理器的各种场景。

该扩展集的设计哲学并非提供执行完整加密算法的单一指令，而是提供加速算法中计算最密集部分的“原语”指令。例如，它不提供一条“AES 加密一个块”的指

令，而是提供多条指令分别对应 AES 轮函数中的不同步骤。这种方法保留了软件的灵活性，使其能够实现不同的加密模式（如 CBC, GCM）或利用这些原语构建新的算法，同时又能获得显著的性能提升。

3.1.5 高性能场景

- **RVA23 标准的诞生背景**

RISC-V 架构凭借模块化与可扩展性优势在嵌入式领域快速渗透，但早期“高度定制化”的发展模式导致指令集扩展碎片化严重。当 RISC-V 向服务器、桌面、移动设备等主流高性能市场进军时，二进制软件兼容性不足的问题日益凸显 —— 不同厂商的处理器实现因支持的扩展组合各异，导致应用程序需针对特定硬件重新编译，极大增加了开发成本与生态协同难度。为解决这一痛点，RISC-V 国际基金会（RVIA）于 2024 年 Q3 正式发布 RVA23 Profile 规范，标志着 RISC-V 从“碎片化定制”向“标准化通用”的战略转型。

- **RVA23 标准的核心描述**

作为面向 64 位应用处理器的标准化配置规范，RVA23 Profile 明确了运行 Linux、Android 等富操作系统所需的“强制扩展集 + 可选扩展”组合框架。其最显著的变革是将 Vector（向量）扩展列为强制要求，同时纳入 Hypervisor（虚拟化）、浮点运算等关键特性，形成高性能计算的基础能力底座。在安全与加密能力方面，RVA23 摒弃了原 RVA22 中的标量加密扩展 Zkn/Zks，转而强制推动向量加密方案（如 Zvkng/Zvksg），通过向量计算的并行优势实现加密性能的提升。标准同时定义了用户模式（RVA23U64）与监督者模式（RVA23S64）的具体要求，为系统级安全隔离提供了架构依据。

- **RVA23 标准的关键作用**

对软件开发者而言，统一的扩展基准大幅降低了跨平台适配成本，Google、Red Hat 等厂商已围绕 RVA23 构建 Android 与 Linux 系统支持体系，加速软件生态成熟。在技术层面，Vector 扩展的强制化使 RISC-V 在 AI 推理、图像处理等计算密集型场景具备竞争力，而 Hypervisor 扩展则为云原生与虚拟化部署奠定基础。更重要的是，通过规范硬件特性基线，RVA23 解决了“可选扩展使用率低” 的行业困境，让高性能与安全特性得以在生态中充分落地。

- **高性能场景可信计算**

RVA23 标准为“高性能场景可信计算”提供了不可或缺的标准化根基。高性能场景对可信计算的核心需求 —— 如硬件级安全隔离、高效加密运算、虚拟化环境可信度量等，均能在 RVA23 的架构框架中找到技术支撑：其强制的向量加密扩展为大规模数据加密提供了性能保障，避免可信计算引入的性能损耗成为系统瓶颈；Hypervisor 扩展与 IOMMU、PMP 三级内存保护机制的结合，可构建从硬件到虚拟化层的可信隔离边界，适配多租户云环境的安全需求；而统一的硬件特性基线则确保了可信计算框架（如远程证明、安全启动）能跨厂商硬件实现标准化部署，避免因架构碎片化导致的可信方案兼容性问题。

3.2 软件编译说明

安装 OS 为 Anolis 系列镜像时，功能测试不需要额外安装软件包，可跳过软件编译说明部分直接进行功能测试即可。

安装 OS 镜像为开源版本时，功能测试需要另外安装测试软件包或模块：tpm2、tdm、grub。注：完整的可信启动信任链包含 grub 度量 OS 内核，支持 tpm2 的 grub 版本需 2.04 或以上，同时需将 grub tpm 模块安装进 grub 内核。

3.3 TPM2.0 功能测试

3.3.1 概述

本章节演示了在一台安装 Anolis OS 的服务器上测试 TPM2.0 功能的完整步骤。演示的主要 TPM 应用包括：BIOS/GRUB/Linux 启动度量，商密测试。

说明：

- 发行的 Anolis OS 已包含 TPM2.0 功能在内的安全功能适配支持，安装后用户可以直接测试使用。
- 支持 tpm2 商密的 tss 和 tools 最低版本分别是 2.3.2-4.0.2 和 4.1.1-5.0.3，如果不满足请更新版本。

表 3-3-1 配置要求

配置	要求
CPU	龙蜥社区支持的 RISC-V 架构芯片
BIOS	PI 版本为 2.1.0.3 或以上，支持 TPM2.0
GRUB	2.04
KERNEL	6.6
TPM2-tss	tpm2-tss-2.3.2
TPM2-abrmd	tpm2-abrmd-2.3.3
TPM2-tools	tpm2-tools-4.1.1

3.3.2 BIOS 安装及 TPM 设置

进入 BIOS 设置，在 TCG Trusted Computing 选项下关于 TPM 的默认设置如图 3-3-1 所示，度量使用 SM3 算法，三个 Hierarchy(Platform, Storage, Endorsement)全部使能，如果没特殊需求，使用默认设置即可。BIOS 中开启 TPM 的设置在不同的 BIOS 下可能会有差异，具体请咨询相应的 BIOS 厂商。以下设置仅供参考。

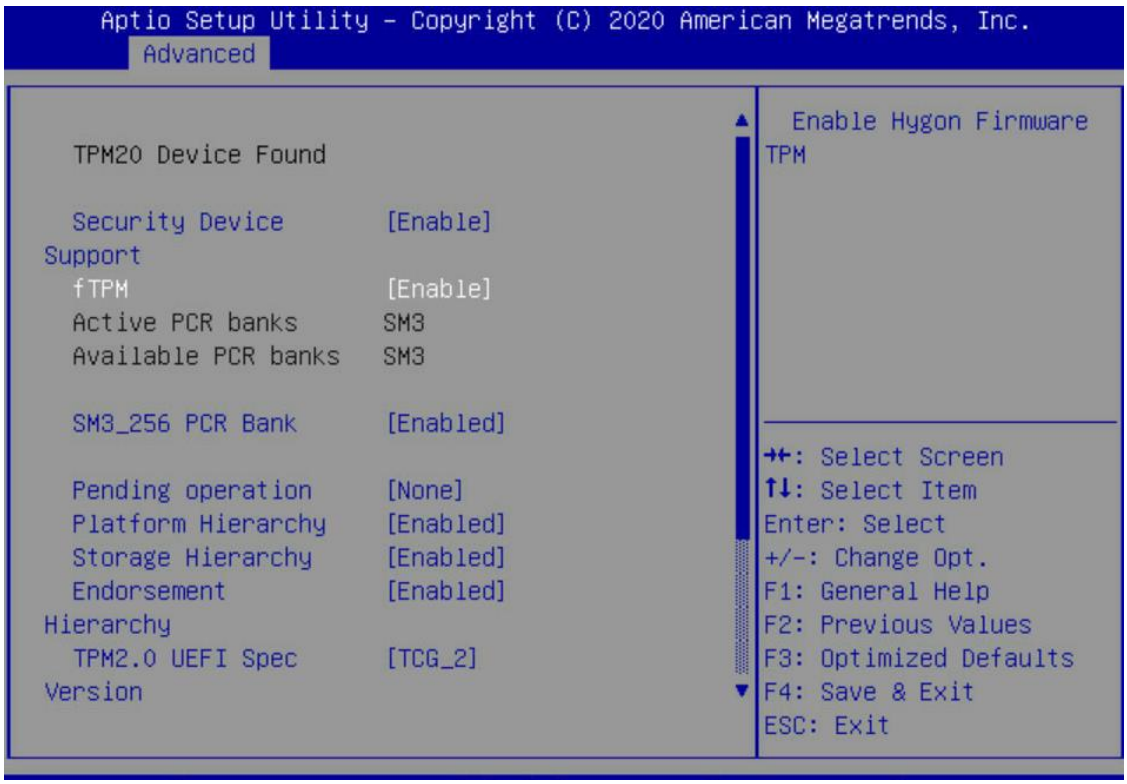

图 3-3-1 BIOS 设置

3.3.3 依赖软件包

tpm2.0 依赖的软件栈或者安装包等都已经集成进 ISO，如果当前环境缺少对应安装包，可直接使用 yum 安装。

需要注意的是 tpm 驱动默认以 ko 模块的方式集成到 kernel 中，如果需要测试 IMA 功能，则需要将 TPM 驱动编译进内核，不能以模块的方式加载。如果不需要使用 IMA 则可以忽略。

3.3.4 配置

安装相应安装包后，在测试 tpm2 功能之前，需要设置如下配置并启动 tpm2-abrmd 服务：

```
$ sudo useradd --system --user-group tss
$ sudo udevadm control --reload-rules && udevadm trigger
$ sudo pkill -HUP dbus-daemon
$ sudo systemctl daemon-reload
$ sudo ldconfig
$ sudo systemctl enable tpm2-abrmd
$ sudo chown tss:tss /dev/tpm0
$ sudo service tpm2-abrmd start
$ systemctl status tpm2-abrmd.service
```

3.3.5 BIOS/Grub/Linux 内核启动度量

启动度量对 TPM PCR 的使用情况如下表：

表 3-3-2 TPM PCR 的使用情况

PCR 编号	度量目标
0	BIOS，包括作为度量根的初始代码
1	BIOS 平台配置
2	OPTION ROM 代码
3	OPTION ROM 配置及数据
4	IPL/Grub 代码
5	IPL/Grub 配置及数据
6	STATE_TRANSITION
7	平台厂商相关控制

PCR 编号	度量目标
8	GRUB 执行的命令字符串，包括配置文件内命令和命令行输入
9	GRUB 打开的文件，包括配置文件，efi 子模块，Linux 内核，Initrd 等
10	Linux IMA 使用

系统起来后使用工具 tpm2_pcrread 读取 PCR。根据 PCR 使用分配，在不改变 BIOS/Grub 配置的情况下，每次系统启动后 PCR 0~9 读值应保持不变，如下：

```
[root@localhost ~]$ tpm2_pcrread sm3_256:0,1,2,3,4,5,6,7,8,9,10,11
sm3_256:
  0 : 0x7A3C9E2B5D8F104632587A9C0E1F3B5D792A4C6E8013579B2D4F6A8C0E214365
  1 : 0x2D5F8A1C3E6B90471369258A0C3E5B7F912468A0C3E5B7F921468A0C3E5B7F92
  2 : 0x9C0E21436587A9C0E1F3B5D792A4C6E8013579B2D4F6A8C0E21436587A9C0EF3
  3 : 0x5B7F912468A0C3E5B7F921468A0C3E5B7F921468A0C3E5B7F92148A0C3E5B7F9
  4 : 0x5C7B9A2D4F6E81305A7C9B2D4F6E81305A7C9B2D4F6E81305A7C9B2D4F6E8130
  5 : 0x9B2D4F6A8C1E30527496B8D0F2A4C6E813579B2D4F6A8C0E21436587A9C0D2F6
  6 : 0x2F4A6C801D3B5E792F4A6C801D3B5E792F4A6C801D3B5E792F4A6C801D3B5E79
  7 : 0x6E81305A7C9B2D4F6E81305A7C9B2D4F6E81305A7C9B2D4F6E81305A7C9B2D4F
  8 : 0x4D6F801B3E5A7C9D2F4A6C801B3E5A7C9D2F4A6C801B3E5A7C9D2F4A6C801B3E
  9 : 0x7A9C0D2F4B6E81357A9C0D2F4B6E81357A9C0D2F4B6E81357A9C0D2F4B6E8135
  10: 0x0000000000000000000000000000000000000000000000000000000000000000
  11: 0x0000000000000000000000000000000000000000000000000000000000000000
```

商密支持的测试

- **商密测试脚本**

包含商密测试的脚本已入库龙蜥仓库，具体见 tpm2-tools 仓库 a8 分支，patch 名称：0001-add-gm-test-case-for-all-commands.patch。

- **运行测试**

命令正确执行的结果如下：

```
[root@localhost tests_gm]$ ./test.sh
test_tpm2_activatecredential.sh ... PASSED
test_tpm2_attest.sh ... PASSED
test_tpm2_changeauth.sh ... PASSED
test_tpm2_clock.sh ... PASSED
test_tpm2_encryptdecrypt.sh ... PASSED
test_tpm2_hash.sh ... PASSED
test_tpm2_nv.sh ... PASSED
test_tpm2_pcr.sh ... PASSED
```

```
10. test_tpm2_policy.sh ... PASSED
11. test_tpm2_random.sh ... PASSED
12. test_tpm2_selftest.sh ... PASSED
13. test_tpm2_sign.sh ... PASSED
14. Tests passed: 12
15. Tests Failed: 0
```

- **测试脚本说明**
  - 每个以 test 开头的脚本是测试 TPM 的一类脚本,比如测试 TPM policy 相关命令的脚本名字为 test_tpm2_policy.sh。
  - 脚本中测试命令都是测试商密的命令：
    - TPM 算法解析是 object:scheme: symdetail 格式的字符串，比如 tpm2_createprimary -C o -g sm3_256 -G eccsm2:null: sm4128cfb -c /tmp/context；
    - 字符串 eccsm2:null:sm4128cfb 是在上面格式的基础上添加商密相关的字符串 sm2，sm4128cfb 以支持商密；
    - 在某些命令中需要增加额外的参数支持商密选择，同时保持兼容以前的密码算法，具体命令可参考脚本中 tpm2_certify，tpm2_createak，tpm2_createek，tpm2_createprimary，tpm2_loadexternal，tpm2_nvdefine，tpm2_quote，tpm2_startauthsession 的使用。

3.4 KTrusted：基于 RISC-V 的轻量级可信增强解决方案

3.4.1 概述

KTrusted 是龙蜥商业操作系统 KOS 操作系统的核心可信组件之一，用于实现运行时动态可信度量与验证机制。其目标是在系统启动后，持续对内核态与用户态

关键对象进行度量、记录和验证，从而保证系统在整个生命周期内处于可验证、可恢复、可追溯的可信状态。

KTrusted 可在不同硬件信任根（如 TPM 2.0、TPCM 或 RISC-V 定制安全引擎）上运行，并通过 KOS 内核的可信接口与度量框架（IMA/EVM 兼容）协同工作，实现端到端的动态可信保护链。

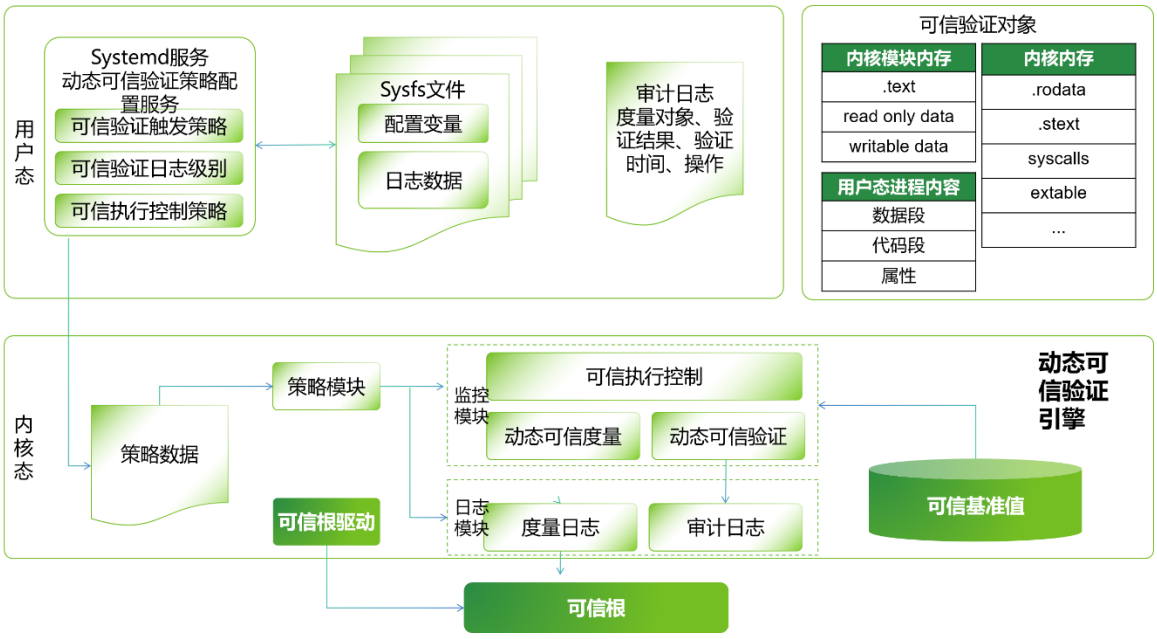

图 3-4-1 KTrusted 可信增强软件架构图

- **技术特点与优势**

1. 持续可信：相较于传统静态度量方案，KTrusted 支持系统运行过程中的动态度量与实时验证，可监控驱动加载、动态链接库注入、内核模块更新等关键行为。

2. 轻量低侵入：采用 eBPF 与内核钩子机制实现零修改集成，CPU 与 I/O 开销 < 3%。

3. 国密算法原生支持：全部度量值与远程证明过程可基于 SM2/SM3/SM4 算法执行，满足等保 2.0 与关保要求。
4. 可扩展信任根接口：兼容 TPM2.0、TPCM 与 RISC-V 安全引擎
5. 可视化与可审计：度量链、策略与验证结果均可通过 KOS 安全控制台展示与审计，支持日志签名与溯源。
6. 生态兼容性强：对 IMA/EVM、SPDM 协议、Linux Security Module (LSM) 接口保持兼容，可无缝适配 RISC-V 生态。

3.4.2 RISC-V 平台下的 KTrusted 适配与实现

在 RISC-V 平台上，KTrusted 基于以下体系实现可信增强：

- **启动阶段集成**：通过 OpenSBI 与 Boot Loader 阶段的 hash 测量结果初始化 KTrusted 基准值。
- **内核接口层**：利用 RISC-V 特有的 PMP（Physical Memory Protection）和 SME（Secure Memory Extension）指令集实现关键度量区保护。
- **安全引擎协同**：可调用 RISC-V SoC 上的 Crypto Co-Processor 或 Caliptra 兼容安全模块完成 SM3/SM4 加速度量。
- **远程证明扩展**：提供 SM2 签名与 SPDM 协议兼容接口，用于可信节点的跨平台互认证。

表 3-4-1 动态可信增强软件安装环境要求

硬件环境要求		
服务器	系统启动固件	支持 TPM2.0
	部件	TPM2.0/TCM2.0

软件环境要求		
操作系统	KOS5.10	6.6.71 及以上

动态可信验证特性提供基于可信根的内核内存关键数据动态可信验证，主要对以下内核内存关键数据进行动态可信验证。

表 3-4-2 动态可信验证对象列表

对象	说明
ex_table	异常表，存放了内核服务访问进程地址空间的每条指令的地址。该表存放在内核代码段的__ex_table 节，其起始和终止地址由 C 编译器产生的两个符号__start__ex_table 和__stop__ex_table 来标识。 在 linux 内核链接脚本文件中将每个目标文件中的__ex_table 节合并，并定义了__start__ex_table 和__stop__ex_table 来标识。
stext 段	用于存放程序代码的，编译时确定,只读
rodata 段	该段也叫常量区，用于存放常量数据
内核 Unix 网络协议族地址	指向内核中提供网络协议栈访问的 ops 数据结构，如 unix_family_ops 等

内核 inet6 协议地址	指向内核中提供网络协议栈访问的 ops 数据结构，如 af_inet_ops 等
中断数据	中断数据
module（内核模块）	内核模块，模块中的信息都是通过 module 结构体获取. module 是一个链表，通过遍历链表，获取所有的模块的信息，用于完整性检测的主要是检测状态为 MODULE_STATE_LIVE 的模块.度量内容主要为：core_layout 和 core_layout.text

可信验证触发机制方面，提供固定周期触发可信验证、内核事件触发可信验证及随机周期触发可信验证等三种方式。

安装使用

rpm -ivh KTrusted-1.0.2-1.noarch.rpm

```
[root@keyarchos ~]# rpm -ivh KTrusted-1.0.2-1.noarch.rpm
Verifying...                          ################################# [100%]
Preparing...                          ################################# [100%]
Updating / installing...
   1:KTrusted-1.0.2-1                 ################################# [100%]
```

启动、查看状态及关闭动态可信验证服务命令如下：

1. systemctl start ktrusted # 启动服务
2. systemctl enable ktrusted # 设置开机自启动
3. systemctl status ktrusted # 查看服务状态
4. systemctl stop ktrusted # 停止服务

```
To start KTrusted DIM please use: systemctl start ktrusted
To enable KTrusted DIM on bootup please use: systemctl enable ktrusted
[root@keyarchos ~]# systemctl start ktrusted
[root@keyarchos ~]# systemctl enable ktrusted
Created symlink /etc/systemd/system/sysinit.target.wants/ktrusted.service → /usr/lib/systemd
[root@keyarchos ~]# systemctl status ktrusted
● ktrusted.service - KTrusted DIM
     Loaded: loaded (/usr/lib/systemd/system/ktrusted.service; enabled; preset: disabled)
     Active: active (exited) since Tue 2025-10-28 01:26:22 CST; 15s ago
   Main PID: 122553 (code=exited, status=0/SUCCESS)
        CPU: 569ms

Oct 28 01:26:22 keyarchos systemd[1]: Starting ktrusted.service - KTrusted DIM...
```

图 3-4-2 查看动态可信验证服务启动状态

查看动态可信验证审计日志

通过 KOS audit 模块记录了动态可信验证审计日志，日志内容为：

表 3-4-3 动态可信验证审计日志格式

事件类型	事件时间	事件日志内容
Type	Msg（unix time）	• Name(对象名称),值包括 Interrupt、NetProtoOpsUnix、NetProtoOpsInet6、NetFamilyUnix、NetFamilyInet6、KernelExTable、KernelRoData、SystemCallTable、ModuleList、ModuleKobjList、KernelSText、ModuleList 等； • Op（可信验证）,值为 Validate； • State（可信验证结果）,值为 Trusted、UnTrusted 等；

		• cause（可信验证类型）,值为 p_check_kintegrity； • res（可信验证执行结果），值为 success/failed。

可通过如下命令查看审计日志：

1. tail -f /var/log/audit/audit.log

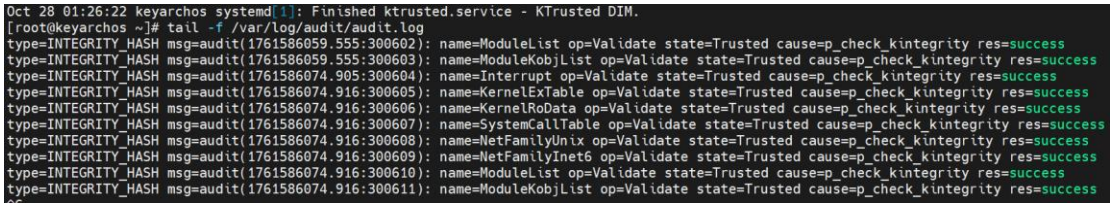

图 3-4-3 查看动态可信验证审计日志

查看动态可信度量日志

可信增强软件动态可信度量日志内容格式参考 TCG 规范，日志内容为：

表 3-4-3 动态可信度量日志格式

本次事件扩展的 PCR	**扩展后 PCR 值**	**度量算法**	**度量结果**	**度量对象**
PCR Index	PCR value	Hash Alg	Hash Vaule	Object Name

可通过如下命令查看度量日志：

1. cat /sys/kernel/security/ktrusted/lkrg/lkrg_ascii_runtime_measurements

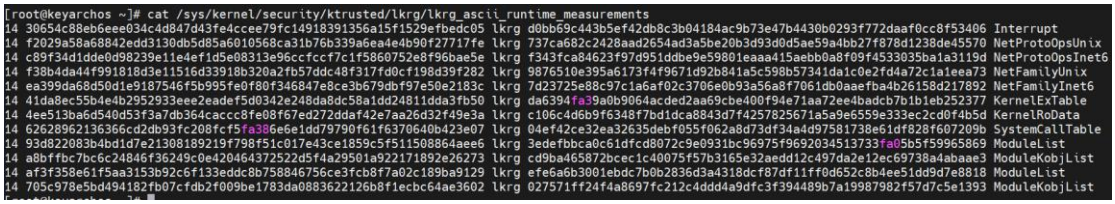

图 3-4-4 查看动态可信验证度量日志

随机周期触发可信验证

随机周期是为了防止固定周期场景下面临的 TOC-TOU 攻击风险。可通过如下命令开启随机周期功能，开启随机周期后固定周期停止。

1. sysctl -w lkrg.rkinterval=1

可通过如下命令关闭随机周期功能,关闭随机周期固定周期恢复，周期为设置随机周期前固定周期的周期值。

1. sysctl -w lkrg.rkinterval=0

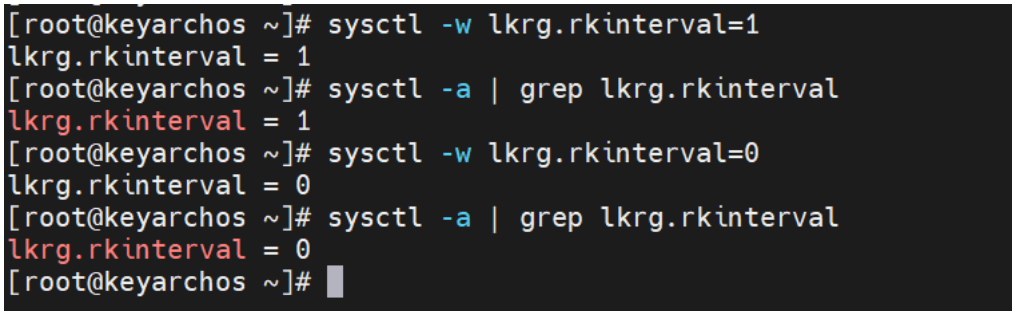

图 3-4-5 开启及关闭动态可信验证随机周期触发机制

3.4.3 KTrusted 可信增强服务的 RISC-V 应用场景

KTrusted 在 RISC-V 系统下可为多类上层应用提供**可信增强服务**

应用场景	可信增强能力

虚拟化管理器（KVM / RISC-V HV）	对 VMM 与 Guest Image 执行度量与签名验证，确保虚机加载完整性。
容器与微服务（KOS Container Runtime）	对 Container Image 和 Runtime 环境执行动态度量与签名校验，防止镜像篡改。
AI 推理与训练框架（KOS AI Runtime）	保护 AI 模型与权重文件完整性，防止模型注入与参数污染。
边缘计算节点	为 IoT / Edge 节点提供低功耗可信验证，支撑 RISC-V 轻量安全场景。
可信存储与加密服务	对加密驱动与 KMS 接口执行度量，保证密钥服务安全。
可信更新机制	对系统与固件更新包执行签名验证与完整性度量，实现可验证更新。

KTrusted 通过轻量化、可扩展的动态可信框架，为 RISC-V 平台提供了持续的运行时可信保障。它既是 KOS 可信体系的重要延伸，也是国产 RISC-V 生态实现“可验证安全”的关键技术支撑。

未来，KTrusted 将与 RISC-V 社区合作，推动可信度量接口标准化，实现开源生态下的动态可信共建。

参考资料

[1] 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(08):1127-1147.

[2] 沈昌祥,公备.基于国产密码体系的可信计算体系框架[J].密码学报, 2015,2(05):381-389.DOI:10.13868/j.

[3] 石文昌编著. 信息系统安全概论. 电子工业出版社，2014.68-71.

[4] 张焕国、赵波等著. 可信计算. 武汉大学出版社.

[5] 威尔·亚瑟（Will Arthur） / 大卫·查林纳（David Challener）等《TPM 2.0 原理及应用指南》. 机械工业出版社. 2017-10-1.

[6] 王勇,张雨菡,洪智等.基于 TPM 2.0 的内核完整性度量框架[J].计算机工程,2018,44(03):166-170+177.

[7] https://www.intel.com/content/www/us/en/developer/articles/code-sample/protecting-secret-data-and-keys-using-intel-platform-trust-technology.html.

[8] 沈昌祥院士《可信计算筑牢网络强国底座》演讲稿.

[9] 浪潮信息张东于 2022 年 KeyarchOS 发布会.

OpenAnolis

龙蜥社区