上海 瓶钵信息科技®有限公司 (https://www.trustkernel.com)

上海交通大学IPADS实验室 (http://ipads.se.sjtu.edu.cn)

移动平台TEE的设计与安全评估

TrustKernel && 上海交通大学　利文浩

关于TrustKernel

上海 瓶钵信息科技®有限公司，依托于上海交通大学软件学院、并行与分布式系统研究所成立，成员均为来自上海交通大学的博士和硕士

公司重点关注系统安全，专注于业界前沿，致力于将研究所的研究成果转化为产品

TEE及其应用简介

TEE 与 REE

TEE三层架构

TA (Trusted Application)层

可信钱包、TUI

TEE层

安全操作系统

对上层TA提供的库

硬件层

CPU状态隔离

内存隔离、外设隔离

ARM TrustZone技术

ARMv6版本开始的安全硬件特性

包括ARM11及Cortex A系列

Contex-A8, Cortex-A9, Cortex-A15等

同时运行一个安全的OS和一个普通的OS

两个系统之间互相隔离运行

安全的操作系统具有更多的权限

TrustZone是一个全系统级别的安全架构

处理器、内存和外设的安全隔离

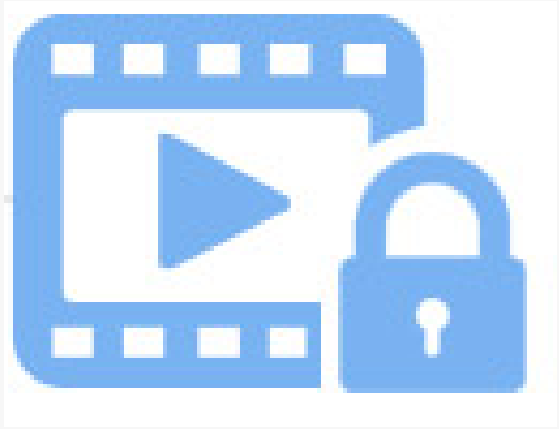
TEE在工业界的实际应用：现有技术方案

Samsung TIMA

Apple: iOS Enclave

Qualcomm: Zeroth

安全支付与解锁



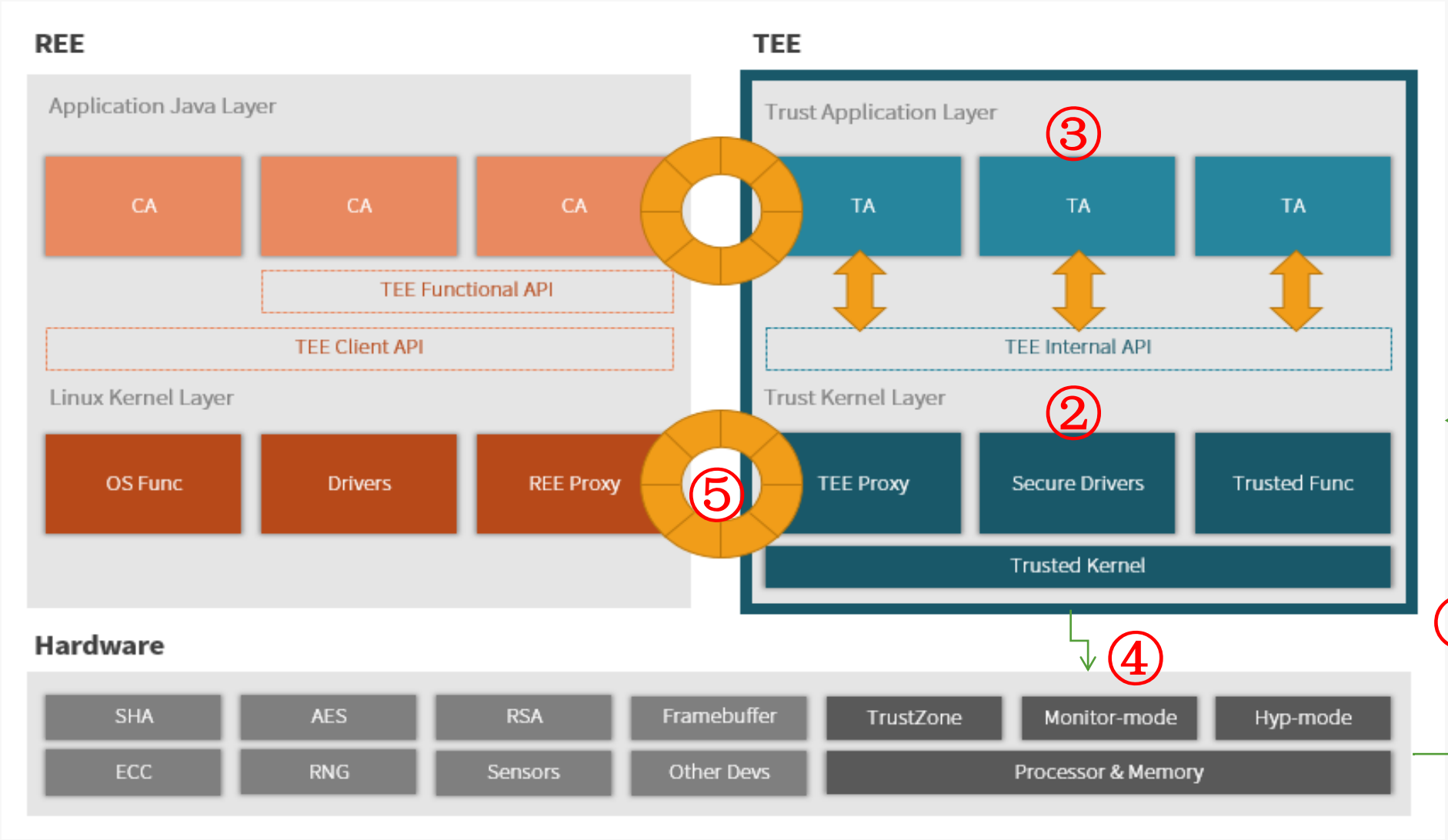
DRM

……

现有TEE是否真的安全？

TEE功能日益复杂与多样化

TEE攻击面分析



①安全启动

②TEE内核自身

③TA安全性

④硬件操作与隔离

⑤TEE与REE交互部分

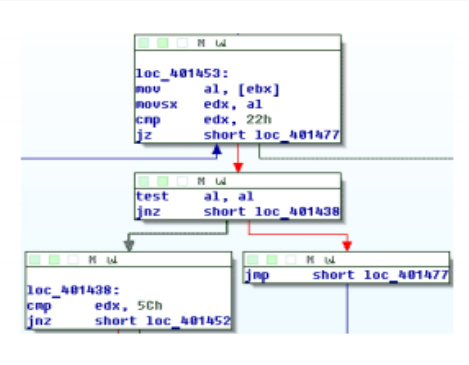
攻击者的Toolkit

REE侧工具

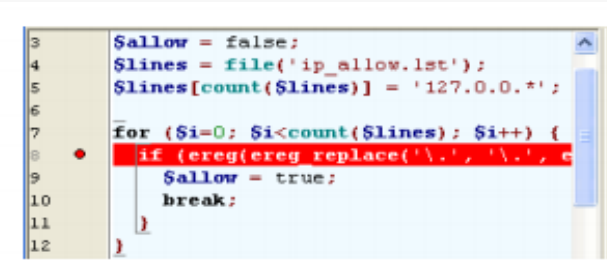
TEE侧工具

Root工具（如KingRoot）

Decompiler（如JEB）

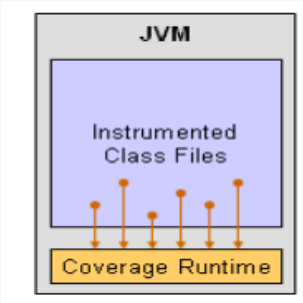


Disassembler（如IDA）



Debugger工具（如GDB）

Introspection工具（如Androguard）



Instrumenting（如ADBI）

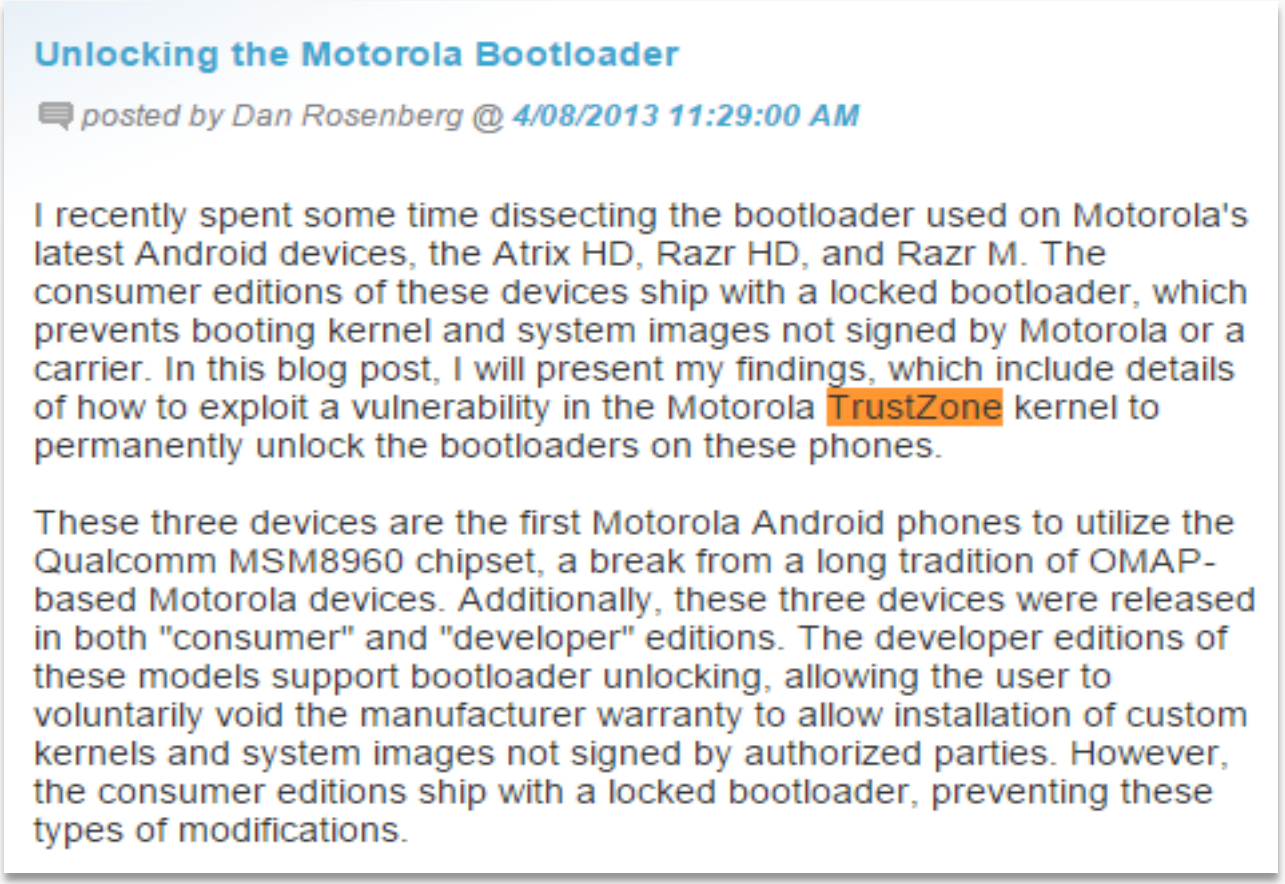
TEE SDK

旁路分析、嗅探工具

实际攻击：针对TEE安全启动

2013年Moto、三星部分手机的TrustZone内核被破解

可解锁Bootloader，加载任意系统



Unlocking the Motorola Bootloader

posted by Dan Rosenberg @ 4/08/2013 11:29:00 AM

I recently spent some time dissecting the bootloader used on Motorola's latest Android devices, the Atrix HD, Razr HD, and Razr M. The consumer editions of these devices ship with a locked bootloader, which prevents booting kernel and system images not signed by Motorola or a carrier. In this blog post, I will present my findings, which include details of how to exploit a vulnerability in the Motorola TrustZone kernel to permanently unlock the bootloaders on these phones.

These three devices are the first Motorola Android phones to utilize the Qualcomm MSM8960 chipset, a break from a long tradition of OMAP-based Motorola devices. Additionally, these three devices were released in both "consumer" and "developer" editions. The developer editions of these models support bootloader unlocking, allowing the user to voluntarily void the manufacturer warranty to allow installation of custom kernels and system images not signed by authorized parties. However, the consumer editions ship with a locked bootloader, preventing these types of modifications.

来源: http://blog.azimuthsecurity.com/2013/04/unlocking-motorola-bootloader.html

实际攻击：针对TEE自身安全

高通 QSEE内核整型溢出 CVE-2014-4322

drivers/misc/qseecom.c文件存在漏洞

没有判断offset和length的正确范围

可能导致攻击代码提权或DoS攻击

华为RTOSck内核任意代码执行 CVE-2015-4422

内核无ASLR、NX

实际攻击：针对TA安全

HTC指纹数据直接明文存放在Android侧

HTC caught storing fingerprint data in unencrypted plain text

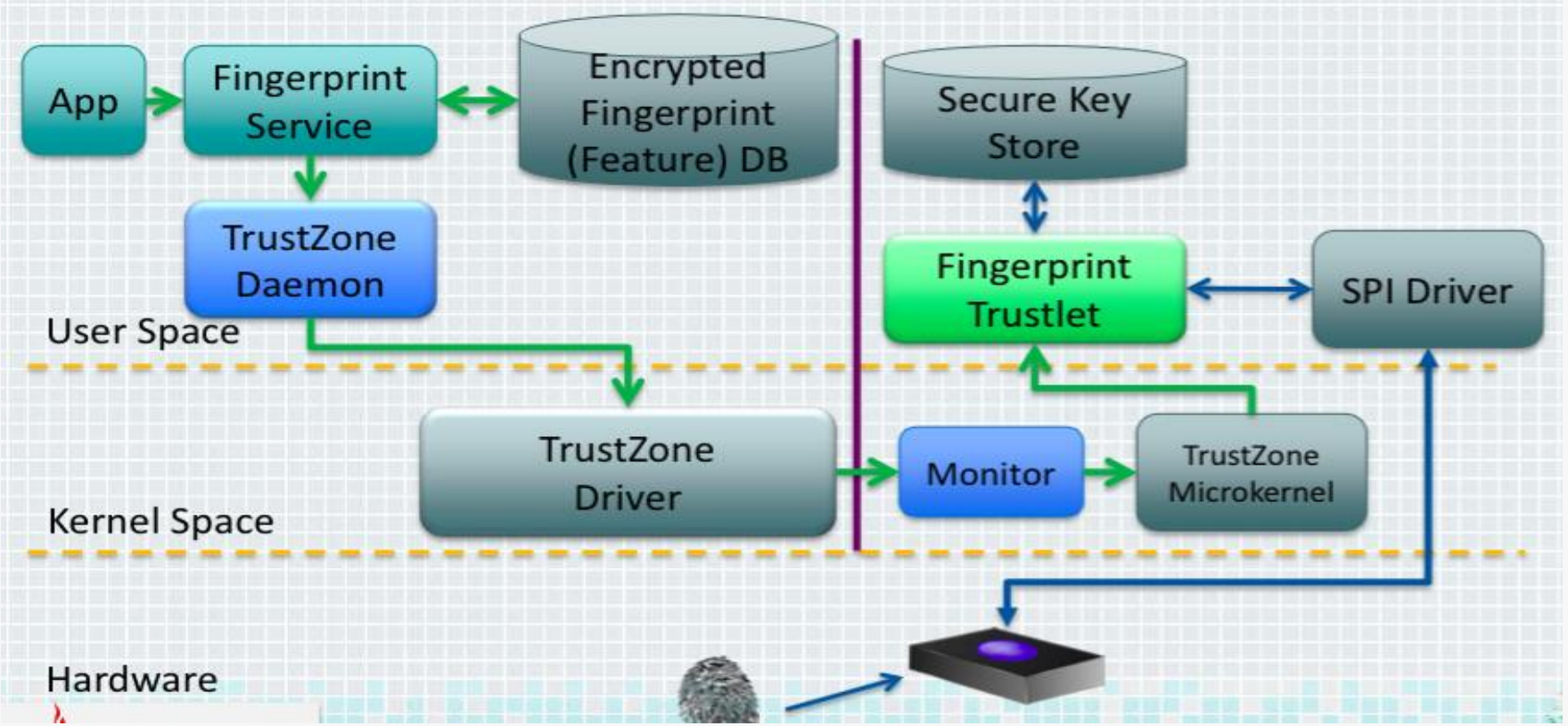
By Joel Hruska on August 10, 2015 at 4:24 pm | 40 Comments

Share This article

For the past few years, both Apple and the various Android manufacturers have been pushing the idea of fingerprint readers, typically on the dubious grounds that biometric security is a better choice compared to a good passcode. New research from the security firm FireEye seems to blow that claim wide open, however. According to FireEye, multiple Android manufacturers protect your fingerprint so poorly, it can be read by plugging the phone into a computer and knowing which folder to access.

实际攻击：针对硬件操作安全

Blackhat’ 15: 三星S5可在Android侧读取指纹设备数据

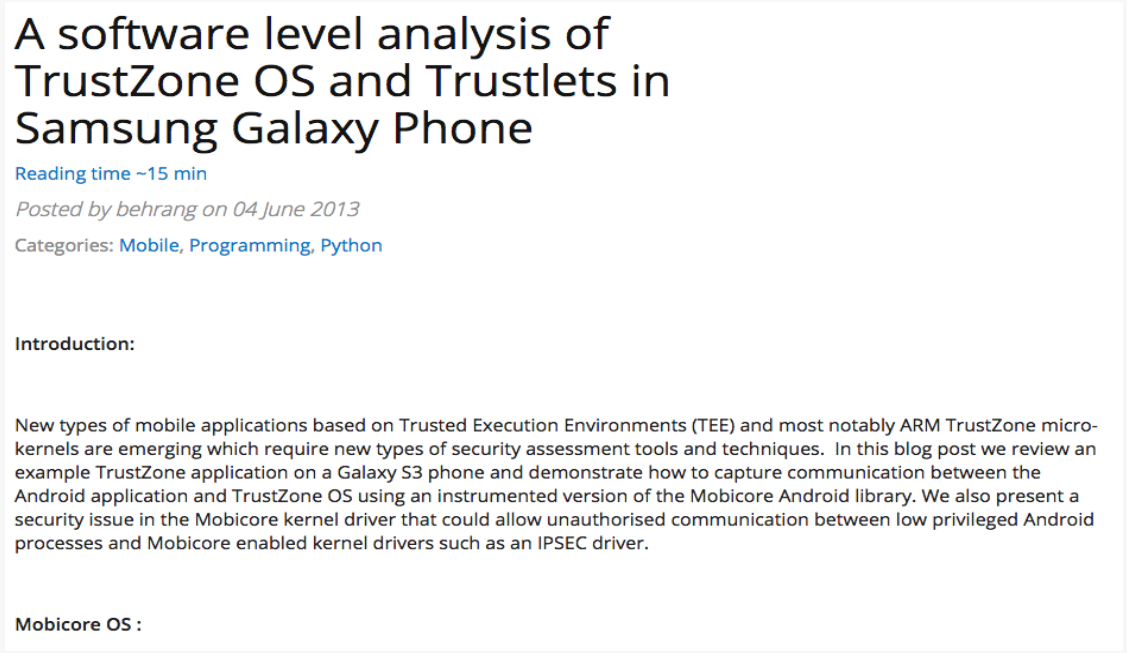


实际攻击：针对TEE与REE交互部分

TEE Driver提权漏洞 CVE-2015-4421

代码边界检查不正确

Samsung手机TEE内核MITM分析



A software level analysis of TrustZone OS and Trustlets in Samsung Galaxy Phone

Reading time ~15 min

Posted by behrang on 04 June 2013

Categories: Mobile, Programming, Python

Introduction:

New types of mobile applications based on Trusted Execution Environments (TEE) and most notably ARM TrustZone micro-kernels are emerging which require new types of security assessment tools and techniques. In this blog post we review an example TrustZone application on a Galaxy S3 phone and demonstrate how to capture communication between the Android application and TrustZone OS using an instrumented version of the Mobicore Android library. We also present a security issue in the Mobicore kernel driver that could allow unauthorised communication between low privileged Android processes and Mobicore enabled kernel drivers such as an IPSEC driver.

Mobicore OS :

小节：只有硬件安全隔离还远远不够

TEE与上层安全应用自身的安全性更需要重视

需要从整体设计与安全评估进行TEE安全增强

现状与趋势	问题与挑战
1、芯片厂商的TrustZone硬件隔离实现机制各异	如何保证不同TEE方案的安全性？
2、大多TEE与手机厂商更关注功能性	如何确保TEE自身安全性？
3、TA不断增加，带来新的安全隐患	如何更有效地将TEE与TA隔离开？
4、TEE无法应对物理攻击	如何提高TEE防御的层次？

T6的安全设计

T6: TrustZone TEE 与 Secure OS

高安全：动静态保护

- SecureBoot完整信任链
- TA签名验证与保护
- TA中不同域与库间互相隔离
- 安全敏感数据明文仅在SoC内部

小尺寸：轻量而易用

- 核心安全代码小于1W行
- 开发库支持完善
- 兼容GlobalPlatform Specs

T6安全设计关键技术

最小化可信计算基

安全复用REE驱动设备代码

非安全敏感功能复用REE系统完成

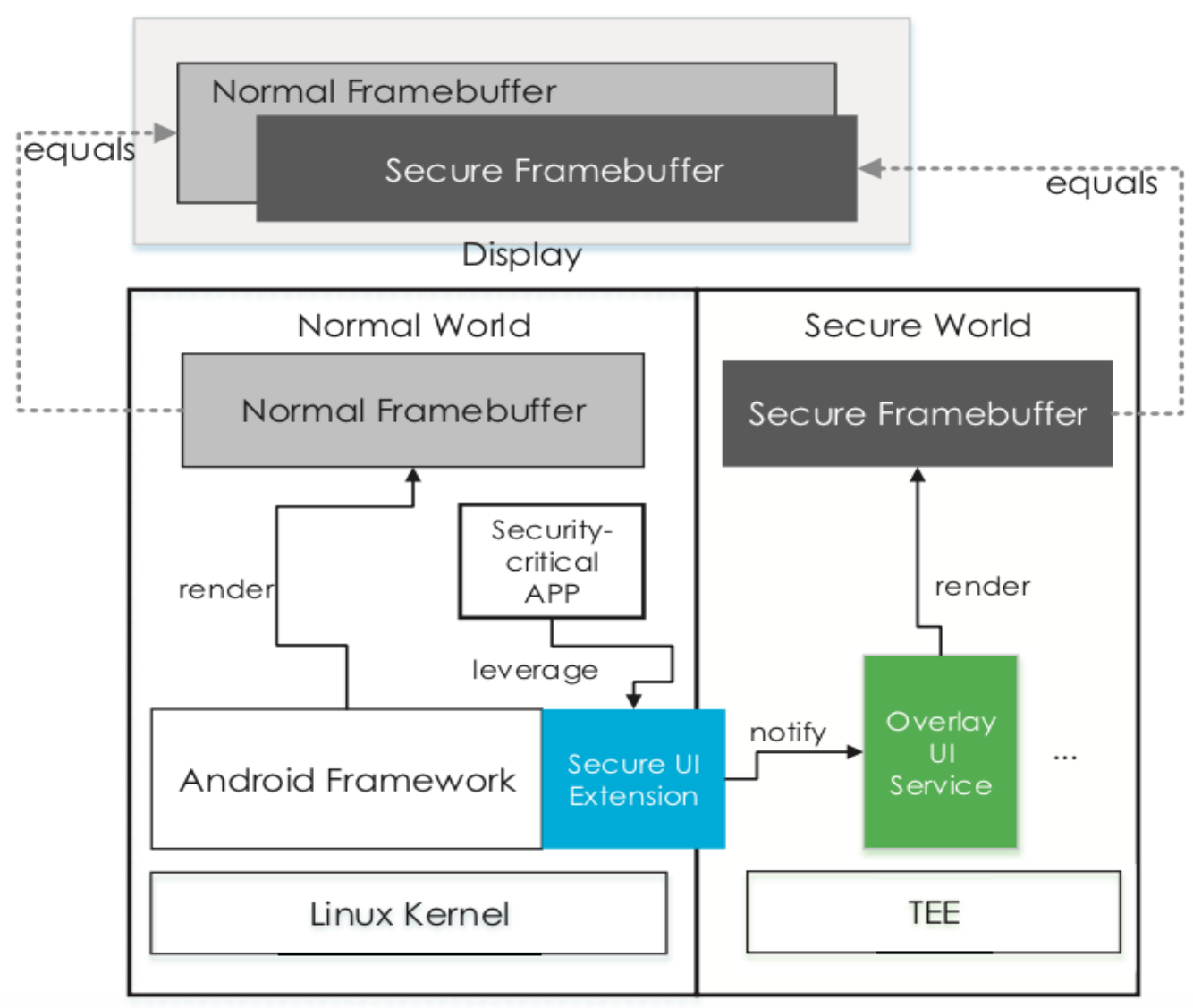
非安全关键逻辑由用户态管理

减少攻击面

沿用现代操作系统安全隔离机制：地址空间隔离、ASLR、NX等

T6-m：TEE完全运行在SoC内部

不可信驱动代码复用案例：TUI



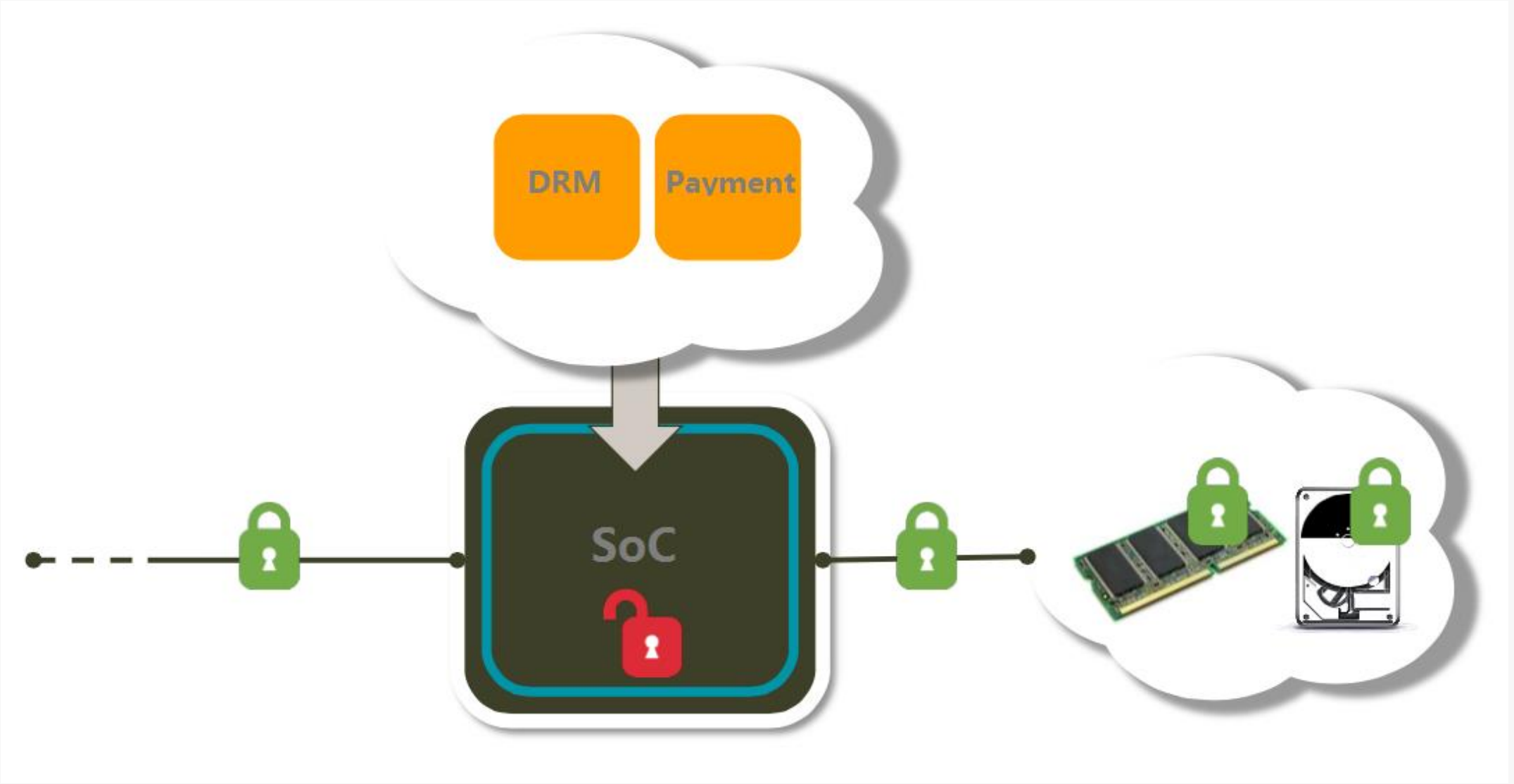
初始化

REE驱动负责初始化

安全显示

TEE需要TUI时将显示与输入设备设为安全设备，并进行安全显示

防御物理攻击: 敏感数据只存在SoC内部



T6-m：敏感数据只存在SoC内部

威胁模型

仅SoC可信：防御物理攻击

On-chip TEE

整个OS运行在SoC内部，外部内存中只保存密文数据

支持后向兼容，保持对现有TA的透明性

实现原理

Soc内部存储：Internal RAM、Locked L2 Cache

Soc Paging: SoC存储不够时加密换出到DRAM

T6-m：敏感数据只存在SoC内部

威胁模型

仅SoC可信：防御物理攻击

On-chip TEE

整个OS运行在SoC内部，外部内存中只保存密文数据

支持后向兼容，保持对现有TA的透明性

实现原理

Soc内部存储：Internal RAM、Locked L2 Cache

Soc Paging: SoC存储不够时加密换出到DRAM

突破TrustZone安全防御限制，进一步提升TEE安全

TEE安全评估：测试工具覆盖面

隔离性

TEE与REE之间的内存、设备隔离

TEE与TA之间、不同TA之间、TA与SE之间的隔离

交互性

TEE与REE之间的交互、TEE与TA之间的交互

不同TA之间的交互、TEE与SE之间的交互

完整性

安全启动、TEE完整性、TA完整性

安全存储

TA、TEE数据和秘钥数据的安全存储

功能性

随机数生成、加密操作的安全性

References:

《GlobalPlatform-TEE Protection Profile.pdf》

《TrustKernel-TEE安全测试规范要点.pdf》

《TrustKernel-TEE安全测试方案.pdf》

一些结论

TEE是一种基于硬件的隔离

TrustZone、虚拟化、SGX等均可实现

基于TrustZone的TEE部署最为广泛

不是100%的TEE都安全

各类CVE开始出现

仅有硬件隔离还不够，软件安全性更重要

T6在增强TEE安全方面的关键技术

最小化可信计算基

On-chip TEE/OS

TEE需要全面的安全评估

硬件实现机制差异化的影响

TEE自身安全性

第三方TA安全性

谢谢

liwenhaosuper@trustkernel.org

https://www.trustkernel.com