# Building a secure IoT system: background and concept And method

# 构建安全的物联网系统：背景、概念和方法

讲者：何小庆/Allan He

2021年4月

allan@esbf.org

嵌入式系统联谊会 www.esbf.org


## 物联网安全的背景和概念



嵌入式系统联谊会 www.esbf.org

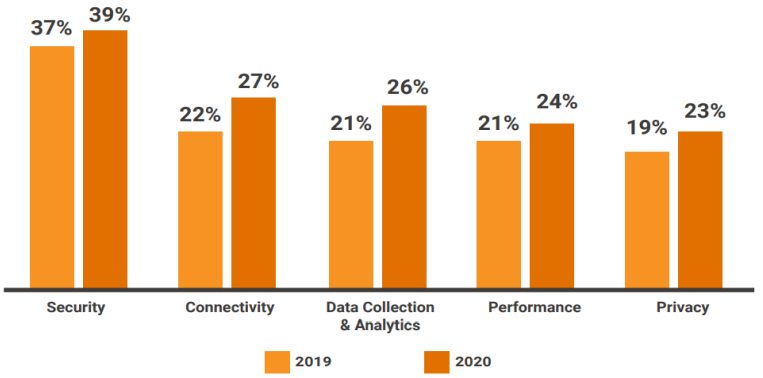
# 嵌入式系统的安全刻不容缓

- Barr Group 是一家咨询服务公司，帮助工程技术人开发嵌入式项目，最近它做了一个市场调查：**60%**的正在开发中的嵌入式系统都是连接到互联网上（物联网），它们之中有**25%**的系统是可能受到攻击的。**48%**受访工程师告诉Barr Group 他们没有加密他们的通讯协议


来源: 单片机与嵌入式系统应用 2019.6

嵌入式系统联谊会 www.esbf.org

# IoT开发人员最关心的问题




**安全从37% -39% 继续位居榜首，连接22%-27%，数据采集和分析21%-26% 性能 21%-24% ， 隐私 19%-23%**

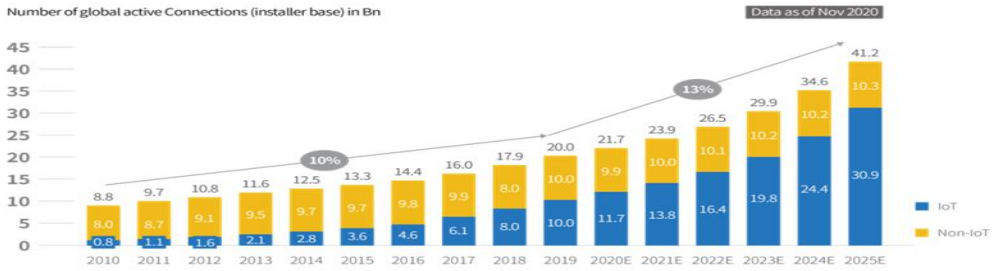
**-数据来自 Eclipse 基金会 2020 IoT Developer Survey**

4

嵌入式系统联谊会 www.esbf.org

# IoT产业发展提速，联网终端数量猛增

- 尽管新冠疫情还在持续，但IoT市场仍在不断增长。据IoT Analytics的统计，2020年，IoT联网设备（如：智能网联汽车、智能家居、工业联网终端）的数量将首次超过非IoT联网设备（如：智能手机、笔记本电脑和台式机）。
- 2020年底，全球217亿个活动联网设备中，IoT设备达到117亿（占比约54%）。到2025年，预计将有超过300亿的IoT连接，全球平均每人将近4台IoT设备。







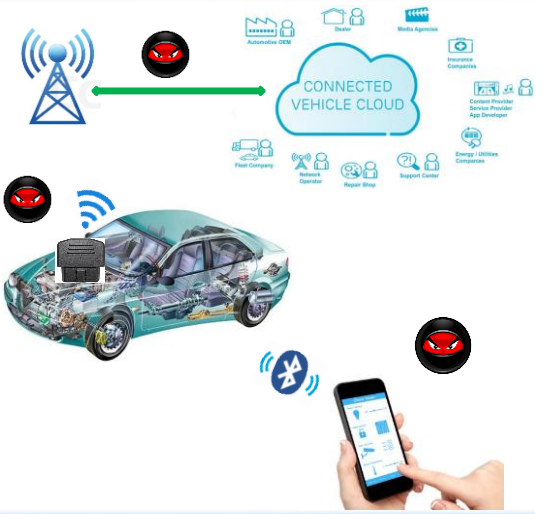

# 安全威胁加剧，全球IoT安全支出飙升

- 随着各类功能丰富的智能设备融入大众的生活，人与设备的联系更加紧密。IoT设备承载了越来越多的生产生活数据和个人隐私信息，安全问题逐渐得到重视。
- 由于IoT设备海量异构的特性，其安全攻击带来的后果也更为多样和严重。例如，智能摄像头被攻击可能带来家庭或工作隐私的泄露，智能手表被攻击可能带来行动轨迹的泄露，智能网联汽车被入侵更是可能直接对人身安全造成严重威胁。


来自:2020年IoT终端安全白皮书-梆梆安全

# 物联网安全隐患错综复杂

- 更多日常活动因为攻击而可能终断。比如大量**可穿戴的医疗健康设备**，都通过智能手机接入互联网，攻击导致设备故障或将危机人们的健康甚至生命
- 互联网和**大数据**通过传感器收集到了大量物的信息，其内容更加广泛，一旦**信息泄漏**危害更大，比如我们驾车的**汽车的位置，个人信息和疾病信息**
- **电网**、**交通运输**、**核电站**和环境监测等**关键系统**若遇到黑客的攻击，将是毁灭性的危害








# 什么是安全?

- 安全的英文词汇是 **Safety 和 Security**， 但两者含义非常不同：
  - Security ——IEC 17799 标准将 Security 定义为保护外来伤害的能力，适用于脆弱和价值的资产，比如人、房产、组织和国家。
  - Safety ——IEC 61508 标准将Safety 定义为将能够造成人员伤害及财务损失以及严重危害环境危险降低到可以接受的范围。
  - ISO/IEC 17799-1《信息安全管理实施细则》
  - IEC61508《电气/电子/可编程电子安全系统的功能安全》

# 关于Safety的几个概念

- 安全关键系统（**Critical systems**）
  - 这样的系统一旦失效，会造成人员和重大财产损失，也可能对环境造成严重的破坏。研究这种系统的目的是通过降低系统发生故障的概率。完美的想法是我们的设计不会失效，系统会非常可靠。真实世界是，即使我们尽全部努力，100%可靠性是不可能。系统失效总会发生，我们能做的是控制风险在一个可接受的范围。
- 安全完整性等级（**Safety integrity level -SIL**）
  - 在控制系统安全和可靠性评估中，有安全完整性等级指标（如IEC 61508-下图）作为参考。通过这些指标，比较计算结果，考虑系统如何设计可以最大限度提高安全和可靠性。

失效概率

可接受的失效率

| Probability of failure | Frequent | Probable | Occasional | Remote | Improbable | Incredible |
|---|---|---|---|---|---|---|
| Acceptable failure rates | ==>$10^{-3}$ | ===>$10^{-4}$ | ===>$10^{-5}$ | ===>$10^{-6}$ | ===> $10^{-7}$ => $10^{-8}$ =>$10^{-9}$ => | |
| | | | | SIL 1 | SIL 2 | SIL 3 / SIL 4 |





# 安全完整性等级简介

- 安全完整性级别（SIL）的概念是为了“提供一个安全功能可以达到的目标”。
- $10^{-9}$失效率意味着10亿小时（约11415年）的运行中会失效一次。相邻等级大致相差一个数量级；SIL 1安全功能将使风险降低10倍，SIL 2降低100倍，SIL 3降低1000倍，SIL 4降低10000倍。
- 系统设计者完成了风险分析后就可以定义不同的SIL了。作为一个软件设计者，你必须达到系统设计者定义的最低安全级别要求。你可以选择的RTOS也自然会受到限制（你甚至可能无法使用RTOS），对嵌入式软件和工具有要求。

| 失效概率 | 可接受的失效率 |
|---|---|
| 非常不可能 | < $10^{-9}$ |
| 非常罕见 | < $10^{-7}$ |
| 罕见 | < $10^{-5}$ |
| 不太可能 | < $10^{-3}$ |
| 可能 | ≥ $10^{-3}$ |

非常不可能：理想状况下失效应该完全不会发生
非常罕见：失效基本不可能发生
罕见：失效在系统的生命周期中不太可能发生
不太可能：在系统的生命周期中会有一次失效
可能：在系统的生命周期中会有多次失效

# 物联网系统安全的目标

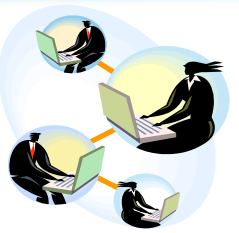
- **物联网为什么需要安全?**
  - 物联网系统攻击的风险是真实存在。对于黑客而言，通过成功的尝试获得经济收益，特别是如果攻击可以像物联网那样大规模地传播，很有吸引力!
- **物联网安全重点保护是?**
  - 物联网安全性旨在**保护代码、数据和系统设备功能**
    - **代码保护**是保证固件知识产权及其代码的完整性
    - **数据保护**是保证用户数据的机密性并避免身份盗用
      - 物联网资产保护，比如健康数据和位置信息，用户账号和密码，交易记录和密钥，以及设备和用户身份等
    - **系统功能安全**应受到重视，以避免设备故障及服务故障





# 物联网安全的基础- 密码学概念

- 加密技术（encryption）
  - **加密算法**(algorithm)
    - 对称加密（Symmetric encryption）DES 算法
    - 非对称加密（Asymmetric encryption）RSA 算法
  - **密钥管理**
    - 加密、解密和破解，从生成到销毁的全过程
- **数据完整性**和身份认证技术(Data Integrity)
  - 消息认证（Message Digests) 验证消息的完整性
  - 身份认证（Authenticate）证实实体身份
  - 数字签名（Digital Signature) 电子密码签名替代书写签名
  - 数字证书 （CA) 是权威机构发行的一种电子印证



# 物联网安全的基础- 网络安全协议概念

- 协议是在对等实体（两方或多方）之间为完成某项任务所执行的一系列确定的步骤，是协议实体必须共同遵循的**一套规则**。安全协议是使用**密码学**实现特定安全目标，在网络中提供各种**安全服务**的协议
- 虚拟专用网协议**IPSec**和传输层安全**TLS**协议，分别在**网络层和传输层**上实现对通信的保护。IPSec使用独立的密钥交换协议实现认证和安全参数协商。TLS使用握手协议和记录协议完成实体间认证和通信保护

| | |
|---|---|
| 应用层 | HTTPS, SSH, PGP, Kerberos,SET |
| 传输层 | SSL, TLS, SOCK5 |
| 网络层 | IPSec |
| 数据链路层 | PPP-PAP/CHAP,WEP |
| 物理层 | 物理层安全 |





arm TRUSTZONE

0x5 HEX-Five

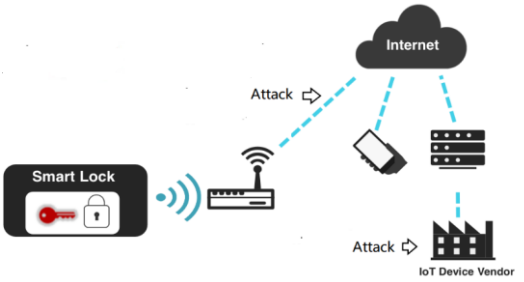
psacertified™

RISC-V Architecture

## 物联网安全的实现方法浅析

# 安全的物联网系统计算架构

- 安全物联网计算架构是为了应对多种形式的攻击，这样的系统架构必须在系统中实现多种类型的安全机制，应在**计算硬件和软件**两个方面考虑系统安全性
- 完整的系统安全实现涉及设备**保护机制**以一系列稳健性的**安全应用**
  - 安全引导确保引导时的应用程序完整性和真实性
  - **安全通信协议**
    例如用于Internet安全通信的TLS 协议
  - **安全固件更新**
    以安全方式下载新固件
    确保身份验证/机密性/完整性
  - **安全密钥存储**的功能，即存储和数据安全



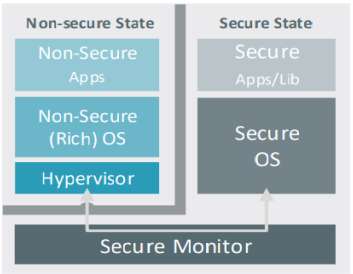





# 软件安全实现技术

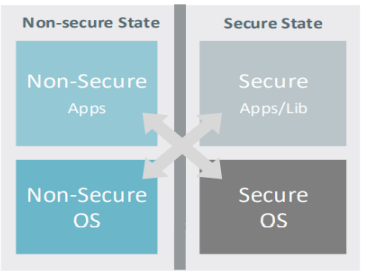
- 实现技术
  - 空间隔离
  - 时间隔离
  - 数据传输安全
- 实现方法
  - 硬件隔离- 多核/多处理器
  - 软件技术实现隔离- MPU/MMU

# TrustZone和 TrustZone-M




**TrustZone (Cortex-A)**

- 发布于2004年
- 执行状态由NS 位确定
- Monitor 模式 和 SMC 指令
- Memory 和 devices: TZASC 和TZPC
- Interrupts: GIC - IRQs vs FIQs




**TrustZone-M (Cortex-M)**

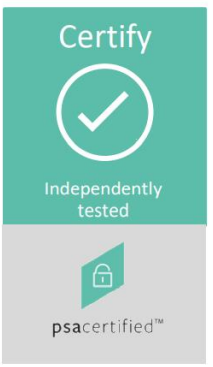
- 发布于2018年
- 执行状态是基于存储器映像方式
- 没有 Monitor 模式 和 SG 指令
- Memory 和devices: SAU 和IDAU
- Interrupts: NVIC – secure 和 non-secure IRQs

17

# PSA-Platform Security Architecture

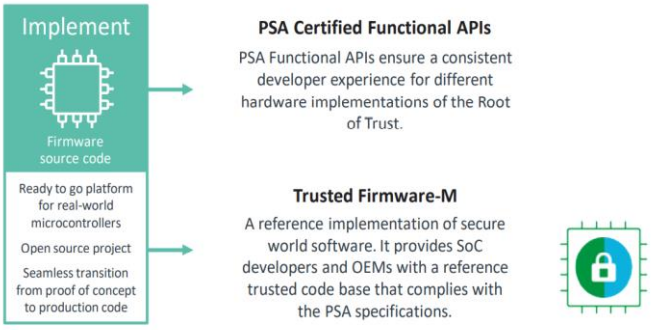
- 平台安全架构是完整的安全产品–公开发布和独立测试




18

# 加速安全之旅：TF-M和 PSA 认证




https://git.trustedfirmware.org/trusted-firmware-m.git/

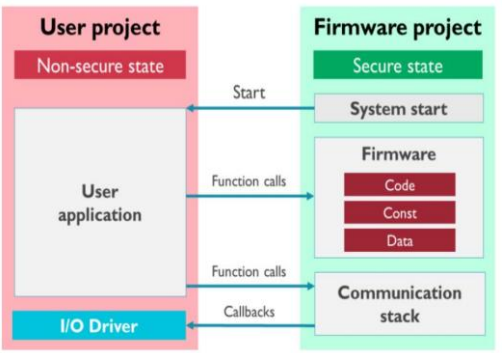

- The PSA website 包括认证设备数据库
- https://www.psacertified.org/certified-products/
- 芯片、模块、软件和组件



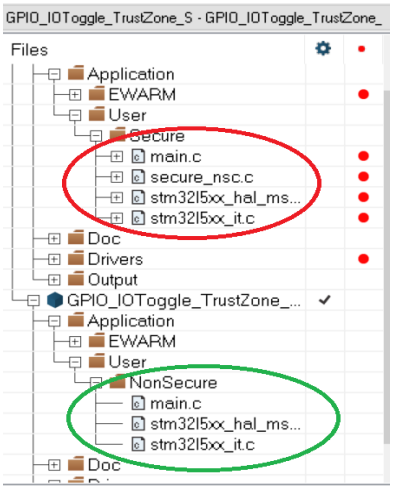


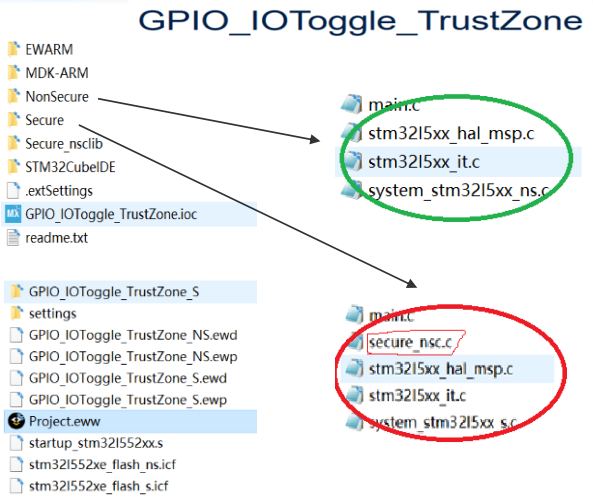
# 案例1: 基于STM32L5TZ 安全计算环境

- STM32L5 是基于M33内核 (TZ-M) 新型MCU
- STM32Cube_FW_L5_V1.3.0
  - *Project-NUCLEO-L552ZE-Q-example-GPIO-GPIO_IOToggle_TrustZone*
- 一个项目里包括两个工程
  - *GPIO_IOToggle_TrustZone_S*
  - *GPIO_IOToggle_TrustZone_NS*
- 启动后的运行流程
  - 配置Flash、RAM、外设、中断的安全属性
  - 切换到NS项目运行
  - NS项目调用S世界里的代码
  - 两个区域的systick和GPIO各自独立运行



# 案例1: 基于STM32L5TZ环境下软件开发

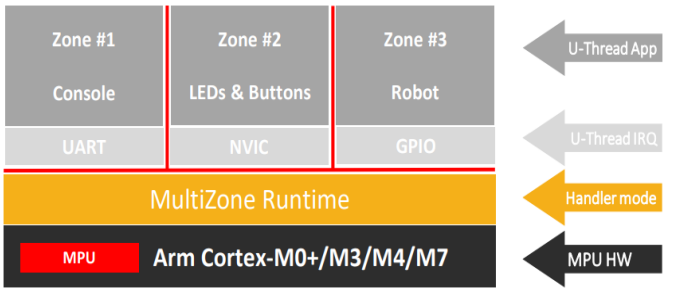


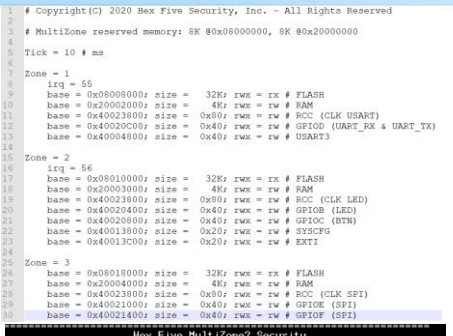



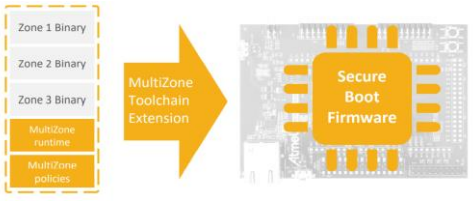


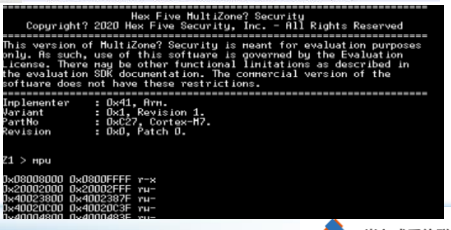



# 案例2：MultiZone 可信计算环境（TEE）



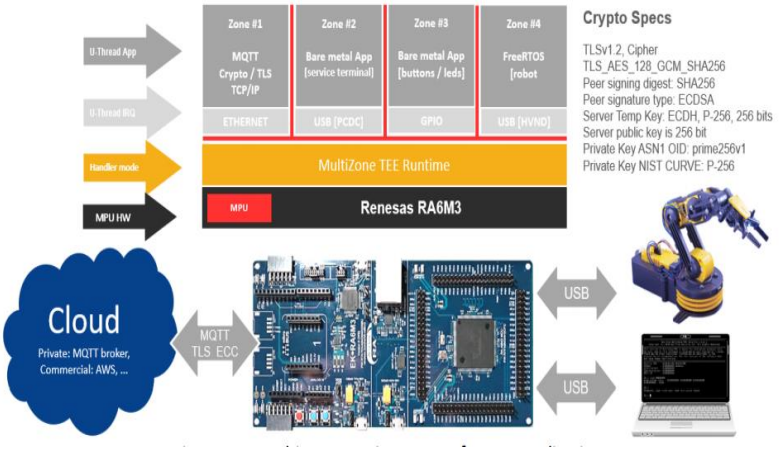






https://github.com/hex-five/multizone-sdk-arm

# 案例2：MultiZone IoT 应用开发




| Stack Component | Features | Size | License |
|---|---|---|---|
| IDE<br>• Renesas e² Studio 7.8.0<br>• Hex Five's reference projects | • MultiZone IoT Firmware: MQTT, TLS, TCP/IP, RTOS, TEE, robot, terminal<br>• MultiZone SDK: TEE, USB Robot, uart terminal, bare metal buttons & leds<br>• MultiZone Blinky: TEE, uart terminal, bare-metal buttons & leds<br>• MultiZone Minimal: TEE, 4 zones available for user applications | | Limited for use with Renesas products |
| BSP & Driven library<br>• Renesas FSP 1.1.0<br>• Hex Five's USB patch | • USB – optional, required for the robotic arm app<br>• UART – optional, required for the MultiZone terminal app<br>• Ethernet – optional, required for MQTT / TLS access to cloud services | 120KB<br>32KB | Limited for use with Renesas products |
| TCP/IP library<br>• UNIP 2.1.1<br>• Hex Five security patches | • IP, ICMP, UDP, TCP, ARP, DHCP, DNS, SNTP, MQTT<br>• Light weight single-threaded execution<br>• Fully integrated with SSL stack | 40KB<br>16KB | Modified BSD permissive commercial use ok |
| SSL library<br>• mbed TLS 2.23.0<br>• Hex Five secure config | • TLSv1.2, Cipher TLS_AES_128_GCM_SHA256<br>• ECC: prime256v1, Private Key NIST CURVE: P-256<br>• Mutual authentication, Cert expiration verification, TLS large fragment | 64KB<br>32KB | Apache 2.0 license permissive commercial use ok |
| Real Time OS (optional)<br>• FreeRTOS 10.3.0<br>• Hex Five integration with TEE | • Secure unprivileged execution of kernel, tasks, and interrupt handlers<br>• No memory shared with TCP/IP and SSL library code<br>• No memory shared with other applications running in separate zones | 32KB<br>16KB | MIT open source license permissive commercial use ok |
| Trusted Execution Environment<br>• MultiZone Security TEE 2.0<br>• RAM3 hw optimizations | • 4 separated Trusted Execution Environments (zones) enforced via MPU<br>• 8 memory-mapped resources per zone – i.e. ram, rom, I/o, uart, gpio, ...<br>• Secure inter-zone messaging – no shared memory, no buffers, no stack<br>• Protected user-mode interrupt handlers mapped to zones – up to 128 | 4KB<br>4KB | free for evaluation, commercial license priced per design – perpetual, no royalties |

https://www.renesas.com/eu/en/products/microcontrollers-microprocessors/ra/ra-partners/hex-five-multizone-iot-firmware.html



嵌入式系统联谊会 www.esbf.org

# 小结

- 基于MCU的物联网设备在节点和边缘，支撑了数据采集和控制的重要使命，因为**MCU系统资源薄弱防护能力低**，亦成为黑客工具的主要对象。
- 基于MCU 的IoT 芯片的安全机制越来越完善，安全功能也大幅度增加，技术上正在从外置向**内置，软硬件，端云结合**的方向发展。
- 物联网安全是一个**复杂的技术和系统工程**，需要产业链的通力合作，需要芯片、设备和云端合作提供一套完整方案，一头热，两头热不能解决问题。
- 物联网团队的**安全合规建设**也非常重要，即设计开发符合对应行业规范，软件编码要不存在安全漏洞，运营管理正常合规。

何小庆 allan@esbf.org

嵌入式系统联谊会 www.esbf.org