让系统无懈可击

浅谈 IoT 设备安全启动过程添加加密元件的重要性

Microchip 策略营销经理 Eustace Asanghanwa

对于任何一个嵌入式系统而言，安全启动都是至关重要的一个组成部分。这一过程可保证系统固件即所有嵌入式系统的大脑与系统制造商的设计初衷保持一致。安全启动确保了嵌入式系统的操作是安全并且可预测的。它的价值在那些因出现故障而可能导致灾难性后果的系统中是显而易见的。类似重要的系统包括家用炉灶和一体式烤箱灶的热控制器、汽车的发动机控制模块、交通灯控制器、植入式医疗设备中的治疗传输系统以及无人驾驶列车的控制器等等。这些系统可能会因为运行它们的固件发生故障而出现失灵或不可预测的操作。而导致此类故障的原因多种多样，可能是由电源浪涌造成内存故障这样的环境因素，也可能是黑客注入恶意代码的执行等等。无论在何种情况下，我们都可以在尝试运行系统之前先通过安全启动过程来检测固件的完整性。

安全启动一直都是在有需要的情况下才会被执行。虽然安全启动作为一个话题而言可能一贯很少被提及，但在很多关键系统中，一直有相应的法规和标准来强制执行安全启动以保障这些系统的安全运行。因此，大部分的电脑鼠标或手持计算器等一些重要性被认为较低的系统都直接跳过了严格的安全启动过程，因为它们出现故障所导致的后果一般都很轻；然而，什么样的构成可以称之为一个关键的嵌入式系统呢？这个定义正因为物联网（IoT）的出现和普及而悄然发生着变化。

IoT 将安全启动推向最前沿

关键系统和非关键系统之间的差异正日渐缩小。随着 IoT 的出现，可以说现在每个嵌入式系统都是一个关键系统。嵌入式系统不再像是一座孤岛，所有的性能和故障都只限于其中。虽然 IoT 将嵌入式系统连接在一起提供了很大的好处，但是这种联网的直接后果就是消除了遏制边界。现在，任何一个连入网络的嵌入式系统都可能是潜在的风险，而世界上任何一个人都可能成为潜在的受害者。

因注入故障到嵌入式系统的固件中而引发的潜在损害从来没有像现在这样大。像电源浪涌和通信错误等自然系统故障的发生大致上还是和以前一样，所以传统的安全启动过程还仍然有效。

但是，人为注入故障尤其是恶意类型故障的发生，无论是在种类上还是复杂程度上都在迅速增长。在过去，攻击者需要获取物理访问权限以便在每一个单独的系统中插入恶意故障。而现在，由于各个系统都是联网的，攻击者只需攻击其中的一个系统即可轻松获得访问其它许多远程系统的权限。这会导致大量设备被恶意控制，关键系统和存储在云端的数据遭到恶意访问，或是因为黑客炫技而发生臭名昭著的数据泄露事件。这也就是为什么我们有必要确保安全启动解决方案必须能够抵御攻击和故障注入的原因。

保障启动过程

安全启动包含两个基本的要素：检测固件完整性的能力和对检测过程完整性的保障。这些由来已久的要素很好理解，它们使用加密技术来实现各个目标，只在加密算法的复杂程度和保障检测过程完整性的安全硬件方法这两方面有所演变。

检测固件的完整性涉及使用加密技术来创建指纹，即一小段压缩的数字编码，可用来表示固件并且轻松地检测出变化。这种加密技术属于一类可生成指纹摘要、被称为散列函数的加密算法。常用的 256 位安全散列算法，或简称为 SHA256，可生成长度为 256 位的摘要。SHA256 是最新的散列算法，虽然它既不是最紧凑的也算不上最精细的，但是它却在安全性与嵌入式系统资源的有效使用二者之间取得了较好的平衡，这些嵌入式系统资源包括电源、代码空间和计算资源等等。

为了设置和实现安全启动，嵌入式系统制造商在工厂制造过程中即对最终的操作固件进行了散列，并在嵌入式系统中同时安装了固件和摘要。在实际操作过程中，嵌入式系统中的一段检测代码会对操作固件进行散列，并将所得的摘要与工厂安装的摘要进行比较。如果摘要完全匹配，即说明操作代码的完整性没有受损。

为了确保检测过程的完整性，最理想的做法是将检测代码放在诸如 ROM（只读存储器）等类型的非可变存储器中，使其不易受到电源浪涌等环境故障矢量以及内存修改疏忽等其它存储损坏情况的影响。为了响应快速变化的市场需求，我们通常会使用锁定版的非易失性存储器技术，比如闪存和 EEPROM，或是类似代替 ROM 的 TrustZone®技术等专用执行环境。

保护启动过程免遭攻击

上述的安全启动过程在没有恶意故障注入的情况下是充分够用的，然而，这种情况在我们生活

的真实世界中并不典型。攻击者只需要创建自己的固件以及相应的散列摘要并将二者都安装到系统中即可击败安全启动过程。这破坏了检测的完整性，因此我们需要有一个验证检测过程。

该验证检测过程需要使用诸如密钥这样的秘密信息，并生成固件的认证摘要或是简单的一个证书（图 1）。之所以这样设计，是因为对手如果想要阻挠检测系统正常工作就需要知道相同的秘密信息以生成一对一致的固件 - 签名。考虑到操作代码的验证过程也需要访问同样的秘密信息，嵌入式系统很可能会被攻击者拆解，试图找出秘密信息。而构建高级嵌入式系统所需分析工具与技术复杂性的大幅提升也为攻击者提供了意想不到的帮助，使他们能够直接或通过相关服务访问这些工具以达到其利用系统的目的。

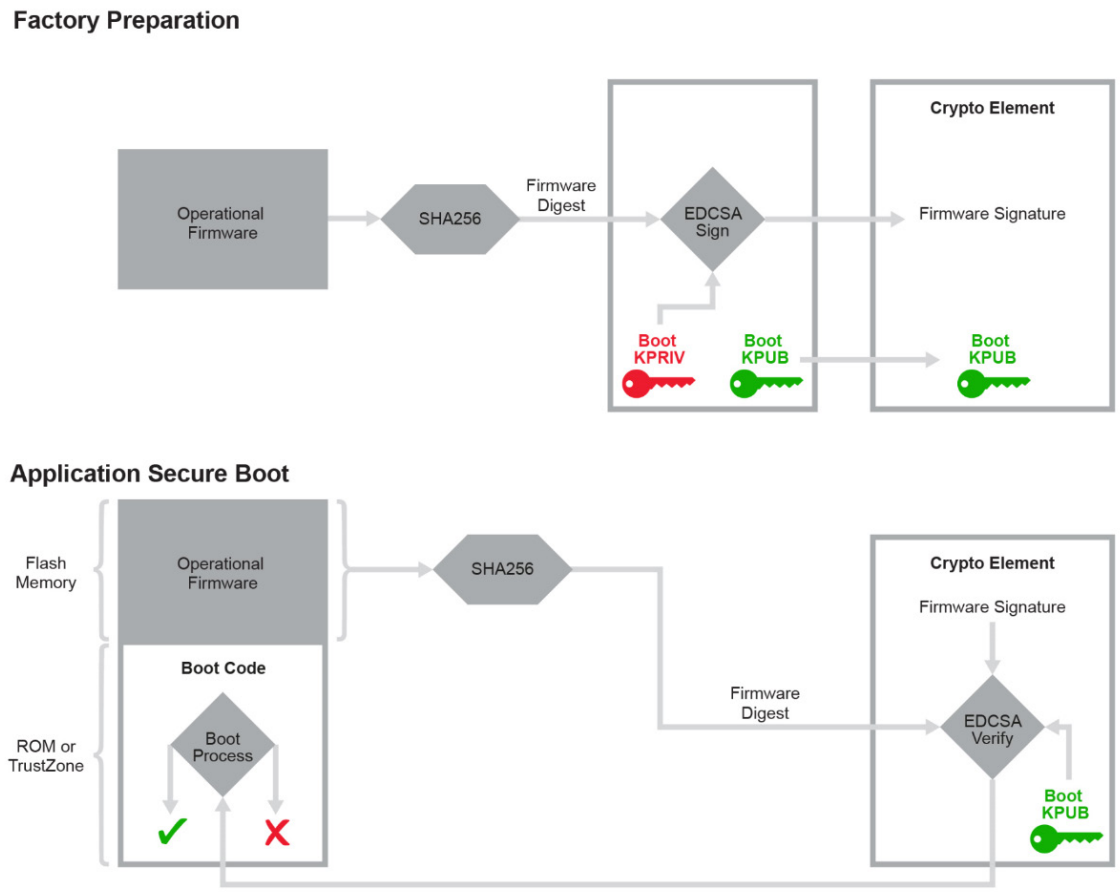

图 1：最佳的通用启动过程，包括签名操作固件的工厂准备和现场验证

不出现特殊情况的话，我们很容易想象出嵌入式系统开发人员和攻击者之间这个犹如猫和老鼠的游戏接下来会如何发展。要不是出现了一种特殊类型的叫做加密元件（CE）的集成电路，那么这个游戏还将继续下去。

加密元件适时阻止攻击

加密元件（CE）是专门设计用于抵御诸如尝试检索机密内容或篡改等攻击行为的集成电路。执行带有 CE 的安全启动操作提供了验证固件检测和验证过程中所需的完整性。CE 可以集成到控制器或独立的元器件中，为系统架构师提供所需的灵活性以迎合其部署需求。

对称与非对称密钥加密技术

虽然安全启动的基本要素即检测和过程保持不变，但是我们却可以选择对称密钥加密或非对称密钥加密技术来实现这两大要素，以控制整个启动验证过程。

对称密钥加密技术在安全启动过程的出厂设置和现场验证两个阶段均使用相同的密钥或相同密钥的导数。如图 2 和图 3 所示，基于 SHA256 算法的验证启动过程就是一个对称密钥启动过程的示例。基于对称密钥的启动过程具备速度方面的优势，但是在保证供应链中启动密钥的机密性方面却可能会遭遇困难。因此，封闭的生态系统是最受大家欢迎的，因为只有唯一的一个实体掌握其中的密钥。

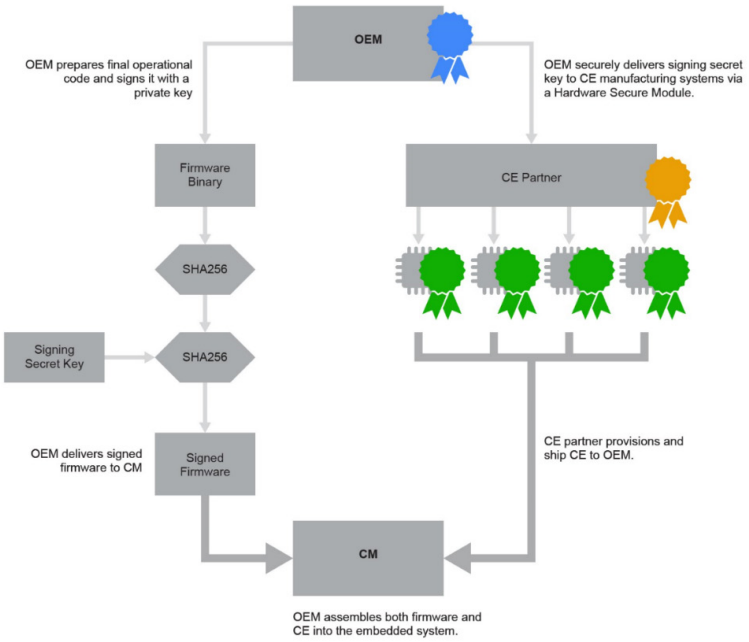

图 2：采用对称密钥保障安全启动的出厂设置

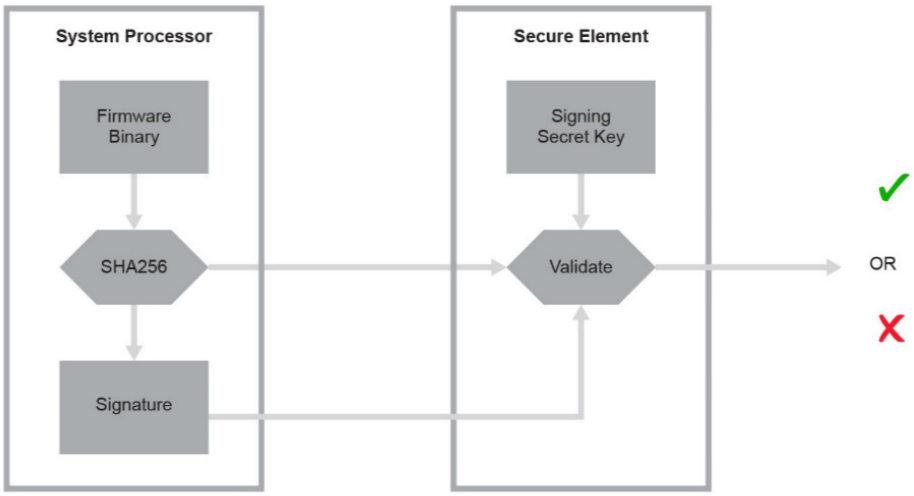

图 3：对称密钥流验证过程

非对称密钥加密技术（图 4）在安全启动过程的出厂设置和现场验证阶段则使用单独的密钥。这两个密钥之间的关系由诸如椭圆曲线加密技术（ECC）等加密算法来控制。ECC 被用在一种叫做椭圆曲线数字签名算法（ECDSA）的特殊协议中，该协议常应用于固件签名和验证。

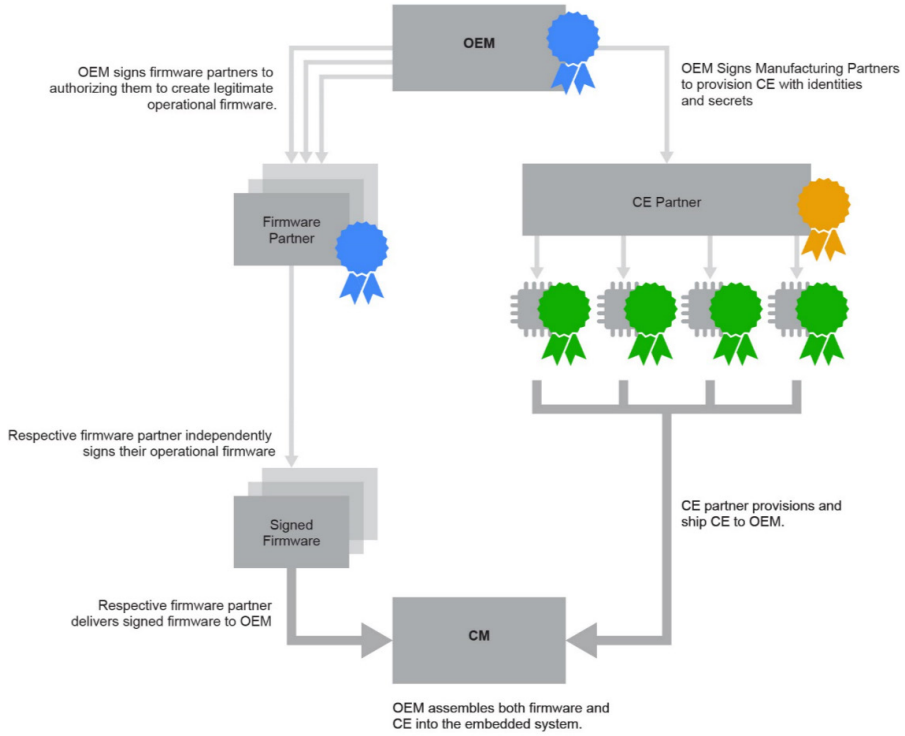

图 4：开放式合作伙伴生态系统中的安全启动设置。通常情况下，原始设备制造商（OEM）都拥有多家合作厂商来为其供应构成嵌入式系统的子系统或某个系统的替代资源。

使用非对称密钥过程，例如 ECDSA，也适用于安全散列算法 SHA。在实践中，SHA256 会检测操作固件以创建一个摘要，然后使用 ECDSA 协议对摘要进行签名以完成整个验证固件过程。这样生成的签名就是一个附有操作固件并可安装到嵌入式系统中的证书（图 5）。

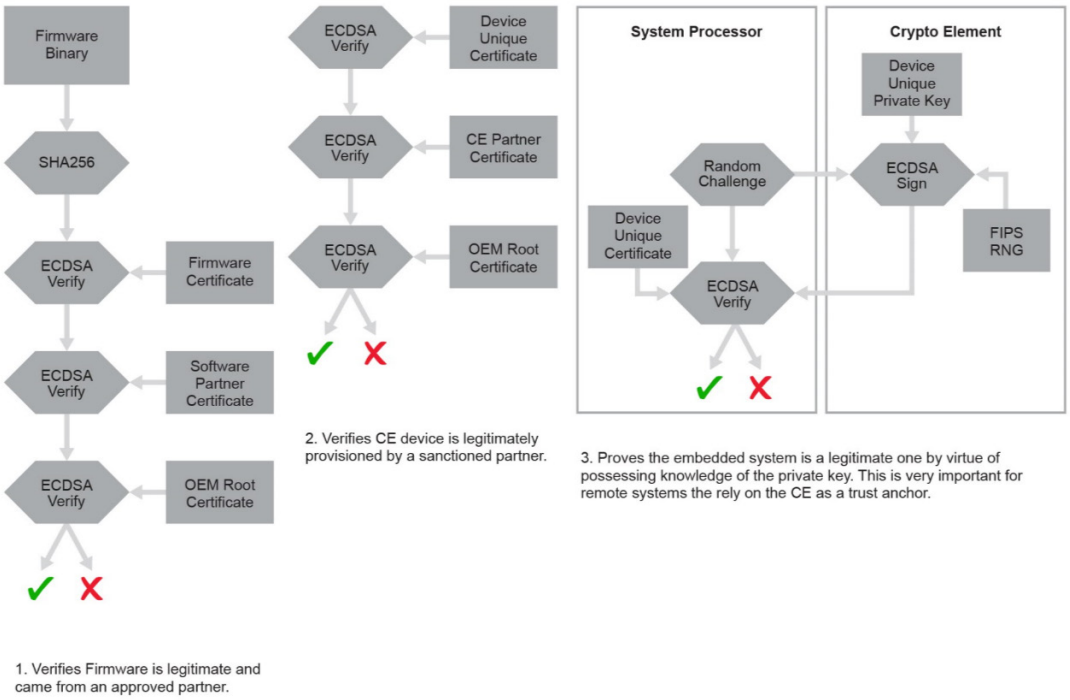

图 5：非对称密钥流的现场验证过程

非对称的密钥结构需要一个私钥和一个公钥，前者必须保密且只被用于出厂设置，而与其在数学上相对应的后者则只用于现场验证阶段。公钥可以被任何人查看而不影响启动过程的安全。因此，基于非对称密钥的安全启动过程更适用于由多个实体共同构成的开放式生态系统。

适合制造业的安全启动

如果一个安全启动过程需要很高的生产物流成本，那么它很快就会被市场抛弃。因此，一个有

效的安全启动过程应该是可以确定操作代码和检测过程的完整性且同时不会大幅增加生产流程的时间或成本。

虽然对于开放式和封闭式生态系统而言最佳的选择分别是不对称和对称密钥启动过程，但是使用加密元件却可以打破这个限制，使得任意一种启动过程都可以应用于任意一个生态系统并且

同时还能保持密钥的机密性。但是，非对称密钥方法可以提供更多的自由度，令设计人员能够在开放式合作伙伴生态系统中轻松打造一个从数学角度看非常严谨的信任链维护过程。

问责制助力安全启动落实

嵌入式系统的安全启动过程一直以来都是由管理产品安全的相关法规和标准所推动。当嵌入式系统是一个在物理上独立的系统、一个孤岛式的存在时，这种模式是非常有效的。然而随着 IoT 的出现，各个系统开始连入物联网，故障遏制边界的消除不仅大大鼓励了攻击者，也大幅提高了各界对安全启动的关注度。事物的远程可访问性意味着我们能更容易地访问嵌入式系统，而这样就令世界上任何一个地方的任何一个人都可能成为系统攻击的潜在受害者。虽然通过事后调查分析可能会揭示出哪个设备是罪魁祸首、哪家制造商应该被追责，但是损害已然造成。为了限制相关责任，产品制造商正开始采取积极的措施在他们的产品中集成防篡改安全启动过程，并添加加密元件来成功保障安全启动。