Trustworthy Secure and Measured Boot on a Raspberry Pi 4

Torben Woltjen[1], Michelle Jakobi[1], Michael Eckel[2], Janik Gorbracht[2] and Richard Sethmann[1]

[1]FRI, University of Applied Sciences Bremen, Germany
[2]ATHENE Center, Fraunhofer SIT, Darmstadt, Germany

torben.woltjen@hs-bremen.de
michelle.jakobi@hs-bremen.de
michael.eckel@sit.fraunhofer.de
janik.gorbracht@sit.fraunhofer.de
richard.sethmann@hs-bremen.de

Abstract: The Raspberry Pi is a widely used platform deployed across numerous domains, including Industry 4.0 environments. This places requirements on hardware and software in terms of IT and OT security. The whole platform has to be secured in a way that integrity and trustworthiness can be guaranteed – even when it is deployed in uncontrolled environments and operating outside the manufacturer's physical control. Currently, that often is not the case and basic Trusted Computing concepts are not implemented in Raspberry Pi-based platforms. The presented approach enables entities such as SIEM systems and backend services to make use of the established Trusted Computing mechanism Remote Attestation to verify the integrity of a Raspberry Pi 4 and its software. Remote Attestation, in turn, depends on Measured Boot to record integrity measurements (hash values) of each boot stage into a TPM 2.0. However, the Raspberry Pi 4 official bootloader supports Secure Boot only. It neither provides Measured Boot functionality nor integrates with a TPM 2.0, preventing the establishment of a hardware-anchored measurement chain required for Remote Attestation. To address these limitations, this paper proposes a method that combines Secure Boot with Measured Boot to enable Remote Attestation on the Raspberry Pi 4. This combination establishes a complete Chain of Trust. While it comes with some shortfalls and cannot defend against sophisticated hardware-based attacks, it still results in a much higher security level for the Raspberry Pi. The proposed concept makes use of the official bootloader's Secure Boot to load a signed second stage custom bootloader image based on U-Boot. Following that, U-Boot acts as the Measurement Root of Trust and measures the subsequent boot stages into the PCRs of a TPM 2.0. Although the proposal builds upon existing technologies, their integration into the Raspberry Pi boot chain has not been available in this form. The security enhancements have a significant impact on many use cases involving Raspberry Pi-based hardware, including industrial devices. It is particularly worth considering if the hardware platform already integrates a TPM 2.0 chip.

Keywords: Raspberry Pi 4, TPM 2.0, Secure boot, Measured boot, Remote attestation, U-Boot

1. Problem and Motivation

As the Raspberry Pi (RPi) and its industrial variant named *Compute Module* are used by many devices in the Industry 4.0 context, it is important to ensure platform security, to which the proposed hybrid secure-measured-boot solution makes a significant contribution. The goal is to improve the security of the RPi platform, so that it becomes more resilient against tampering and can be used as a verifiable and trustworthy entity in a computer network.

Secure Boot and Measured Boot are crucial security features designed to protect computer systems from malware attacks and unauthorized modifications, particularly during the boot process. This provides a trustworthy foundation for applications running on such a protected system. Secure Boot ensures that only trusted and signed software is executed during boot-up, preventing malicious bootloaders or tampered operating systems from being run on the computer. This is especially important for combating persistent threats in which malware is injected into a platform, causing the system to always start in a compromised state. Meanwhile, Measured Boot measures each stage of the boot process to a Trusted Platform Module (TPM), enabling security features such as Remote Attestation. This enables other systems to remotely verify the integrity of a system.

Together, these features shift trust in the boot process lower down from the system firmware to specifically designed tamper-resistant hardware components such as TPMs, providing a significantly higher level of protection against tampering and malware attacks. By offering an additional layer of security through precise monitoring and verification of the entire boot process, Secure Boot and Measured Boot have become essential in maintaining the integrity and trustworthiness of modern computing systems.

A complete and continuous Chain of Trust (CoT) directly from the initialization of the device hardware for *Measured* Boot is not possible on the RPi 4, because the official but proprietary bootloader does not have TPM

driver support built-in, therefore it cannot measure the next boot stage and subsequently cannot extend those measurement values to the TPM registers. Furthermore, as the official bootloader is closed-source, there is no development access for the community to add such support as a third-party driver. However, the official bootloader supports *Secure* Boot, which can ensure on a hardware level that the subsequent stage of the boot process can only run code signed by a trusted party. It should be noted that the RPi 4 is the first generation to support Secure Boot, which means that the usage of Secure Boot is not possible on previous RPi generations. (Raspberry Pi Ltd, 2023b)

A logical solution to the problem of missing Measured Boot support on the RPi 4 platform is to combine the two boot features and start measuring as early in the boot process as possible.

2. Background

This section provides the necessary background to understand the technologies discussed in this paper and ensures that the terminology used is clear and precise.

2.1 Raspberry Pi

The RPi is a popular compact single board computer used for private or industrial use cases. In this paper, the term *RPi* specifically refers to the RPi 4 platform. There is also a Raspberry Pi 4 Compute Module variant intended for industrial applications, which serves as the basis for several industrial-grade products. While most of the configuration in the context of Secure Boot is the same, small details can vary.

2.1.1 OTP

One Time Programmable (OTP) memory is a region of memory with multiple so-called registers (zones of memory) that can only be programmed once. The cells are basically fuses that are burned for storing ones and zeros as data, which makes them single-time writable.

2.1.2 EEPROM

Electrically Erasable Programmable Read-Only Memory (EEPROM) is non-volatile memory that can be rewritten but is slow to write. It is commonly used to store small amounts of configuration data.

2.2 Trusted Computing

The term Trusted Computing (TC) is shaped by the Trusted Computing Group (TCG) and describes a variety of standards that bring trust and security into computing as well as networking. The trust can be established using a hardware-based Root-of-Trust (RoT). (Trusted Computing Group, 2025a)

2.2.1 Trust

Trust, in this context of the TCG, refers to a reliable assumption about how a system will act. It is specified that trust "is meant to convey an expectation of behavior." (Trusted Computing Group, 2025b) This means that trust is not blind confidence but a reasoned belief grounded in evidence. In Trusted Computing, such trust enables predictable interactions and forms the basis for assessing whether a system is operating as intended.

2.2.2 TPM 2.0

The TPM is a type of Hardware Security Module (HSM) widely used as a hardware trust anchor to protect systems against attackers. Its broad acceptance comes from its ability to ensure system integrity, protect sensitive data, and enable secure operations in both consumer and enterprise environments. The current version which is also used for this paper is named TPM 2.0. (Trusted Computing Group, 2017)

2.2.3 Remote attestation

Remote attestation allows a system to remotely detect the software status of another system in a network. Basically, it can verify the hash values of the software executed on a remote system, by using challenge response mechanisms to securely receive the trustworthy measurement hash values from the TPM of the remote system. By comparing those hashes with known good values, a system's state can remotely be classified as secure or potentially compromised.

2.2.4 Root of Trust

The RoT is a "component that performs one or more security-specific functions, such as measurement […]. It is trusted always to behave in the expected manner, because its misbehavior cannot be detected […]" (Trusted

Computing Group, 2017). To assure this, a RoT must be intrinsically secure, as it provides the secure starting point for securing everything loaded after it. Therefore, it must be assumed that the official bootloader provides this RoT, being the initial component loaded on the RPi platform.

The *Root of Trust for Measurement* (*RTM)* should be implemented as early as possible in the boot process of a system, as its task is to "bootstrap the process of building a measurement chain for subsequent attestation of other firmware and software that is executed on the computer system" (NIST - National institute of standards and technology, n.d.).

2.3 Bootloader

The bootloader is part of the firmware and is stored in non-volatile memory, such as Read Only Memory (ROM) or EEPROM. It initializes the hardware and loads the operating system. It typically includes hardware initialization, boot device selection and operating system loader execution. It initializes the system memory layout through the memory map management. In case of Unified Extensible Firmware Interface (UEFI) with Secure Boot enforcement, the bootloader verifies the cryptographic signatures of loaded components to prevent unauthorized code execution.

2.3.1 Official bootloader

When speaking of the "official bootloader", this paper specifically means the proprietary bootloader shipped with the RPi, which is developed by the Raspberry Pi Ltd.

2.3.2 U-Boot

Contrary to this, U-Boot is a widely used open-source bootloader, available for RPi among many other (mostly embedded) platforms, however it can only be used as a second-stage bootloader on the RPi, loaded by the official bootloader.

U-Boot can boot the next stage in two different modes: Legacy and UEFI. The "legacy" mode is the classic and more simplistic way to boot embedded devices. It is how almost all RPi 4 Linux distributions are booted.

However, there is also the option to use UEFI on U-Boot which means that U-Boot exposes a simplified UEFI API targeted for embedded devices to the next boot stage. This also gives the option to use Secure Boot as defined by the UEFI specification. (The U-Boot development community, n.d.)

2.3.3 Secure Boot

Secure Boot describes a concept where a platform only boots an image that is signed by a trusted party. The image is signed with a private key, while the corresponding public key is stored securely on the platform so that it cannot be changed by an untrusted party. If the signature of the boot image cannot be verified with the public key on the platform, the boot process will be terminated with an error message.

Although various terms exist for this concept, this paper makes use of the term *Secure Boot*, as it is most widely recognized for this technology, originating in the UEFI specification, and is now supported by nearly all modern personal computers and servers. For example, the RPi bootloader flag to activate Secure Boot is called *Signed Boot* (although the official documentation (Raspberry Pi Ltd, 2023a) also refers to the feature as *Secure Boot*). In the U-Boot bootloader, this concept is called *Verified Boot* (Raspberry Pi Ltd, 2023b).

2.3.4 Measured Boot

The term *Measured Boot* refers to the process of measuring the hash of the code and the configuration data for each subsequent boot stage in the boot process of a system. The hash is typically stored securely in an HSM such as the TPM. This information can e.g. be forwarded to servers to determine whether a computer is allowed to log into a network. This process is known as Remote Attestation. In summary, Measured Boot provides the technological basis for Remote Attestation.

Like Secure Boot, Measured Boot checks the integrity of the system by itself but will not block the boot process in case of unexpected measurement values. That would only be achieved by additionally using Secure Boot.

2.4 Related Work

While all the basic technologies utilized in this paper have been individually explored and implemented before, no prior work has integrated these features into a continuous CoT and demonstrated the feasibility of combining Secure Boot and Measured Boot on the RPi platform.

There is official documentation available from the Raspberry Pi Ltd about how boot security can be implemented on the RPi 4. This includes a whitepaper describing the technology involved in achieving a CoT (Raspberry Pi Ltd, 2023b) as well as a how-to guide for how to use Secure Boot on the RPi 4 (Raspberry Pi Ltd, 2023a).

Important work has been done by Johannes Holland, who shows in his GitHub repository (Johannes Holland, 2022) how to integrate and compile TPM 2.0 support into U-Boot on the RPi 4. This lays the foundation for enabling Measured Boot on the RPi 4, because TPM support is needed to securely store measurement values when loading the subsequent boot stages.

3. Use Cases

The reasons why Measured and Secure Boot on a RPi can be important are best explained by different use cases and why these features offer an advantage for them. It will also be discussed which requirements have to be fulfilled for satisfying these use cases.

3.1 Industrial Use

In the context of Industry 4.0 and Industrial Internet of Things (IIoT), IT networks and thus information security is more important than ever, due to measurement values and control commands being transferred remotely. When these get compromised, this can lead to expensive damage of industrial equipment such as production machinery, possibly stopping production completely or even lead to human injuries. With Measured and Secure Boot, the attack surface can be significantly reduced.

Industrial-grade hardware based on the RPi Compute Module is available, such as the KUNBUS Revolution Pi 4 (KUNBUS GmbH, 2025). The device adapts the Raspberry Pi platform for use in industrial environments and includes an on-board TPM 2.0 which is needed for Measured Boot. Another example for such hardware is the Seeedstudio EdgeBox RPi 200 which is marketed as an *IoT Edge Device* or *Industrial Edge Controller*; however, it does not include a TPM 2.0 (Seeed Technology Co.,Ltd., 2025).

Requirements for industrial use cases are mainly reliability. A system must not shut down due to false positives; its behaviour should be deterministic. Also, performance should not be significantly impacted by security measures, especially not in terms of application software or data transfer. A longer time to boot a system can be acceptable, though.

3.2 Personal Use

The RPi is a commonly used component in DIY projects for smart home, home server and Home Theatre PC (HTPC) setups. Of course, such a central component in a home network has to be trustworthy. Not only can it possibly control devices such as smart door locks, it is also within the boundaries of the user's LAN, behind the firewall of the router. So, if the RPi was compromised, an attacker could use it to access the user's home network and perform Man-in-the-Middle (MitM) attacks to capture login credentials via DNS spoofing, phishing, and similar techniques. Measured and Secure Boot can help mitigate these attacks and make them harder to execute.

4. Threat Model

A threat model is defined to understand possible attack vectors. In the evaluation, it will be examined how well the implemented concept works to mitigate these threats. It is assumed that an attacker possesses the capabilities of the Dolev-Yao adversary, i.e., full control over network communication, including intercepting, modifying, replaying, and delaying messages. While the original Dolev-Yao model focuses on network-level attacks, this work adopts the model for a stronger hardware focus: the attacker can attempt to replace or roll back software and firmware on a device, both at rest and during updates. However, the attacker cannot break standard cryptography or the TPM's protection mechanisms.

4.1 Assets and Trust Anchors

The relevant entities involved in the boot process can be categorized as assets and trust anchors, which have to be analyzed:

- Assets: Second-stage bootloader (U-Boot), U-Boot environment, initramfs, Device Tree Binary (DTB)/overlays, Linux kernel, TPM PCRs and event log (for Linux Integrity Measurement Architecture (IMA)), attestation results.

- Trust anchors: System on Chip (SoC) boot ROM and the official RPi bootloader (EEPROM), OTP-fused hash of the customer public key, TPM 2.0 (including its Platform Configuration Registers (PCRs)), and standard cryptographic primitives.

4.2 Capabilities

This section outlines the main attack vectors as well as assumed capabilities of the attacker:

- Network: Full MITM on all channels; can replay quotes/logs, but cannot forge TPM quotes or signatures.
- Local software: Gains OS-level control (e.g. root), can modify /boot contents, U-Boot environment, initramfs, DTB, Linux kernel, but cannot alter OTP or the official bootloader.
- Physical hardware: brief hands-on access (swap SD card / eMMC, attach to USB/UART ports); no MITM attacks on the RPi bus system, e.g. no interception of messages between software and TPM on the SPI bus. No invasive chip attacks (no decap, microprobing, fault injection).

4.3 Security Goals

This section outlines the security objectives that the provisions in this paper are intended to achieve:

- Prevent execution of unauthorized second-stage bootloaders.
- Produce verifiable measurements for initramfs, DTB/overlays, Linux kernel and other relevant configuration to enable remote attestation.
- Detect (and optionally prevent) rollback to older, vulnerable components.
- Optionally enforce integrity of OS (U-Boot Verified Boot).
- Optionally bind secrets (e.g. LUKS keys) to expected PCR values.
- Optionally measure integrity of OS user-space via Linux IMA.

4.4 Out of Scope

- Invasive physical attacks against OTP/EEPROM/TPM (e.g. decapping, microprobing, fault injection).
- Hardware-based MITM attacks on RPis bus system, e.g., no interception of messages between software and TPM on the SPI bus
- Compromise of vendor signing keys for the official bootloader.
- Runtime attacks: Exploitation of application-layer vulnerabilities beyond Secure Boot and Measured Boot.

5. Concept

As described earlier, the RTM has to be placed as early in the boot process as possible. On the RPi, the earliest stage where this can be implemented is U-Boot, which is loaded directly after the official bootloader. Thus, it can be assumed that U-Boot acts as the RTM, while the TPM provides a secure and shielded location for storing the measurement values. Ideally, the official bootloader would serve as the RTM, as it is placed earlier in the boot process. However, implementing custom features in this proprietary software is not feasible.

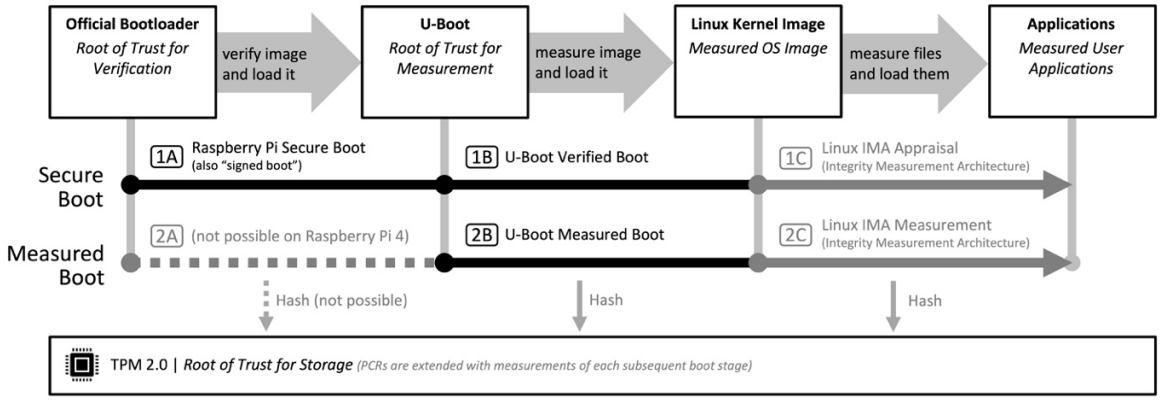

Figure 1: Chain of Trust for the Secure and Measured Boot process on the Raspberry Pi 4

The complete CoT is shown in Figure 1. The boot chain begins with the official bootloader which acts as a root of trust for the system and uses Secure Boot to securely load the signed U-Boot bootloader image.

Subsequently, U-Boot measures the Linux kernel image and other necessary boot data like the DTB and initial ramdisk and extends the measurement values to the TPM 2.0 PCRs as shown in Table 1. Now, U-Boot loads them into the RAM and runs the Linux kernel.

Table 1: Intended use of PCR banks according to the TCG TPM 2.0 Mobile Common Profile (Trusted Computing Group, 2015) compared to the real use by U-Boot legacy Measured Boot mode

PCR	TCG TPM 2.0 Mobile Common Profile	U-Boot (Legacy Mode)
0	Platform firmware code integrated into system board or SoC	U-Boot version string
1	Platform firmware data associated with code measured in PCR0	DTB and environment variables
2	Platform firmware code (including drivers) added by VARs	not in use
3	Platform firmware data associated with code measured in PCR2	not in use
4	Pre-OS diagnostics and OS loader code	not in use
5	Pre-OS diagnostics and OS loader data associated with code measured into PCR4	not in use
6	Platform-specific OS and Secure Boot code	not in use
7	Platform-specific Secure Boot policy data associated with code measured in PCR6	not in use
8	n/a	OS kernel image
9	n/a	initrd (initial ramdisk)
10	n/a	Linux IMA (not implemented here)

Optionally, Secure Boot can also be enabled for this stage, so that only a correctly signed image would be loaded. In the next step, the Linux kernel will load the user space and it is conceivable to use the Linux IMA mechanisms to measure the accessed data. However, the implementation of Linux IMA was not in the scope of this paper.

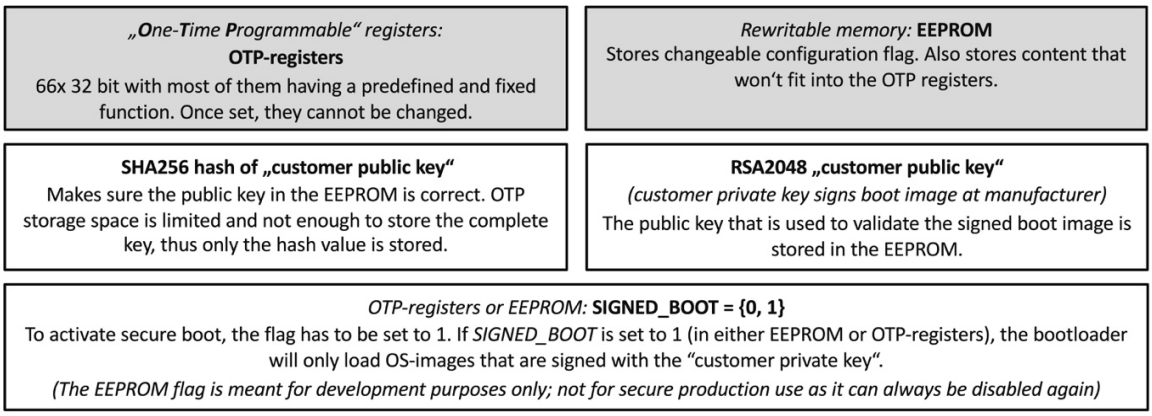

Figure 2: Data stored in OTP registers and EEPROM for Secure Boot on the RPi 4

To save the configuration data and cryptographic keys needed to implement Secure Boot, the RPi 4 uses OTP memory and EEPROM storage. The allocation of the OTP registers is shown in Figure 2. A Secure Boot implementation purely using EEPROM is possible for testing purposes, so that no permanent changes to the platform are persisted. For secure productive use, the OTP registers must be used. Changing the OTP registers is permanent by definition and cannot be undone, which is a security feature by itself. Secure Boot on the RPi 4 requires at least bootloader EEPROM version 2022-01-06. As soon as the SIGNED_BOOT flag in the OTP-registers is set to 1, only bootloader versions that support Secure Boot can be installed, to prevent circumventing the mechanism by simply installing an older bootloader without Secure Boot support. The public key for the signed boot image is stored in the EEPROM and its integrity is confirmed by a hash of it written into the OTP registers. This is done to save storage space in the relatively small OTP memory.

6. Evaluation

The proposed implementation brings a lot of advances in security for the RPi, but it still has some shortcomings that the user has to be aware of.

6.1 Considerations

When using the described mechanisms to secure a RPi, one should consider that there are multiple parties that they need to trust or else the whole boot process would not be trustworthy. It is crucial to recognize the need for trust in multiple parties to ensure the integrity of the boot process. That includes all parties involved, e. g. the Raspberry Pi Ltd as the creator of many of the scripts and tools involved as well as owning the private RSA key for signing the official bootloader. Trust must be placed in their ability to securely store this key and prevent any potential leaks.

Obviously, the manufacturers of the underlying hardware have to be trusted as well, that includes Broadcom as they are responsible for the CPU as well as the TPM manufacturer and others. Furthermore, the U-Boot developers, the implementation of their security features and the source code hosting and build architecture has to be trusted. Vulnerabilities in relied on software and libraries like OpenSSL, which is used to create the keys, also have to be considered.

6.2 Security Achievements

The proposed approach strengthens the security of the RPi platform by establishing a verifiable CoT during the boot process. RPi Signed Boot ensures that only customer-signed images can be executed, while U-Boot (legacy mode) measures critical boot components, including the U-Boot environment, initramfs, DTB/overlays, and the Linux kernel. The resulting measurements stored in the TPM PCRs enable the use of Remote Attestation on the Raspberry Pi platform.

Although not implemented in this work, the established measurement chain provides the foundation for extending integrity verification to the operating system using Linux Integrity Measurement Architecture (IMA) as demonstrated by Jäger, Lorych and Eckel (2022) or for binding secrets, such as LUKS keys, to specific TPM PCR states.

6.3 Limitations

6.3.1 Swap of SD card / eMMC

A physical attacker can always modify the content of the SD card or eMMC offline to alter boot files and software of the RPi. Most of these changes can be detected and a successful boot be prevented. It mostly depends on which specific steps from Figure 1 are implemented. Steps 1A to 1C will stop the system from booting if any of the monitored files are changed. Steps 2A to 2C will at least make unwanted file changes transparent to other systems via Remote Attestation.

That means that if U-Boot Verified Boot is not used, an attacker can still modify the Linux kernel but not the bootloader. With U-Boot Measured Boot in place, this change could still be detected via Remote Attestation.

6.3.2 Early-stage measurement limitation

With the official bootloader not having support for the TPM, the first stage (U-Boot) cannot be measured into the PCRs, so that trust in the context of Remote Attestation only begins in U-Boot.

6.3.3 OTP memory

The OTP memory can be considered being safe against tampering via software, because the data is burned into the hardware. However, they are not necessarily safe against sophisticated physical attacks which can possibly make the content visible or even change the data stored by them. While there are ways for hardware manufacturers to also protect areas like the OTP registers against hardware tampering (e.g. with wire frames to enclose the area of the chip for intrusion detection), it is unlikely that this has been done for the RPi 4 as it is not mentioned in any documentation. In summary, storage in the OTP registers can be considered safe enough for consumer-grade devices, but cannot be trusted in critical equipment that has to be protected against physical attacks from state attackers, which are categorized as Advanced Persistent Threats (APTs).

6.3.4 Leak of private key and downgrade attack

There are two major issues due to hardware limitations imposed by the RPi platform: If the public signing key is compromised, it cannot be replaced by a new one. So, if a private key used for signing valid U-Boot images is leaked, there is no mechanism to revoke it and an attacker could sign their own malicious U-Boot images with the leaked private key.

The same problem applies to the official bootloader: If a specific version of it contains a vulnerability, an attacker could always roll back to that vulnerable version which will always have a valid signature. Unlike other hardware platforms, the RPi does not allow for replacing the public signing key or implementing a minimum version policy. This is an improvement that the platform manufacturer should look into for future revisions of the RPi chipset. With the current limitations, a per-device or per-customer signing key can possibly limit the number of affected devices in some cases.

6.4 Complexity

The effort to set up a system with the proposed Secure and Measured Boot mechanisms is much higher than simply installing a standard Linux distribution on the RPi 4. Apart from that, some of the steps are significantly more complex than others. That especially is the case for step 1B (U-Boot Verified Boot) shown in Figure 1 which requires extensive U-Boot configuration and also introduces some practical problems. One example is the updatability of the Linux kernel, because it is merged into a U-Boot specific Flattened Image Tree (FIT) image which means the kernel cannot simply be updated from package repositories like it is commonly done in most Linux distributions. Generally, even if this problem could be solved, updates still need to be signed by the maintainer, i.e. in most industrial use cases the manufacturer of the device.

7. Conclusion and Future Work

This paper demonstrates that combining Secure Boot and Measured Boot significantly improves the security capabilities of the RPi platform by establishing a continuous CoT from early boot stages to the operating system. While certain limitations remain due to hardware and proprietary components of the platform, the proposed approach enables verifiable integrity and supports security mechanisms such as Remote Attestation.

The presented approach is particularly suitable for industrial use cases, where higher setup complexity is acceptable in exchange for increased trustworthiness. Future work includes evaluating UEFI on U-Boot to improve flexibility and extending the CoT into user space using Linux IMA.

Acknowledgements

This research is directly supported by the German Federal Ministry of Research, Technology and Space in the context of the project TRUSTnet (TRUSTnet Consortium, 2026) (ID: 16KIS1788 – Hochschule Bremen / 16KIS1786 – Fraunhofer SIT). Further, this research work was supported by the National Research Center for Applied Cybersecurity ATHENE (National Research Center for Applied Cybersecurity ATHENE, 2024). ATHENE is funded jointly by the German Federal Ministry of Research, Technology and Space and the Hessian Ministry of Science and Research, Arts and Culture.

Ethics declaration: This research did not involve human participants, animals, or personal data. No ethical approval was required.

AI declaration: AI tools were used during the preparation of this paper exclusively for language refinement, grammar correction, and stylistic improvements. The scientific content, technical concepts, analyses, and conclusions were developed by the authors. All input was reviewed and validated to ensure correctness.

References

Jäger, L., Lorych, D. and Eckel, M. (2022) "A Resilient Network Node for the Industrial Internet of Things", *ARES '22: Proceedings of the 17th International Conference on Availability, Reliability and Security,* New York, NY, USA, https://doi.org/10.1145/3538969.3538989.

Holland, J. (2022) *TPM 2.0 in U-Boot on Raspberry Pi 4,* [online], https://github.com/joholl/rpi4-uboot-tpm.

KUNBUS GmbH (2026) *Industrial Raspberry Pi: Reliable and flexible - Revolution Pi,* [online], https://revolutionpi.com/revolution-pi-series.

National Research Center for Applied Cybersecurity ATHENE (2024) *National Research Center for Applied Cybersecurity ATHENE c/o Fraunhofer Institute for Secure Information Technology SIT*. [online], https://athene-center.de/.

NIST - National Institute of Standards and Technology (n.d.) *Core Root of Trust for Measurement (CRTM)*, [online], https://csrc.nist.gov/glossary/term/core_root_of_trust_for_measurement.

Raspberry Pi Ltd (2023a) *How To Use Raspberry Pi Secure Boot*, [online], https://pip.raspberrypi.com/categories/1260-security.

Raspberry Pi Ltd (2023b) *Raspberry Pi 4 Boot Security,* [online], https://pip.raspberrypi.com/categories/1260-security.

Seeed Technology Co.,Ltd. (2026) *EdgeBox RPi 200 - Raspberry Pi IoT Edge Device | IoT Gateway | 4GB RAM, 32GB eMMC, WiFi*, [online], https://www.seeedstudio.com/EdgeBox-RPI4-A-4G32G-WiFi-p-4971.html.

The U-Boot development community (n.d) *UEFI on U-Boot — Das U-Boot documentation*, [online], https://docs.u-boot.org/en/latest/develop/uefi/uefi.html.

Trusted Computing Group (2015) *TPM 2.0 Mobile Common Profile*, [online], https://trustedcomputinggroup.org/resource/tcg-tpm-2-0-mobile-common-profile.

Trusted Computing Group (2017) *TCG Glossary*, Version 1.1 Revision 1.0 edn, [online], https://trustedcomputinggroup.org/resource/tcg-glossary/.

Trusted Computing Group (2025a) *Trusted Computing*, [online], https://trustedcomputinggroup.org/trusted-computing/.

Trusted Computing Group (2025b) *Trusted Platform Module Library Part 0: Introduction*, Version 184th edn, [online], https://trustedcomputinggroup.org/resource/tpm-library-specification/.

TRUSTnet Consortium (2026) *TRUSTnet project – Trusted Core Network in Real and Simulated Industry 4.0 Environments*, [online], https://trustnet-project.de.