January 2020

Supply Chain Verification of Hardware Components Using Open-Source Root of Trust

Vadim Sukhomlinov

Marius Schilder

Andrey Pronin

Randall Spangler

Supply Chain Verification of Hardware Components Using Open-Source Root of Trust

Abstract:

This publication describes techniques and apparatuses for an open-source silicon-based Root of Trust (RoT) (a security chip, a RoT chip) solution for supporting supply chain verification of hardware components embedded in or on a circuit board (*e.g.*, motherboard). In aspects, an open-source read-only (RO) Component Verifier, which is part of an RO Firmware, is provided to establish trust at all levels, such as a field-modifiable or read-write (RW) Firmware, an operating system (OS), cloud computing, component manufacturers, and so forth. The security chip includes two classes of non-volatile memory (storage): RO non-volatile memory (*e.g.*, RO NVRAM), which is accessible only by the RO Firmware, and RW non-volatile memory (*e.g.*, NVRAM), which is accessible by the RO Firmware and the RW Firmware. The security chip stores private keys (*e.g.*, EK, AIK) used in for provisioning component Secrets in the RO NVRAM, preventing access to the RW Firmware. Without sharing any details, the RO Component Verifier of the RO Firmware is able to access the contents of the RO NVRAM. Such an intrinsically secure communication channel between the RO Firmware and the RO NVRAM, as well as preventing access to the RO NVRAM by the RW Firmware, is guaranteed by the hardware design of the security chip. This solution enables component manufacturers to use computationally inexpensive common protocols for identity verification (*e.g.*, MARS, DICE) to communicate the Secrets associated with the components, while entrusting the open-source security chip to protect these secret keys.

Keywords:

Firmware, Root of Trust (RoT), open-source, hardware, cryptographic encryption, symmetric cryptography, Measurement and Attestation RootS (MARS), Device Identifier Composition Engine (DICE), reboot, Trusted Platform Module (TPM), security chip, memory, read-only memory, device identifier (ID), supply chain, component, circuit board, motherboard.

Background:

Original device manufacturers (ODMs) of user equipment (UE), such as smartphones, notebooks, laptops, and the like, often embedded in or on a circuit board (*e.g.*, a motherboard) a myriad of hardware components. The components may be integrated circuits (ICs), application processors (APs), controllers, and so forth. The components perform complex functions and frequently include read-write (RW) firmware. These components are often activated during the early stages of a boot sequence of the UEs and are responsible for important aspects of the functionality, safety, and security of the UE. A vulnerability in any of these components can compromise the security of the UE.

These security risks are one reason why UE manufacturers follow standard supply chain practices that include sourcing hardware components from trusted component manufacturers. However, even following such practices, the risk that hardware components have been compromised (*e.g.*, replaced by adversaries, firmware altered) remains, raising a problem of trust in the supply chain of the components.

In an attempt to address some of these concerns, ODMs may perform on-chip verified boots, perform cryptographic identifications of the components, ensure secure component manufacturing, check and monitor the boot firmware, ensure silicon physical security, and/or

ensure transparent development of the UE and the components that are embedded in or on the motherboard of the UE. The implementation of some of these procedures requires that the components have integrated identities, however some components do not have integrated identities. One reason may be that a strong cryptographic method, such as asymmetric cryptography, which uses a public key and a private key, is expensive silicon-wise.

Alternatively, symmetric (instead of asymmetric) cryptography is more suitable for some of the components, because symmetric cryptography often requires fewer logic gates. Symmetric cryptography can successfully be utilized using several standards supported by the Trust Computing Group (TCG). Specifically, TCG standards, such as a Trusted Platform Module (TPM), Measurement and Attestation RootS (MARS), and Device Identifier Composition Engine (DICE), can be used to provide a strong cryptographic identity. Component manufacturers can use these standards to successfully send symmetric cryptographic identification to UEs with circuit boards that have a closed-source read-only (RO) Component Verifier, which can securely hold symmetric keys. The component manufacturers, however, need further assurances to entrust their Secret (private) cryptographic keys to an open-source RO Component Verifier, which utilizes an open-source firmware, because the symmetric key can be easily readable by any developer that has access to the open-source firmware of the RO Component Verifier.

Description:

This publication describes techniques and apparatuses for an open-source silicon-based Root of Trust (RoT) solution for supporting supply chain verification of hardware components. In aspects, an open-source RO Component Verifier establishes trusted communication between a firmware, an operating system (OS), cloud computing, component manufacturers, and so forth.

Figure 1 illustrates an architecture for implementing the silicon-based RoT solution (such as a security chip or an RoT chip) for supporting supply-chain verification of hardware components.

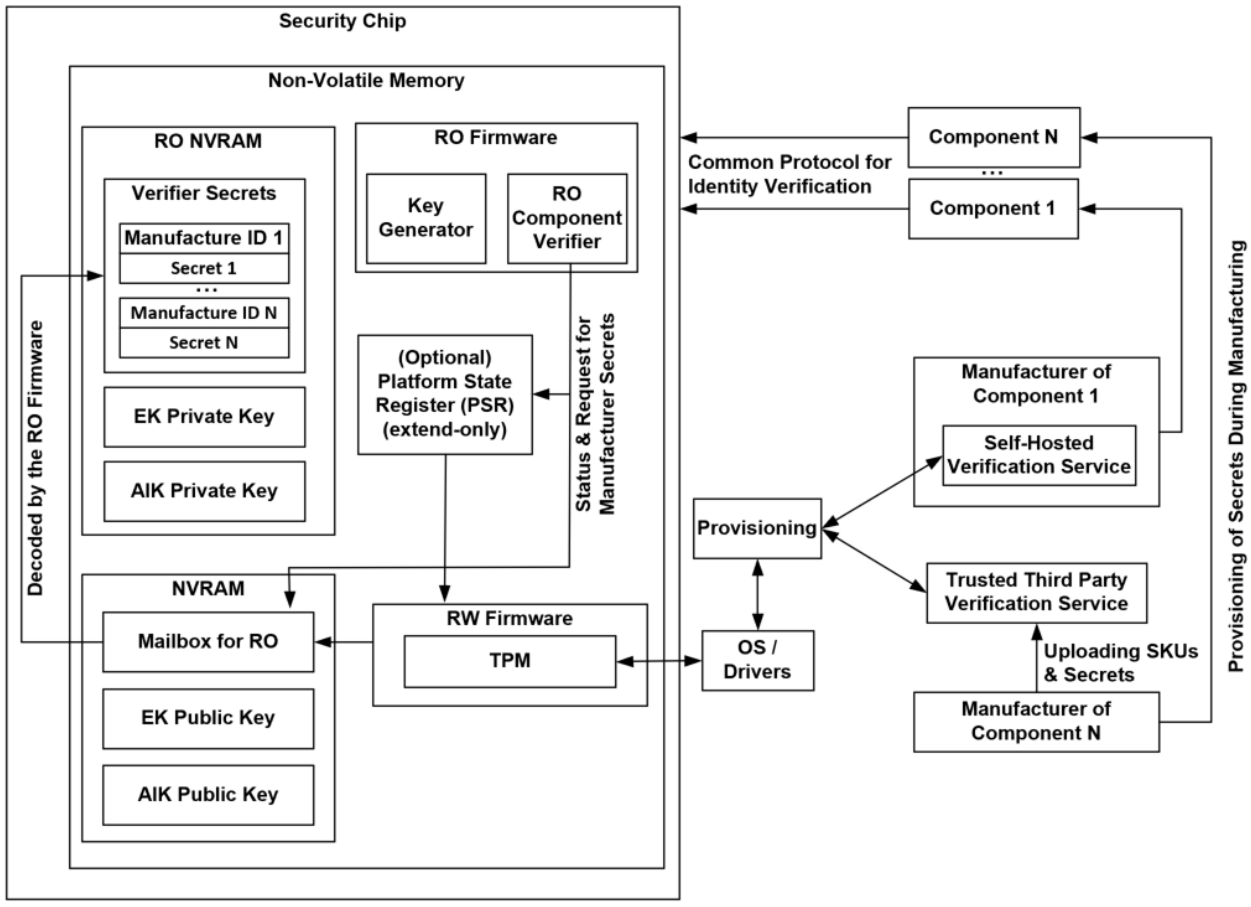

Figure 1

The security chip architecture illustrated in Figure 1 includes two classes of non-volatile memory: RO non-volatile memory and RW non-volatile memory. The RO non-volatile memory may be RO non-volatile random-access memory (RO NVRAM). The RO NVRAM stores and is accessible only by an RO portion of the firmware (RO Firmware). On the other hand, the RW non-volatile memory, which may be any non-volatile random-access memory (NVRAM) (*e.g.*, static random-access memory (SRAM)), is accessible by an RW portion of the firmware (RW Firmware, field-modifiable firmware, or updatable firmware). The source code of the RO Firmware can be open source and can be made available for security inspection. Similarly, the

hardware design of the appropriate security mechanism can be open source and can be made available for security inspection. The hardware design grants access to specific areas only to the RO Firmware, ensuring that the RO Firmware can trust the RW Firmware.

The RW Firmware may implement functionalities, such as a TPM functionality, a smart card, a hardware security module, and so forth. Of note, the RW Firmware implements a communication channel between the RO Firmware, the OS, and management applications, and subsequently, external services. Such a mechanism can be an extension of common protocols used by the TPM, the smart card, or can be vendor-specific.

The RO Firmware implements one or several public key cryptography-schemes like Rivest-Shamir-Adleman (RSA) or Elliptic Curve (EC), which are used for the attestation of the RO Firmware to component manufacturers (*e.g.*, Manufacturer of Component 1, Manufacturer of Component 2, …, Manufacturer of Component N) and for securing communication channel(s). The public keys can include an Endorsement Key (EK) or chip identity key with a certificate from a security chip vendor, Attestation Identity Keys (AIKs) for each component requiring provisioning of a secret, and encryption keys sent to the manufacturers. The RO Firmware stores public parts of the AIKs, the EK, and the encryption keys in the NVRAM, which are accessible by the RW Firmware and may be communicated to third parties, such as component manufacturers. The use of an EK and AIKs is similar to TPM, where the EK is used to receive a certificate for the AIK from a private certificate authority (PCA) and the AIKs with their certificate are used for the attestation of the RO Firmware while preserving privacy. The specific process for obtaining an AIK certificate may be vendor-specific for the security chip and may resemble the TPM process.

Some portion of the RW NVRAM is allocated as a Mailbox for RO. The Mailbox for RO facilitates communication between the RO Firmware and the RW Firmware. The Mailbox for RO helps preserve the privacy of communication between a component manufacturer and the RO Firmware of the security chip. Additionally, in the Mailbox for RO, the RW Firmware stores encrypted data received from the component manufacturer aided by OS Drivers and provisioning applications (illustrated as "Provisioning" in Figure 1), but the RW Firmware does not possess keys to access information provided by the component manufacturer. Such information can only be decrypted using keys that are accessible by the RO Firmware.

On the other hand, the RO Firmware can access the private part of AIKs, EK, and encryption keys, either by storing them in RO NVRAM or by having exclusive access to seed for key derivation. In addition to the AIK and EK private keys, the RO NVRAM also stores Verifier Secrets, which may include Manufacture IDs and associated Secrets (*e.g.*, credentials). As is illustrated in Figure 1, the Verifier Secrets in RO NVRAM may contain a Manufacture ID 1 and an associated Secret 1, a Manufacture ID 2 and an associated Secret 2, …, and a Manufacture ID N and an associated Secret N, where N can be any finite integer. The Verifier Secrets portion of the RO NVRAM is read-write accessible by the RO Firmware during the early stages of a boot sequence of a UE. Note that, the Verifier Secrets portion of the RO NVRAM is inaccessible by the RW Firmware.

The RO Firmware includes the RO Component Verifier and a Key generator (Key-gen). The RO Component Verifier may be a generic RO Component Verifier of third-party components implementing some of the approaches proposed by TCG DICE or MARS standards. The RO Component Verifier sends status updates and requests for component Secrets to the Mailbox for RO, and optionally to a Platform State Register (PSR) (extend-only), which may interface with the

RW Firmware. The RW Firmware implements externally facing functionality that serves as an untrusted channel for communication between RO Firmware and the rest of the world, for example, communication between the RO Firmware and the OS.

RO Firmware Functionality

Any manufacturer that uses the security chip architecture illustrated in Figure 1 can verify and inspect the open-source hardware design of the security chip to ensure that the RO NVRAM is only accessible by the RO Firmware. Further, the RO Firmware of the security chip can also be verified and inspected to ensure proper protection of secrets. Without sharing any details regarding Secrets of a component, the RO Component Verifier accesses the contents of the RO NVRAM. Such an intrinsically secure communication channel between the RO Firmware and the RO NVRAM is guaranteed by the hardware design of the security chip.

When a UE that includes the described security chip is powered up or after the reboot, the security chip starts executing the code of the RO Firmware. As is illustrated in Figure 1, the RO Component Verifier, which is part of the RO Firmware, through the Mailbox for RO, checks if an EK private key is stored in the RO NVRAM. If the RO Component Verifier determines that an EK private key is not available, then the Key-gen generates an EK private key from a seed to match an EK certificate. With the EK key confirmed, the RO Component Verifier, enumerates third-party components that the RO Component Verifier is connected to using various protocols, such as a serial peripheral interface (SPI), inter-integrated circuit (I2C, I^2C, or IIC), 1-wire, and so forth, using a common protocol (*e.g.*, MARS, DICE, TPM) and receive the identity of each component (Manufacture ID).

The RO Component Verifier checks if Verifier Secrets (symmetrical keys provided by component manufacturers) for the discovered Manufacture IDs are available in the Mailbox for RO. If the associated Secret of the component is available, the RO Component Verifier follows a common protocol (*e.g.*, MARS, DICE) for identity verification to verify the identity of the components. Results of the verification, including a list of known components, components for which the Verifier Secret is not available, and components that failed identity verifications, are signed by the RO Firmware EK, are posted for the RW Firmware, and are reflected in the PSR, which can only be extended. Thus, the PSR reflects the state of the platform. The desirable state of the PSR, in which all components are successfully verified for a particular type of UE, is known by the RW Firmware.

For each component for which a Verifier Secret is not available, an AIK (a new key or a reused key that was already created across multiple components/manufacturers) and an encryption key is generated. Also, a request for the certificate of a Verifier Secret from the PCA for each AIK is posted to the RW Firmware, which in turn makes it available to the OS and the provisioning applications. Each public part of the encryption key is signed by the AIK and also posted to the RW Firmware. Then, the provisioning application can combine the certificate from the PCA with the encryption key signed by the appropriate AIK and send the request to the component manufacturer to obtain the associated Verifier Secret. The RW Firmware stores each response (containing an encrypted Verifier Secret signed by the component manufacturer) in the Mailbox for RO and reboots the security chip to pass control to the RO Firmware to repeat the process with updated information as mentioned earlier. Note that the RO Firmware may have a hardcoded whitelist of manufacturers, including their public keys, to verify the authenticity of the received data.

Then, the RO Component Verifier transfers control to the RW Firmware. If the associated Secret is not available, the RO Component Verifier sends a request to the RW Firmware for further inquiry regarding the identity of the components. After sending the request to the RW Firmware, the RO Firmware disables access to the RO NVRAM, verifies the signature of RW Firmware, and passes control to the RW Firmware.

RW Firmware Functionality

The RW Firmware checks if there are outstanding requests for AIK certification by the PCA (*e.g.*, receipt of a Manufacture ID for further inquiry from the RO Component Verifier). If there is an outstanding request for AIK certification, the RW Firmware notifies the OS of the same. The RW Firmware then checks if there are any outstanding requests for secret verification from any component manufacturer. Note that Figure 1 illustrates Manufacturer of Component 1, Manufacturer of Component 2, …, Manufacturer of Component N, where N can be any finite integer.

For utilization by a provisioning software (illustrated as "provisioning" in Figure 1) of the OS, the RW Firmware may expose an interface to retrieve a list of the abovementioned requests for Secret verification and to provision the Secrets received from the component manufacturers. In response to receiving an encrypted Secret from a component manufacturer, the RW Firmware saves the encrypted Secret in the Mailbox for RO located in the NVRAM. Then, the RW Firmware transfers control to the RO Firmware to verify whether the encrypted Secret from the component manufacturer matches the Secret stored in the Verifier Secrets in the RO NVRAM. Thus, the RW Firmware may facilitate the transferring of Secrets through unsecured communication channels,

but the Secrets themselves cannot be read, decrypted, nor decoded by the RW Firmware, the OS, or any other portions of the unsecured communication channels.

Provisioning of Secrets for Component Verification

In aspects, a component manufacturer (*e.g.*, Manufacturer of Component 1, Manufacturer of Component 2, …, Manufacturer of Component N) sends a stock-keeping unit (SKU) and an associated Secret of a component to a Verification Service. The Verification Service may be hosted by the component manufacturer, or the component manufacturer may entrust a trusted third party to communicate any Secrets, as is illustrated in Figure 1. During part of the verification process, each Manufacture ID is validated for whether it is on a whitelist for a current platform. If the Manufacture ID is not on the whitelist, then a potential trust issue may be present. In aspects, the Manufacture ID may use a uniform resource locator (URL) as part of a component identifier. The provisioning software (illustrated as "provisioning" in Figure 1) may connect to the specified URL and present the AIK public key that is associated with the component. The component manufacturer can validate that the request comes from an authentic source by checking the validity of the AIK certificate. Then, the component manufacturer releases the Verifier Secret of a particular component, encrypts the Verifier Secret with the public part of the encryption key, which was provided by RO Firmware, signs with its own signing key, and produces and stores a blob in a standard format containing Secrets, which are required for verification. The actual Verifier Secret is encrypted with the encryption key provided by the RO Firmware and, as such, can be decrypted only by the RO Firmware that has exclusive access to the associated part of the private encryption key. Note that the private encryption key can also be generated from seed and is accessible only by the RO Firmware. Specifically, the blob is received by the provisioning

software and passed to the RW Firmware of the security chip. Then, the RW Firmware stores the blob in the Mailbox for RO. When all responses are received, the provisioning software notifies the security chip to start a reboot of the security chip and to execute the code of the RO Component Verifier(a part of the RO Firmware).

Component Manufacturer Verification Service

In aspects, the Verification Service (*e.g.*, a self-hosted Verification Service, a trusted third-party Verification Service) establishes a trusted communication channel between the security chip and the component manufacturer, and the Verification Service provides data to establish the authenticity of the components. In other aspects, the Verification Service determines if a request is coming from a valid device containing the described security chip and from a valid customer. By utilizing the PCA and the AIK certification, the component manufacturer can verify that the components are used by the intended customers, meet export requirements, and match the content of the AIK certification (*e.g.*, Manufacture ID, Serial Number, Component Type) of the components.

In detail, using the verification service, the PCA, and the AIK certification, the component manufacturer can validate that the security chip contains the associated Secret of the component. To speed up part of the component verification process, the component manufacturer may record public data of the component (*e.g.*, Manufacture ID, Serial number, Component Type) onto a public database (or blockchain), so the OS can perform the initial verification. This database may contain part of the component's public key that is used in the verification process. Note that the RW Firmware, the OS, nor any other untrusted communication channel can access the AIK private key, the EK private key, or the associated Secret of the component.

Component Behavior

Each component follows a common protocol (*e.g.*, MARS, DICE) to prove the identity of the component without needing to use asymmetric cryptography. Part of an identity verification request by a component may be the public data of the component, such as Manufacture ID, Serial number, and Component Type. Then, the RO Component Verifier uses the public data of the component to locate the associated Secret, which is stored in the RO NVRAM. In addition, the RO Component Verifier executes the common protocol to ensure that the component has the associated Secret and that the Secret that is stored in the RO NVRAM matches the Secret of the component.

Summary

In summary, the techniques and apparatuses include: (a) implementation of a generic RO Component Verifier, as part of an open-source RO Firmware, which verifies hardware components that are used in an open-source circuit board (*e.g.*, a motherboard); (b) the circuit board includes an open-source security chip; (c) the security chip includes an RO non-volatile memory area, herein referred to as RO NVRAM, that is accessible only by the RO Firmware; (d) implementation of an open-source RW Firmware, which serves as an unsecured communication channel between an OS, cloud computing, component manufacturers, and so forth, without the ability to read, decode, nor decrypt any Secrets associated with the components; (e) implementation of a Mailbox for RO that facilitates communication between the RO Firmware and the RW Firmware, where any communicated Secrets can only be decrypted by the RO Firmware; (f) implementation of an optional PSR, which can only be extended (*e.g.*, PSR = HASH (old_PSR || value)) and brought to a known value on a reboot of the circuit board; (g) the ability to perform multiple reboots that can

execute a common protocol for identity verification (*e.g.*, MARS, DICE) and can provision component Secrets between the RO Firmware and the component manufacturer; and (h) the RO Component Verifier can communicate in parallel with one or more component manufacturers. These methods can be complementary to classical cryptographic methods using asymmetric cryptography when the component supports them.

References:

[1] Bhat, Akshay. "Secure Boot and Encrypted Data Storage." Timesys. July 13, 2017. https://www.timesys.com/security/secure-boot-encrypted-data-storage.

[2] Patent Publication: US20160359635A1. Tamper-protected hardware and method for using same. Priority Date: March 11, 2011.

[3] Patent Publication: US20150154424A1. Method and Apparatus for Secure Execution Using a Secure Memory Partition. Priority Date: June 30, 2000.