Akraino Platform Security Questionnaire

March 2021

Daniil Egranov, Arm

Platform and Firmware Assessment Questionnaire

- This applies to the hardware, firmware that comprise the Platform Security RoT that forms the Secure Processing Environment (SPE).
- This questionnaire provides assessment for the platform security components based on Immutable Platform Root of Trust and Platform Security Root of Trust.

Immutable Platform Root of Trust

1. **The chip shall support a hardware mechanism(s) to isolate the Secure Processing Environment (SPE) from the Non-secure Processing Environment (NSPE).**

(Describe how isolation is implemented, for example through TrustZone or dual cores.) Example of response for Yes: The Cortex-M33 (ARMv8-M architecture) supports TrustZone. The Secure Processing Environment is executed in secure mode.

Immutable Platform Root of Trust (Continue)

2. **The chip shall support Secure Boot, initiated from code in the immutable Platform Root of Trust. Note: that asymmetric signing is expected, however, symmetric signing can be accepted.**

(Describe which cryptographic functions and key sizes are used for secure boot, and how the cryptography is implemented, such as use of a hardware cryptographic accelerator or software in immutable code. Also describe how the Immutable code is implemented and if in some form updateable on-chip memory (such as EEPROM or Flash) how that is locked.)

Immutable Platform Root of Trust (Continue)

3. **(If applicable) The chip shall support a security lifecycle, i.e. protecting critical security parameters and sensitive data based on device lifecycle state and enforcing the rules for transition between states. Lifecycle states can typically be classed as follows, i) non-secure assembly and test, ii) provisioning, secured provisioned and operational, iii) decommissioned, and iv) debug, if debug of a secured provisioned device is supported.**

(Describe supported lifecycle states and transition rules.)

Immutable Platform Root of Trust (Continue)

4. **The chip shall support the storage or derivation of following minimum set (or equivalent) of critical security parameters, in such a way that prevents unauthorized reading and resists tampering by means such as physical, electrical or software (such as external probing of the chip for confidential data):**

   - **A secret Hardware Unique Key (HUK), with at least with 128-bits of entropy, used for deriving other device secrets**
   - **A Platform RoT Public Key, or hash of, used for authenticating the first updateable firmware component code during secure boot. If symmetric signing is unavoidable, the key must be unique per device.**
   - **A secret attestation key and identifier that uniquely identifies the attestation key**
   - **An identifier that uniquely identifies the Platform Security RoT on the chip.**

   These keys and identifiers may be injected during chip manufacture or during the manufacture of the device or derived from the HUK. They can also be derived from a Physically Unique Function (PUF).

(Describe key size for each key, and if applicable the key derivation method for the Attestation Key. If HUK is derived from a PUF, provide a rationale of key uniqueness. Describe the protection of the functions to read the keys. Also describe how the chip data are protected from tampering.)

Platform Security Root of Trust

1. **The Platform Security RoT shall support secure update of firmware and any of Application RoTs. Updates may be delivered either from locally connected devices (such as removable media) or from remote servers. Updates shall be validated by the Platform Security RoT to check integrity and authenticity prior to execution (see .1 in Immutable Platform Root of Trust section) and, optionally, before installation. This includes the executable code and any related data, such as configuration data or a manifest. The cryptography used shall comply with requirements (updates (see .4 in Platform Security RoT section)).**

(Describe how updates are validated, including the cryptographic algorithms, the key size and where the keys used for validation are stored. Justification is required if local validation of update from remote servers prior to installation cannot be supported, typically due to resource constraints.)

Platform Security Root of Trust (Continue)

2. **The update mechanism shall prevent unauthorized rollback of updates (see .1 in Platform Security RoT section) and protect the current reference firmware version number in an anti-rollback counter, in secure storage (for example, protected flash or OTP). A mechanism may be provided to support authorized rollback for recovery reasons. Anti-rollback is strongly recommended but not mandatory in PSA.**

(Describe the versioning information used to detect rollback and how it is protected in integrity and against rollback and over or underflow. If supported, describe how authorized rollback is implemented.)

Platform Security Root of Trust (Continue)

3. **The Platform Security RoT shall perform access control for modification and use of Platform Security RoT critical security parameters and for System software or Device sensitive data managed by the Platform Security RoT. For example, the Platform Security RoT shall control access to any such data stored in a protected flash region or in OTP.**

(Describe the System software subjects concerned by access control and how they are identified or authenticated)

Platform Security Root of Trust (Continue)

4. **The Platform Security RoT shall use best practice cryptography for protection of its assets, as recommended for instance by national security agencies. This includes the provision of a suitable source of random data. There should be no reliance on proprietary cryptographic algorithms or customization of standard cryptographic algorithms. Platform Security requires equivalence of at least 128-bit security. NB: Weak cryptographic algorithms or key sizes may be available for specific uses (e.g. legacy) and with specific guidance. They shall not be used in any way that reduces the security of the best practice cryptography. A TRNG or a suitably seeded Deterministic Random Bit Generator can be used.**

(List the cryptographic algorithms provided by the Platform Security RoT and the supported key sizes. Also describe how random number generation is performed.)

System Software Assessment Questionnaire

- This section applies to the software executing in the Non-secure Processing Environment (NSPE).
- The provided answers apply only to the context in which the System software is used. For instance, the vendor may only provide the cryptographic algorithms that are used, not all the algorithms supported by the System software.

System Software Security

1. **The System software shall support update of the system software and the application specific software, either from locally connected devices (such as removable media) or from remote servers.**
   Updates shall be validated by the system software or the Platform Security RoT to check the integrity and authenticity prior to execution and, optionally, installation. This includes the executable code and any related data, such as any manifest and configuration data.
   The cryptography used shall comply with requirement (5.).

(Describe how updates are validated, including the cryptographic algorithms, the key sizes and where the keys used for validation are stored. Justification is required if local validation of an update from remote servers prior to installation cannot be supported, typically due to resource constraints.)

System Software Security (Continue)

2. **The update mechanism shall prevent unauthorized rollback of system software, any applicable application software and authentication data. A mechanism may be provided to support authorized rollback for recovery reasons.**
   Anti-rollback is strongly recommended but not mandatory in Platform Security.

(Describe the versioning information used to detect rollback and how it is protected in integrity and against rollback and overflow. If supported, describe how authorized rollback is implemented. Note that use should be made of the Platform Security RoT for the most secure solution.)

System Software Security (Continue)

3. **The System software shall rely only on the Platform Security RoT for all queries of the Platform Security RoT (chip) identity.**

(Describe how the Platform Security RoT identity is used in preference to other identities that may exist.)

System Software Security (Continue)

4. **The System software shall use secure storage to protect sensitive data and provide this functionality for application data. It shall additionally bind the sensitive data to a specific device instance and, if supported, security lifecycle state (3.).**
   The cryptography used for secure storage shall comply with requirement (5.).

(Describe how secure storage is implemented. Note that use should be made of the Platform Security RoT secure storage service for the most secure solution.)

System Software Security (Continue)

5. **The System software shall use best practice cryptography as required by applicable standards or recommended by national security agencies, covering choice of algorithms and key lengths, and random number generation based on the identified threats. There should be no reliance on proprietary cryptographic algorithms or customization of standard cryptographic algorithms.**
   This Platform Security requires equivalence of at least 128-bit security level.

(Describe the cryptographic algorithms provided by the System software, supported key sizes and how they are implemented. Note that use should be made of the Platform Security RoT cryptographic service for the most secure solution.)

System Software Security (Continue)

6. **For two-way communication protocols and for each network interface, the System software shall provide the ability to authenticate remote devices and servers when establishing a connection.**

(Describe how this requirement is met.)

System Software Security (Continue)

7. **The System software shall provide the ability to encrypt and integrity check data exchanged with remote devices and servers.**

(Describe how this requirement is met.)

System Software Security (Continue)

7. **The System software shall use secure protocols, compliant with requirement (4.), for authentication and encryption of two-way communication. The selected protocols shall not leak data that would lead to the identification of vulnerable devices.**

(Describe how this requirement is met.)

System Software Security (Continue)

9. **Functionality that is not needed for the intended use of the System software shall not be installed, or shall be disabled if non-installation is not practical.**

(Describe how this requirement is met.)

System Software Security (Continue)

8. **The System software shall support an attestation method that can be used to prove the genuineness of the device. If possible, the current security lifecycle state of the device should be included.**

(Describe how this requirement is met. Note that use should be made of the Platform Security RoT secure attestation service for the most secure solution.)

System Software Security (Continue)

9. **The System software should provide logging of security relevant events and errors. The log should include sufficient details to determine what happened and should be integrity protected.**

(Describe how logs are protected and how they can be retrieved if necessary.)

System Software Security (Continue)

10. **(Optional) If the System software supports logging, it shall restrict access to the log files to authorized users only.**

(Describe how this requirement is met.)

System Software Security (Continue)

11. **Data input via network and any other interfaces shall be validated defensively against malformed input. Data transferred via critical system software Application Programming Interfaces (API) shall be validated defensively against malformed input.**

(Describe how this requirement is met.)

System Software Security (Continue)

12. **Where supported, the System software shall enable the execution of application specific software and system software with the lowest level of privilege necessary for the intended function.**
Where supported, each authenticated user, application, process, etc., shall have limited privileges based on pre-determined and/or securely configurable access controls.

(Describe how this requirement is met.)

System Software Security (Continue)

13. **If the System software has a mechanism to reset passwords and critical security parameters they shall not be resettable to any universal factory default value. Such data must not be easily determined by automated means or obtained from publicly available information.**

(Describe how this requirement is met.)

System Software Security (Continue)

14. **If the System software makes use of passwords they should conform with security best practices, in particular, password length and complexity, and the number of failed authentication attempts (refer for instance to NIST SP 800-63B guidelines for memorized secrets).**
    Where default passwords are used, they must be unique per device and must not be easily determined by automated means or obtained from publicly available information.

(Describe how this requirement is met.)

System Software Security (Continue)

15. **If the System software makes use of critical security parameters for user authentication, the cryptography used for that feature shall comply with requirement (4.).**

(Describe the cryptographic algorithms and key sizes used for user authentication.)

System Software Security (Continue)

16. **If the System software allows security-relevant configuration changes via a network or other interface, the related configuration change shall only be accepted after authentication.**
    Examples of security-relevant changes include:
    - **access control management for remote or local users, configuration of network keys,**
    - **passwords policy (such as changes or thresholds), update policy (such as query frequency, automatic installation, server address, rollback),**
    - **configuration of cryptography (such as default key length), access to network interfaces and authentication policy (such as account lock thresholds after failed authentication attempts).**

(Describe how this requirement is met.)

System Software Security (Continue)

17. **If the System software allows persistent storage of personal configuration data it shall allow only the owner or an authorized entity to read and erase this data.**

(Describe how this requirement is met.)

Questions