TPM 2.0 Key Certification

Sarah Lavinia Johnson

December 1, 2022

Relevant Background Definitions

Key Attributes

FixedTPM: non-duplicable
Restricted: operations are limited to TPM-generated data

Key Types

Primary: Created by TPM based on the current Primary Seed when executing the TPM2_CreatePrimary command. May be persisted within the TPM. Otherwise must be recreated after a TPM reset.

Ordinary: Created by TPM based on seed taken from the RNG when executing the TPM2_Create command. Must be the child of another key. May be persisted within the TPM or persisted external to the TPM in the form of an encrypted key blob. The blob is only loadable using the parent key's authorization.

Certificates

X.509 Digital Certificate

Public key and data about the subject (i.e., identity) signed by a certificate authority

Validating a certificate requires a certificate chain in order to verify a chain of trust to a trust anchor (e.g., a root certificate)

Secure Device Identifiers

Definition

Identifier that is cryptographically bound to a device

Requirements

Attestation Key (AK): FixedTPM Restricted signing key
Device Identification Key (DevID): FixedTPM non-Restricted signing key

Initial Keys (IAK/IDevID)

Created by OEM at manufacturing time
Should be Primary keys
Recommended to be used only for enrollment of an LAK/LDevID
Typically the IAK certificate is the trust anchor for certificates created later

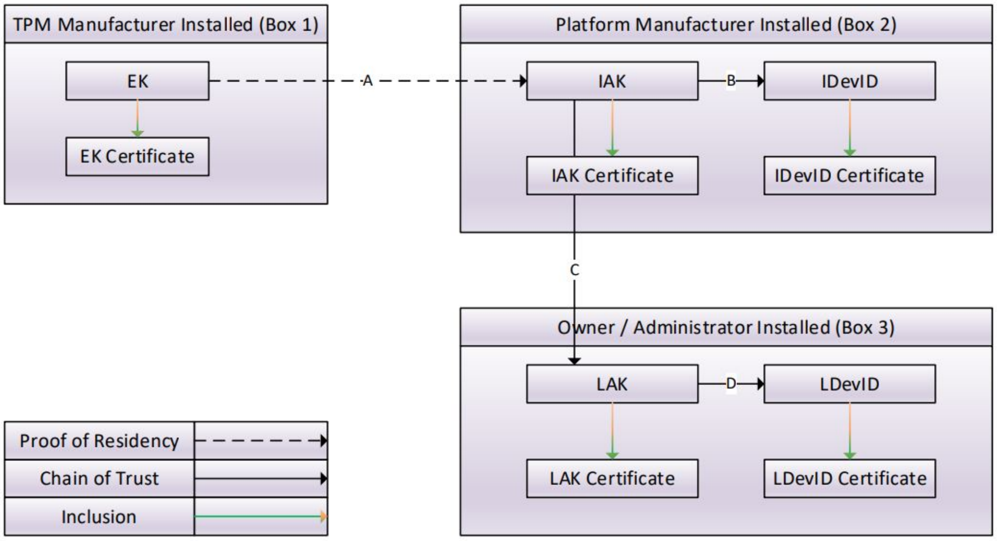
Local Keys (LAK/LDevID)

Created by owner
Should be Ordinary keys
Recommended to be used only for one application

Note regarding Endorsement Key (EK)

Cannot be used as a secure device identifier because it is a storage key (not a signing key) and identifies a TPM (not a device)

Relationships of Keys/Certificates



Box 1: The EK Certificate is signed by the TPM Manufacturer's CA and binds the EK to a specific TPM

Line A: The IAK is verified by the OEM's CA to have the correct key properties and to be resident in the same TPM as the EK

Line B: The IDevID is verified by the OEM's CA to have the correct key properties and to be resident in the same TPM as the IAK

Box 2: The IAK Certificate and IDevID Certificate is signed by the OEM's CA and binds the IAK and IDevID to a specific device

Line C: The LAK is verified by the Owner's CA to have the correct key properties and to be resident in the same TPM as the IAK

Line D: The LDevID is verified by the Owner's CA to have the correct key properties and to be resident in the same TPM as the LAK

Box 3: The LAK Certificate and LDevID Certificate is signed by the Owner's CA

Certificate Authority

General Requirements

Must verify TPM residency of a key

Should support a standard certificate tranport protocol that provides protection from replay attacks and provides confidentiality and integrity (e.g., Enrollment over Secure Transport (EST))

OEM Creation of IAK Certificate based on EK Certificate

Procedure assures that the new IAK is resident in the same TPM as the EK and that the EK resident in this TPM corresponds to the EK certificate

Detailed Procedure

0. The OEM creates and loads the IAK (a FixedTPM Restricted Signing key)
1. The OEM builds the certificate signing request (i.e., the TCG-CSR-IDEVID structure) containing
   (a) Platform identity information (includes the device model and serial number)
   (b) The EK certificate
   (c) The IAK public area
2. The OEM uses TPM2_Hash on the CSR so that the IAK can sign it (since IAK is Restricted)
3. The OEM uses TPM2_Sign on the hash of the CSR using the IAK (proves control of the IAK to the CA)
4. The OEM sends the CSR and the signed hash of the CSR to the CA

5. The CA verifies the received data by
   (a) Checking that the hash matches the CSR
   (b) Checking the signature on the hash of the CSR using the IAK public key (extracted from the CSR)
   (c) Checking the signature on the the EK certificate using the TPM manufacturer's CA public key
   (d) Verifying the attributes of the IAK

6. The CA uses TPM2_Hash on the IAK public area to calculate the cryptographic name of the IAK

7. The CA uses TPM2_GetRandom to generate a nonce

8. The CA uses TPM2_MakeCredential to produce an encrypted credential blob containing
   (a) The cryptographic name of the IAK
   (b) Secret nonce
   (c) Encrypted with the EK

9. The CA sends the encrypted credential blob to the OEM

10. The OEM uses TPM2_ActivateCredential command to release the nonce by
    (a) Decrypting the credential blob using the EK (proves the EK is loaded on the TPM)
    (b) Verifying the name of the IAK (proves the IAK private part is loaded on the same TPM)

11. The OEM returns the nonce to the CA

12. The CA verifies the nonces match

13. The CA issues the IAK certificate

Symbolic Representation

0. OEM executes TPM2_CreatePrimary to get IAK, IAK^{-1}

1. OEM makes TCG_CSR_IDevID containing
   (a) deviceInfo = (prodModel, prodSerial)
   (b) $\text{cert}_{\text{EK}} = [(\text{EK}, \text{tpmInfo})]_{\text{TM\_CA}^{-1}}$
   (c) IAK

2. OEM executes TPM2_Hash to get #CSR

3. OEM executes TPM2_Sign to get $[\#\mathrm{CSR}]_{\mathrm{IAK}^{-1}}$
4. OEM sends CSR and $[\#\mathrm{CSR}]_{\mathrm{IAK}^{-1}}$ to CA
5. CA verifies CSR and $[\#\mathrm{CSR}]_{\mathrm{IAK}^{-1}}$
    - (a) ChechHash #CSR with CSR
    - (b) CheckSig $[\#\mathrm{CSR}]_{\mathrm{IAK}^{-1}}$ with IAK
        - i. Extract IAK from CSR
    - (c) CheckSig $\mathrm{cert}_{\mathrm{EK}}$ with TPM_CA
    - (d) check attributes of IAK
6. CA executes TPM2_Hash to get #IAK
7. CA executes TPM2_GetRandom to get r
8. CA executes TPM2_MakeCredential to get $\{\mathrm{cred}_{\mathrm{IAK}}\}_{\mathrm{EK}}$ with $\mathrm{cred}_{\mathrm{IAK}}$ containing
    - (a) #IAK
    - (b) r
9. CA sends $\{\mathrm{cred}_{\mathrm{IAK}}\}_{\mathrm{EK}}$ to OEM
10. OEM executes TPM2_ActivateCredential to get r'
    - (a) Decrypt $\{\mathrm{cred}_{\mathrm{IAK}}\}_{\mathrm{EK}}$ with EK^{-1}
    - (b) Check #IAK
11. OEM sends r' to CA
12. CA checks r' = r
13. CA sends $\mathrm{cert}_{\mathrm{IAK}} = [(\mathrm{IAK}, \mathrm{deviceInfo})]_{\mathrm{OEM\_CA}^{-1}}$ to OEM

Owner Creation of LAK Certificate based on IAK Certificate

Procedure assures that the new LAK is resident in the same TPM as the IAK

Detailed Procedure

0. The Owner creates and loads the LAK (a FixedTPM Restricted Signing key)
1. The Owner uses TPM2_Certify to produce a signed TPM2B_Attest structure (proves the LAK is loaded on the same TPM as the IAK)

2. The Owner builds the certificate signing request (i.e., the TCG-CSR-LDEVID structure) containing
   (a) The signed TPM2B_Attest structure
   (b) The IAK certificate
3. The Owner uses TPM2_Hash on the CSR so that the LAK can sign it (since LAK is Restricted)
4. The Owner uses TPM2_Sign on the hash of the CSR using the LAK (proves control of the LAK to the CA)
5. Owner sends the CSR and the signed hash of the CSR to the CA
6. The CA verifies the received data by
   (a) Checking that the hash matches the CSR
   (b) Checking the signature on the hash of the CSR using the LAK public key (extracted from the TPM2B_Attest structure)
   (c) Checking the signature on the TPM2B_Attest structure using the IAK public key (extracted from the IAK certificate)
   (d) Checking the signature on the IAK certificate using the OEM's CA public key
   (e) Verify the attributes of the LAK
7. The CA issues the LAK certificate

Symbolic Representation

0. Owner executes TPM2_Create to get LAK, LAK^{-1}
1. Owner executes TPM2_Certify to get $[\text{TPM2B\_Attest}_{\text{LAK}}]_{\text{IAK}^{-1}}$
2. Owner makes TCG_CSR_LDevID containing
   (a) $[\text{TPM2B\_Attest}_{\text{LAK}}]_{\text{IAK}^{-1}}$
   (b) $\text{cert}_{\text{IAK}} = [(\text{IAK}, \text{deviceInfo})]_{\text{OEM\_CA}^{-1}}$
3. Owner executes TPM2_Hash to get #CSR
4. Owner executes TPM2_Sign to get $[\#\text{CSR}]_{\text{LAK}^{-1}}$
5. Owner sends CSR and $[\#\text{CSR}]_{\text{LAK}^{-1}}$ to CA
6. CA verifies CSR and $[\#\text{CSR}]_{\text{LAK}^{-1}}$
   (a) CheckHash #CSR with CSR
   (b) CheckSig $[\#\text{CSR}]_{\text{LAK}^{-1}}$ with LAK

        i. Extract LAK from $[\text{TPM2B\_Attest}_{\text{LAK}}]_{\text{IAK}^{-1}}$ in CSR

   (c) CheckSig $[\text{TPM2B\_Attest}_{\text{LAK}}]_{\text{IAK}^{-1}}$ with IAK

        i. Extract IAK from cert_{IAK} in CSR

   (d) CheckSig cert_{IAK} with OEM_CA

   (e) check attributes of LAK

7. The CA sends $\text{cert}_{\text{LAK}} = [(\text{LAK}, \text{deviceInfo})]_{\text{Owner\_CA}^{-1}}$ to Owner

Related Works

TPM 2.0 Keys for Device Identity and Attestation [2]
Trusted Platform Module Library Specification, Family 2.0 [3]
Formal Analysis of Protocols Based on TPM State Registers [1]
Automated Proof for Authorization Protocols of TPM 2.0 in Computational Model [4]

References

[1] Stephanie Delaune et al. "Formal Analysis of Protocols Based on TPM State Registers." In: *IEEE 24th Computer Security Foundations Symposium*. June 2011, pp. 66–80. DOI: 10.1109/CSF.2011.12.

[2] Trusted Computing Group. *TPM 2.0 Keys for Device Identity and Attestation*. Version 1.12. October 2021.

[3] Trusted Computing Group. *Trusted Platform Module Library Specification, Family 2.0*. Version 1.59. November 2019.

[4] Weijin Wang et al. "Automated Proof for Authorization Protocols of TPM 2.0 in Computational Model." In: *Information Security Practice and Experience*. Ed. by Xinyi Huang and Jianying Zhou. Springer International Publishing, May 2014, pp. 144–158. ISBN: 978-3-319-06320-1. DOI: 10.1007/978-3-319-06320-1_12.