TODO

Sarah Lavinia Johnson

B.S. in Computer Science, University of Kansas, 2021

B.S. in Mathematics, University of Kansas, 2021

Submitted to the graduate degree program in Department of Computer Science and the Graduate Faculty of the University of Kansas in partial fulfillment of the requirements for the degree of Master of Science in Computer Science.

Perry Alexander, Chairperson

Committee members

Michael Branicky

Emily Witt

Date defended: TODO

The Thesis Committee for Sarah Lavinia Johnson certifies
that this is the approved version of the following thesis:

TODO

Perry Alexander, Chairperson

Date approved: TBD

Abstract

Abstract

Acknowledgements

Acknowledgements go here.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Chapter 2

Background

2.1 TPM 2.0

A Trusted Platform Module (TPM) is a microcontroller that complies with the ISO/IEC 11889:2015 international standard. The TPM and its specification was designed by the Trusted Computing Group (TCG) to act as a hardware anchor for PC system security [Arthur et al. (2015)]. To this end, TPMs contain the abilities necessary for secure generation of keys, secure storage of keys, NVRAM storage, algorithm agility, enhanced authorization, device health attestation, device identification, and more.

In order to understand device identification, we must first look at a simplified version of the process for creating and distributing TPMs to the end user. The following capitalized titles of TPM Manufacturer, OEM, and Owner/Administrator are keywords that will be referenced throughout this paper.

1. TPM Manufacturers produce TPM chips according to the international standard.

2. These TPM chips are distributed to the original equipment manufacturers (OEMs).

3. OEMs produce devices (e.g., PCs) with these TPM chips integrated.

4. These devices are distributed to the end users (Owners/Administrators).

Steps 1 and 3 each include a procedure for provisioning keys for identification. In particular, the TPM Manufacturer will supply identification for the TPM in step 1, and the OEM will supply identification for the device in step 3.

2.1.1 Keys

All keys discussed in this document utilize asymmetric cryptography. Although the TPM 2.0 has capabilities for utilizing symmetric keys, it is outside the scope of this paper.

A TPM key may be created using one of two commands.

- `TPM2_CreatePrimary`: A Primary key is produced based on the current Primary Seed. A Primary key may be persisted within the TPM. Otherwise it must be recreated after a TPM reset.

- `TPM2_Create`: An Ordinary key is produced based on a seed taken from the TPM random number generator (RNG). An Ordinary key must be the child of another key. It 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 in the TPM that created it.

Keys have attributes that are set at creation-time. These attributes are permanent and include the following: `FixedTPM`, `Sign`, `Decrypt`, `Restricted`, amongst others. The `FixedTPM` attribute indicates that the private key cannot be duplicated. All keys considered in this paper have this attribute set. A TPM key pair with the `Sign` attribute set consists of a private signing key and a public signature-verification key. When properly handled, private signing keys can provide integrity, authenticity, and nonrepudiation. A TPM key pair with the `Decrypt` attribute set consists of a public encryption key and a private decryption key. When properly handled, public encryption keys can provide confidentiality. A key with both the `Sign` and `Decrypt` attributes set is called a Combined key. US NIST SP800-57 disallows the use of Combined keys for the reason that it may weaken the security guarantees associated with one or both of the attributes. Moreover, a TPM key pair may have the `Restricted` attribute set, limiting the operations of the private key to TPM generated data.

The `Restricted` attribute can provide important security implications. A restricted signing key may only sign a digest that has been produced by the TPM. Enforcement of this constraint is

reliant on a 4-byte magic value called `TPM_Generated` [Trusted Computing Group (2019)]. All structures that the TPM constructs from internal data begins with this value. Such structures include keys, platform configuration registers (PCRs), and audit digests. These structures contribute to two primary use cases for restricted signing keys: (1) key certification and (2) attestation. Use case (1) proves that a new key resides in the same TPM as some known restricted key. Use case (2) utilizes the `Restricted` attribute to provide assurance that a signature over PCRs or audit logs was in fact over a digest generated by that particular TPM. Additionally, a restricted signing key can sign data supplied to the TPM externally by using the `TPM2_Hash` command. In this case the `TPM2_Hash` command produces a ticket asserting that the TPM itself calculated this hash and will later sign it. A restricted signing key will not sign external data without this ticket. To prevent spoofing of the two primary use cases described above, the `TPM2_Hash` command will only produce a ticket if the external data does not begin with the `TPM_Generated` value.

A restricted decryption key is called a storage key. Only storage keys can be used as parents to create or load child objects or to activate credentials [Arthur et al. (2015)]. All TPMs are shipped with an essential storage key: the endorsement key. The endorsement key (EK) is installed by the TPM manufacturer and stored in a shielded location on the TPM. The corresponding EK certificate serves a significant role in the enrollment of secure device identifiers. This process will be discussed in further detail in later sections.

A certificate contains an identity and a public key and is signed by a trusted certificate authority. The term certificate specifically refers to an X.509 v3 digital certificate. The EK certificate includes the public part of the EK itself as well as various assertions regarding the security qualities and provenance of the TPM [Trusted Computing Group (2021a)]. The EK certificate binds a specific TPM to an EK. For keys created by entities other than the TPM manufacturer, a certificate's identity field will contain non-TPM device information. This information should be globally unique per device [Institute of Electrical and Electronics Engineers (2018)].

2.2 Inductive Propositions

[Pierce et al. (2022)]

Chapter 3

Secure Device Identity

A secure device identifier (DevID) is an identifier that is cryptographically bound to a device [Institute of Electrical and Electronics Engineers (2018)]. A device with DevID capability includes an Initial Device Identifier (IDevID) provided by the original equipment manufacturer (OEM). This IDevID must be stored in a way that protects it from modification. Since a TPM has capabilities to protect keys against compromise, it is an ideal choice for IDevID storage. Additionally, a device with DevID capability may support the creation of Locally Significant Device Identifiers (LDevIDs) by a device owner or network administrator. An LDevID cannot be transferred to a device with a different IDevID without knowledge of the private key used to produce the cryptographic binding.

When using a TPM key for secure device identity, there are restrictions on the attributes that it can have in order to enforce the best security practices; the key must have the `FixedTPM` and `Sign` attributes set and the `Decrypt` attribute not set. Furthermore the key may optionally have the `Restricted` attribute set. When the `Restricted` attribute is set, such a key is called an attestation key (AK). This DevID gets its special name due to its unique ability to be used as a parent node in a chain of certificates. The acronym AK is prefixed by the letter I or L denoting Initial or Locally Significant respectively.

The issuers of device identity certificates are known as Certificate Authorities (CAs). CAs are further identified by the creator of the keys that they certify (e.g., the CA that issues certificates for IAKs and IDevIDs is known as the OEM's CA and the CA that issues certificates for LAKs and LDevIDs is known as the Owner/Administrator's CA). The OEM's CA must carefully verify TPM residency and attributes of a key before signing a certificate due to the important security and

Key	Type	FixedTPM	Signing	Decrypting	Restricted	Creator
EK	Primary	X		X	X	TPM Manufacturer
IAK	Primary	X	X		X	OEM
IDevID	Primary	X	X			OEM
LAK	Ordinary	X	X		X	Owner/Admin
LDevID	Ordinary	X	X			Owner/Admin

Table 3.1: Summary of EK, IAK/IDevID, and LAK/LDevID Requirements and Recommendations

identity implications provided by these certificates. All CAs should support a standard certificate transport protocol that provides confidentiality, integrity, and protection from replay attacks. These transport protocols are outside the scope of this paper. We will assume CAs to be following this recommendation precisely.

[Barker (2020)]

3.1 Certificate Chain

A chain of certificates can be used to verify a chain of trust to some trust anchor. The IAK certificate is

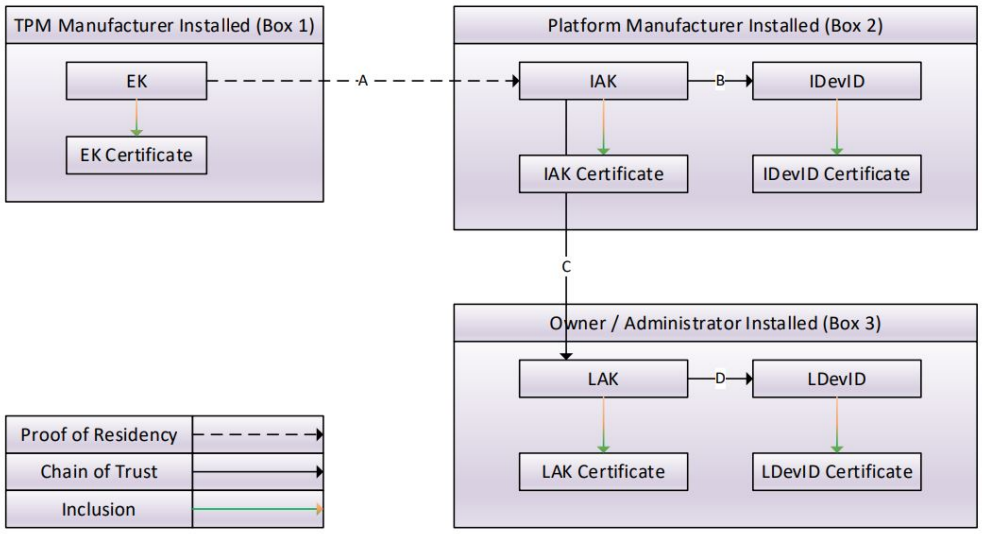

Figure 3.1: Summary of EK, IAK/IDevID, and LAK/LDevID Relationships [Trusted Computing Group (2021b)]

- 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/Administrator'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/Administrator'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/Administrator's CA.

Chapter 4

Identity Provisioning

4.1 Protocols

4.1.1 OEM Creation of IAK Certificate based on EK Certificate

4.1.2 Owner/Administrator Creation of LAK Certificate based on IAK Certificate

4.2 Assurances

Chapter 5

Conclusion

5.1 Conclusion

5.2 Future Work

References

Arthur, W., Challener, D., & Goldman, K. (2015). *A Practical Guide to TPM 2.0: Using the Trusted Platform Module in the New Age of Security*. Apress.

Barker, E. (2020). Recommendation for key management. *NIST Special Publication 800-57 Part 1 Revision 5*.

Institute of Electrical and Electronics Engineers (2018). IEEE Standard for Local and Metropolitan Area Networks - Secure Device Identity. *IEEE Std 802.1AR-2018*.

Pierce, B. C., de Amorim, A. A., Casinghino, C., Gaboardi, M., Greenberg, M., Hriţcu, C., Sjöberg, V., & Yorgey, B. (2022). *Logical Foundations*. Software Foundations series, volume 1. Electronic textbook.

Trusted Computing Group (2019). *Trusted Platform Module Library Specification, Family 2.0, Level 00, Revision 01.59*.

Trusted Computing Group (2021a). *TCG EK Credential Profile For TPM Family 2.0, Level 00, Version 2.4, Revision 3*.

Trusted Computing Group (2021b). *TPM 2.0 Keys for Device Identity and Attestation*.

Appendix A

Misc stuff

Appendix B

Misc Stuff 2