The FutureTPM project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 779391.

FutureTPM H2020 PROJECT: Device Management Use Case

FutureTPM 1st Workshop, 19th October 2018, Lisbon

HWDU

Silviu.Vlasceanu@Huawei.com, Roberto.Sassu@Huawei.com

Outline

- Use Case Overview
- Technology Overview

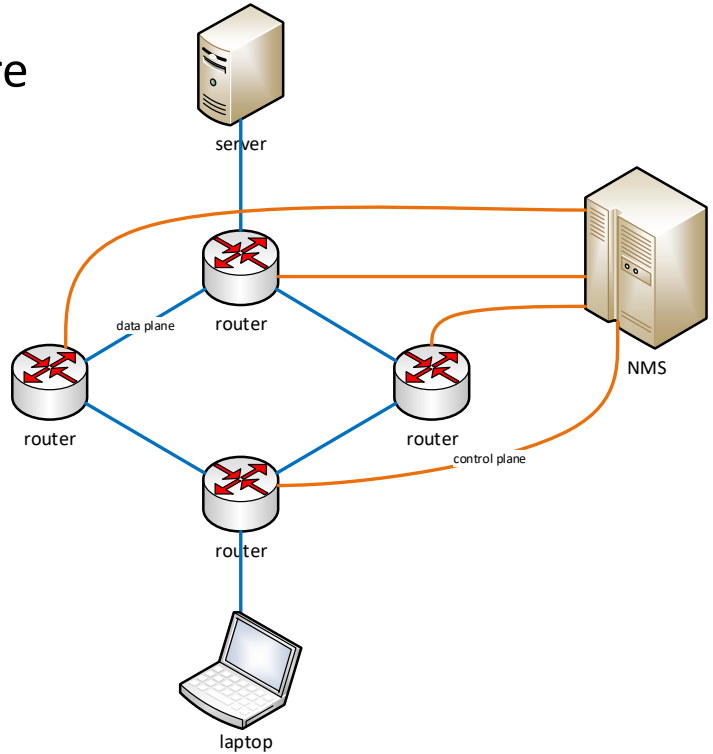
Use Case Overview

Management of enterprise network infrastructure

Infrastructure components

- Network elements (e.g. routers)
- Network Management System (NMS)
- Endpoints (e.g. laptops, servers)

Goal: show security risks and address them with trusted computing



Issues without Trusted Computing

Device identification

- Identity not bound to the hardware (keys can be stolen)

Software integrity

- Routing policy does not depend on routers trust state

Data integrity and confidentiality

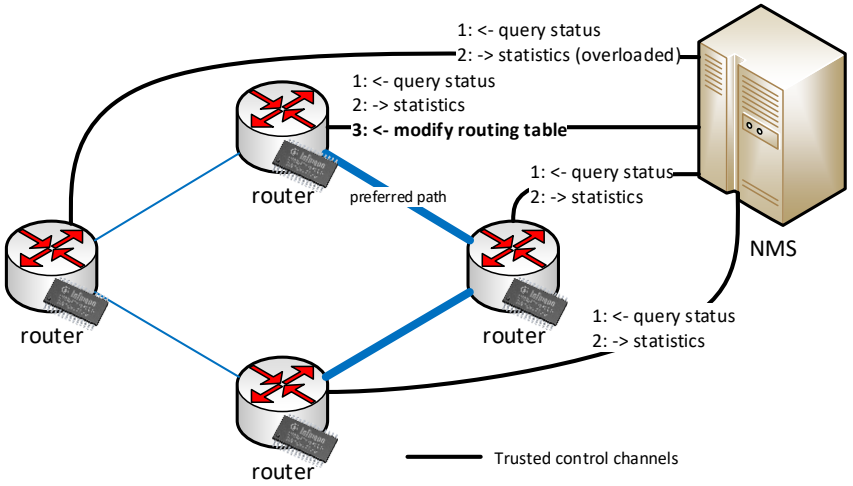
- Data can be accessed even when the device is compromised

Solution with Trusted Computing

Router keys bound to device and firmware/software and protected by the TPM

NMS communicates with routers through trusted channels (e.g. TLS)

Routing policy depends on trust states



Actors (Users)

Network administrator

- Defines trust and routing policies

Network operator

- Deploys the device in the infrastructure

End-user

- Contacts a server in the network infrastructure

Technology Overview

Comprehensive Integrity Verification: our proposal for the protection of the network infrastructure based on trusted computing

- Secure communication between NMS and devices (with TPM keys on the devices)
- Integrity evaluation of the entire OS of network devices

Integrity Problem

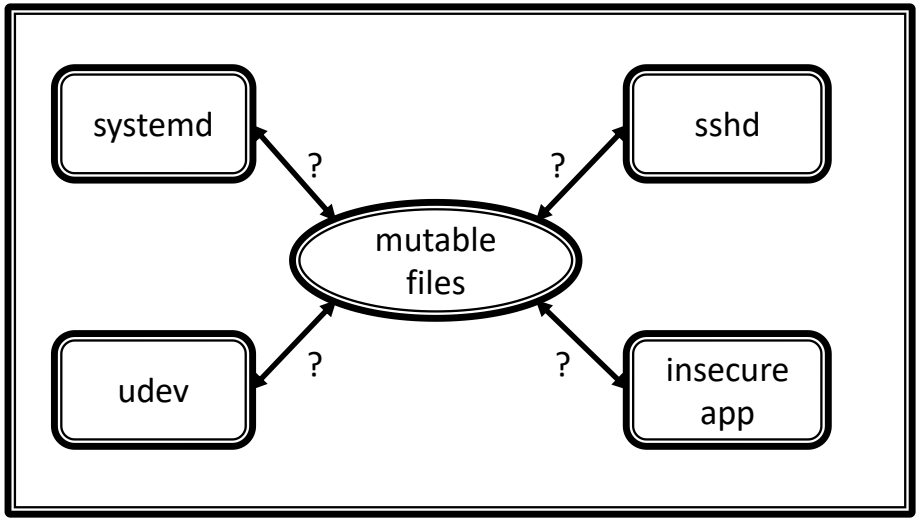
Context: well-defined TCG specifications for verifying components in early stage of boot process

Evaluating the integrity of OS (kernel + applications + their state) is much more complex

- Reference measurements and verification services
- What information must be supplied to verifiers
- How to analyze them

Complexity of the problem limits availability and adoption of TC technologies

Run-time Integrity

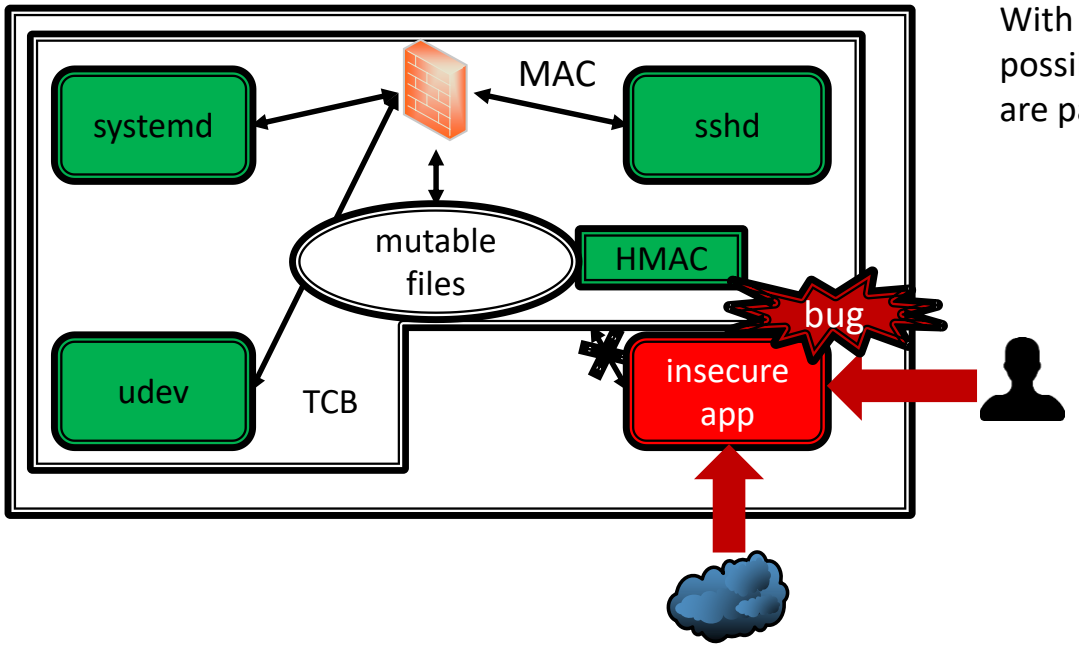


PCR	Digest	Component
10	known	systemd
10	known	udev
10	known	sshd
10	known	insecure app
10	**unknown**	mutable file

IMA measurement list

How to deal with mutable files?

Run-time Integrity – State of Art Solution



With Mandatory Access Control it is possible to define which applications are part of the Trusted Computing Base (TCB)

MAC can enforce an integrity policy, such as Biba or Clark-Wilson

MAC and integrity policy become part of the evidence to be sent to verifiers [1]

[1] Policy-Reduced Integrity Measurement Architecture (PRIMA)
Trent Jaeger, Reiner Sailer, and Umesh Shankar

Comprehensive Integrity Verification

Reduce TCB size, by considering processes interactions discovered on the target system

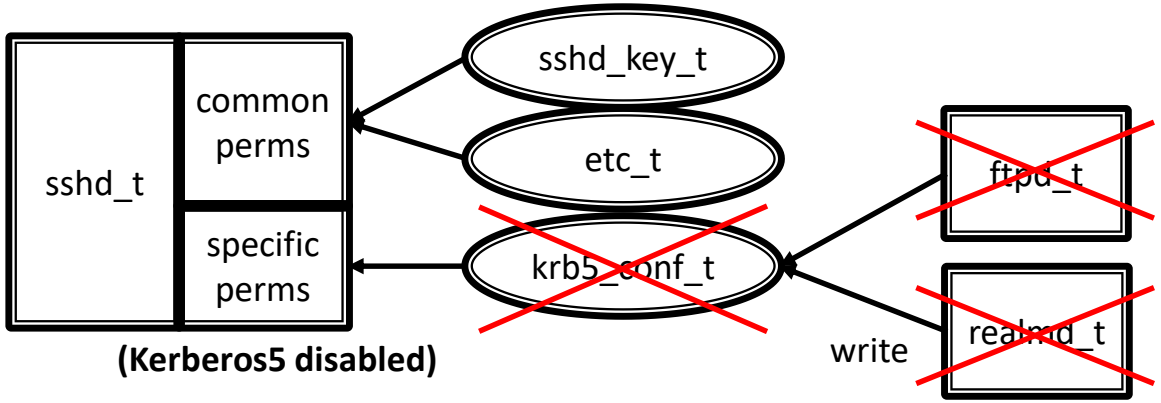
- With a new Linux Security Module (LSM), called Infoflow LSM

Detect malicious updates of mutable files throughout their entire lifetime

Streamline integration of remote attestation in existing infrastructures

Reduce amount of Subjects in the TCB

Example: information flow analysis for sshd (included in the TCB)



With PRIMA, Kerberos5 would be added to the TCB (high risk) or would have to be manually excluded (too much effort)

With our proposal, Kerberos5 is automatically excluded

Permissions taken from SELinux policy of Fedora 27

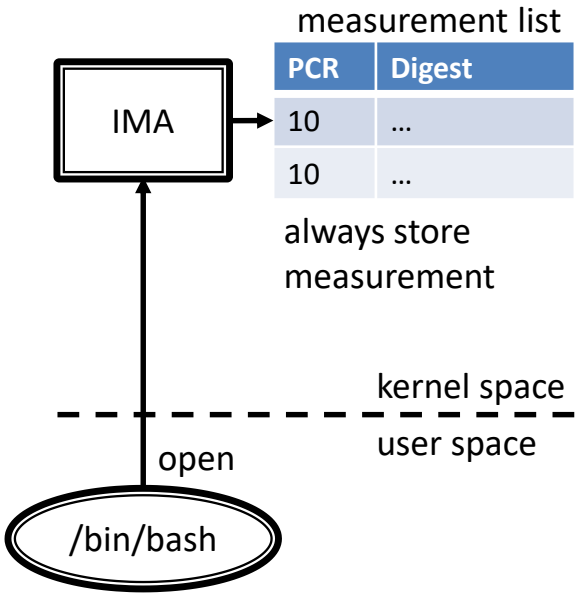
Detect Malicious Updates of Mutable Files

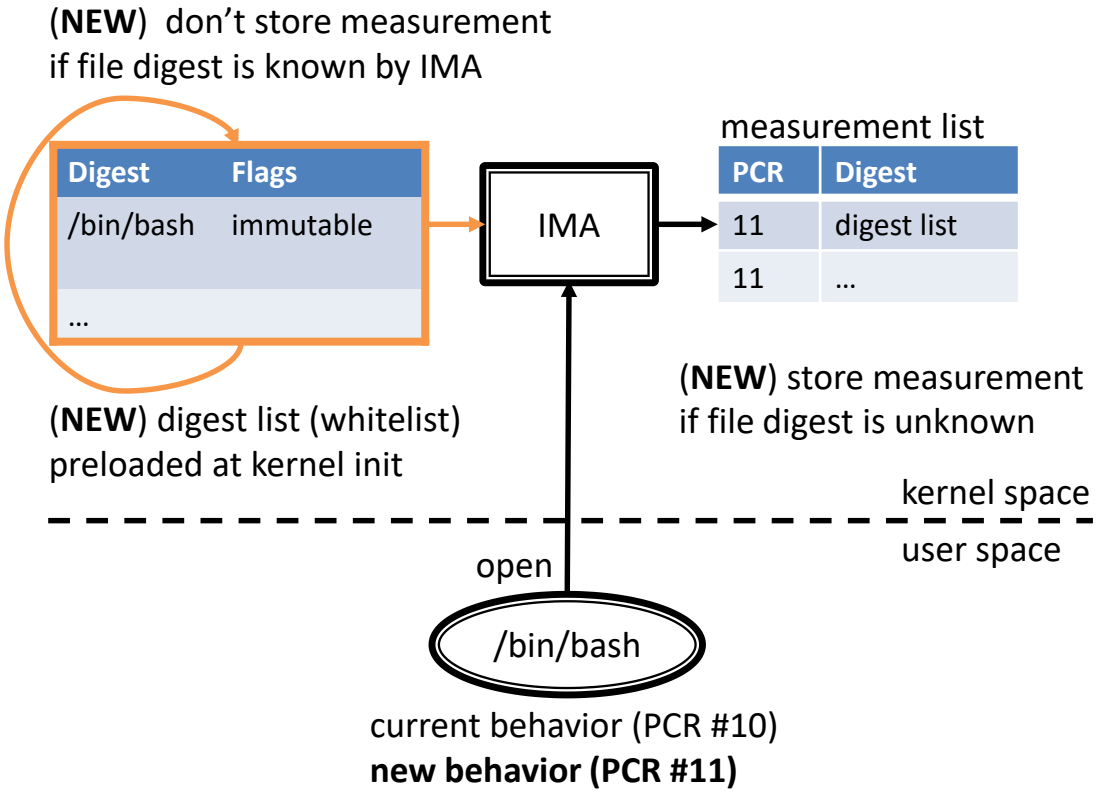
State of the art:

- IMA Appraisal/EVM protect the integrity of data/metadata against offline attacks
- EVM key is sealed with TPM, but not to OS
  - Key can be used with MAC protection disabled
  - A valid HMAC does not imply that the mutable file was updated when the system was good
- IMA PCR not predictable
  - Depends on which and when file are accessed
  - OS integrity cannot be included in sealing policy

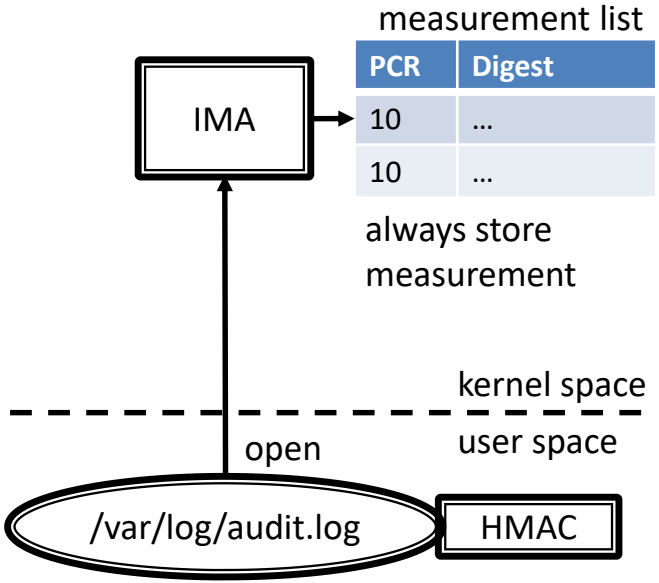
Our IMA Digest Lists extension solves the predictability issue

IMA Digest Lists

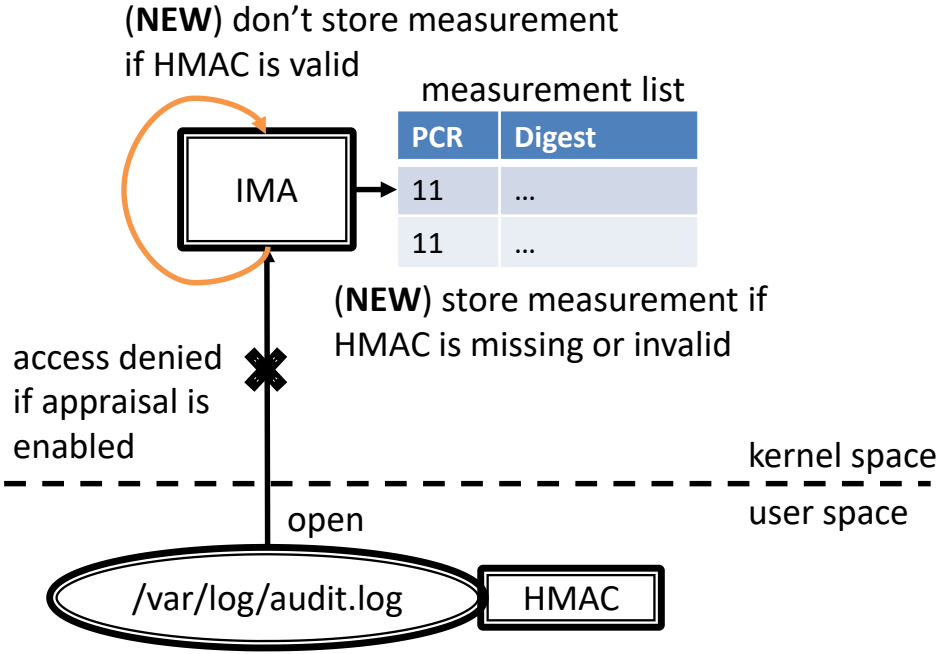




Handling of Mutable Files

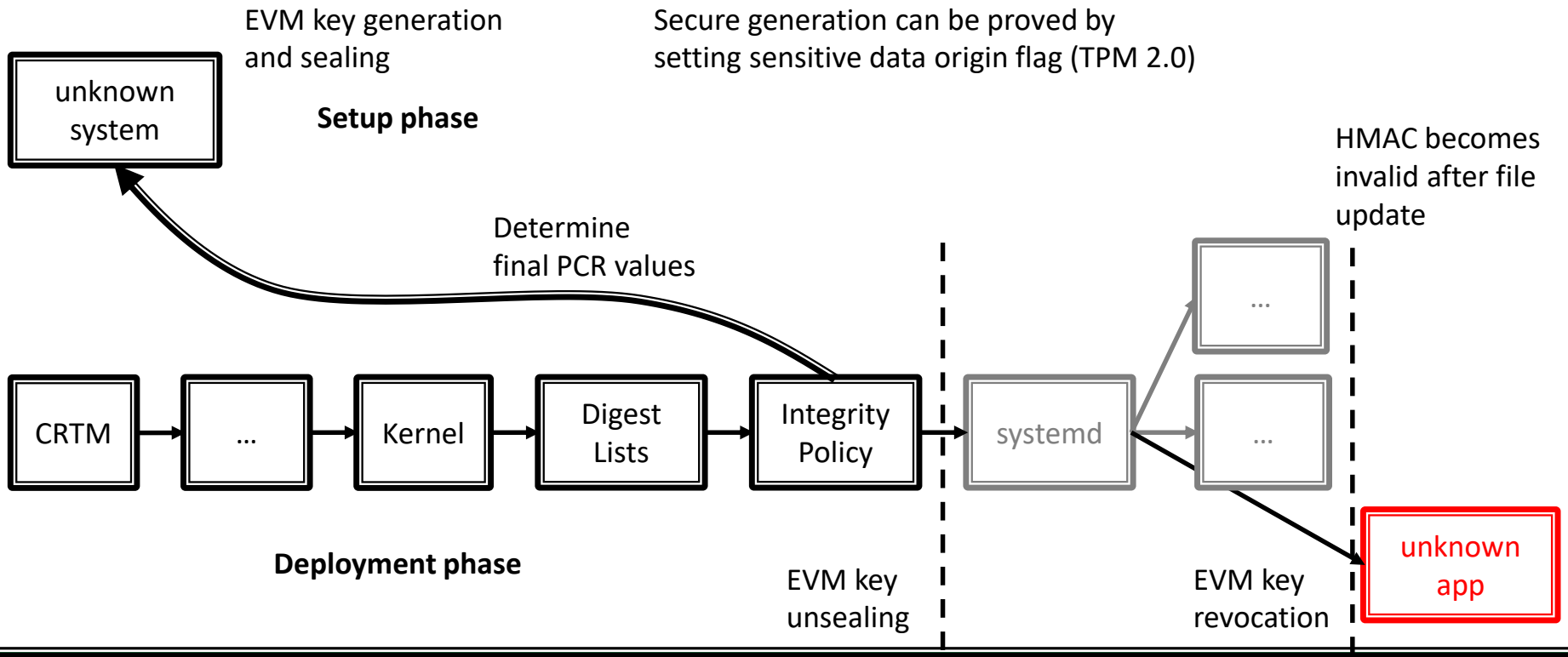


current behavior (PCR #10)

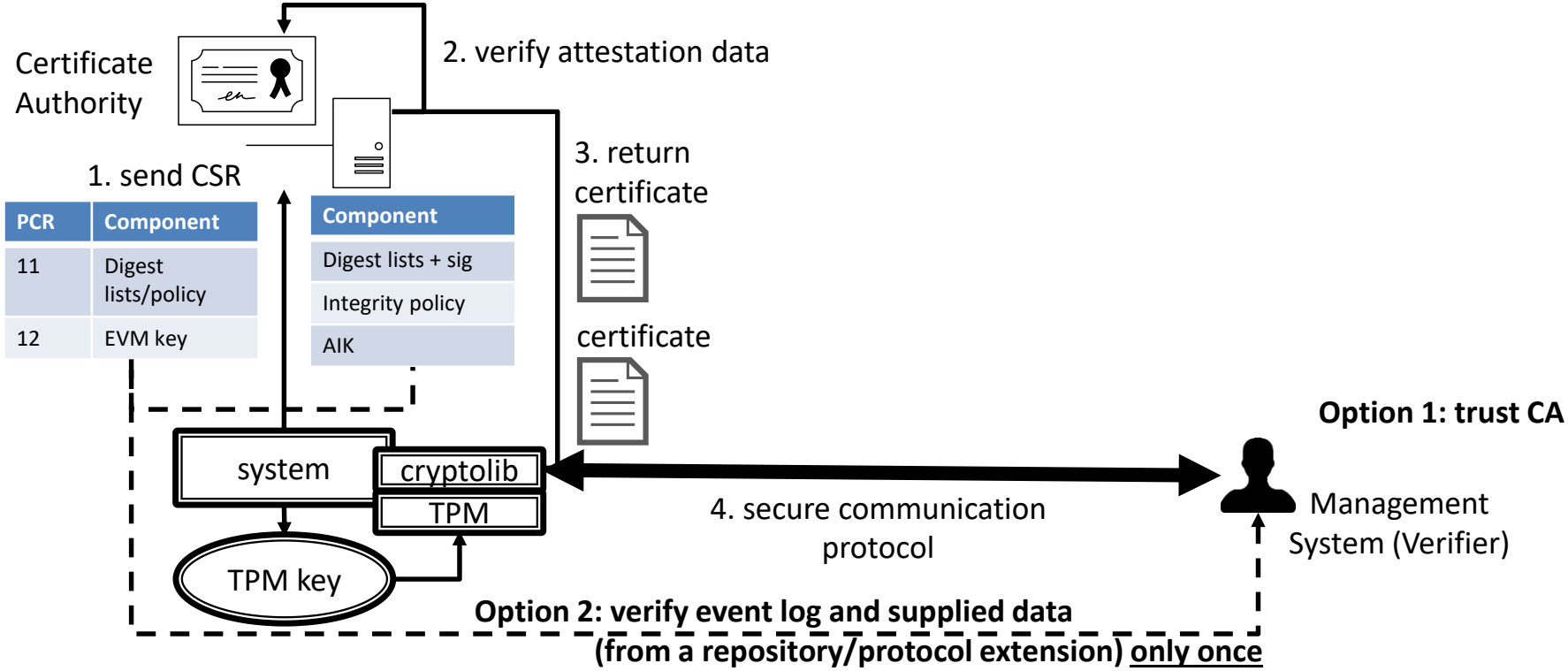


current behavior (PCR #10)
new behavior (PCR #11)

Lifecycle of a EVM Key Sealed to OS



Implicit Remote Attestation – Verification Options



Source Code

Digest lists source code

- Kernel space: https://github.com/euleros/linux (tag: ima-digest-lists-v3)
- User space: https://github.com/euleros/digest-list-tools (tag: v0.2)

Binary packages for Fedora 27, openSUSE Leap 42.3

- Wiki: https://github.com/euleros/digest-list-tools/wiki

Digest lists overview

- https://develop.trustedcomputinggroup.org/2018/05/30/digest-lists-extension-for-linux-ima/

Conclusions

Existing TCG techniques are not practical enough to evaluate OS integrity

- Parallel execution, mutable files are the main obstacles
- Mandatory Access Control is necessary to reduce the code to be trusted

Identifying a TCB of the operating system is a complex problem

- General purpose OSes are prioritizing backwards compatibility over integrity
- Integrity models are often violated (e.g. ssh server reads data from the network)
- System designers' task is to determine whether a subject or object should be added to the TCB

Our solution aims to increase the adoption of TC technologies

- By providing a more comprehensive integrity verification, first on a system with more strict assumptions on usability
- By lowering the requirements for integration with existing products (e.g. Network Management Systems)

FutureTPM Grant Agreement No. 779391

“The FutureTPM project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 779391.”

If you need further information, please contact the coordinator:

TECHNIKON Forschungs- und Planungsgesellschaft mbH
Burgplatz 3a, 9500 Villach, AUSTRIA
Tel: +43 4242 233 55 Fax: +43 4242 233 55 77
E-Mail: coordination@futuretpm.eu