Proposal: Addressing 3 Key Problems of Attestation in Confidential Computing

Muhammad Usama Sardar

Ack: Nikolaus Thümmel
Funding: CPEC

Chair of Systems Engineering
Technische Universität Dresden

CCC Attestation SIG

September 27, 2022

Outline

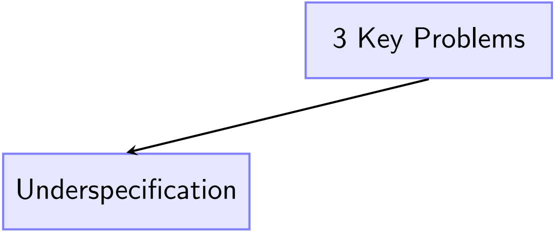
1. Motivation
   - Example 1
   - Example 2
   - Example 3

2. Goals

3. Related Work
   - RATS
   - Neimi et al.

4. Approach

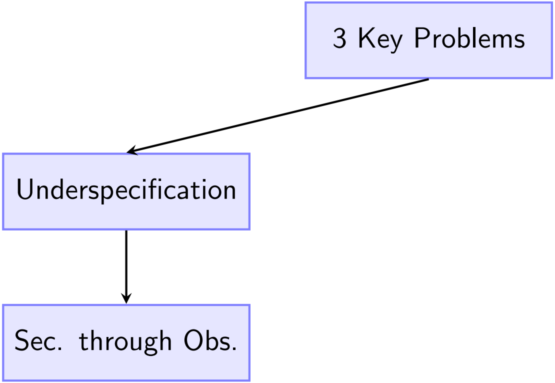
Motivation



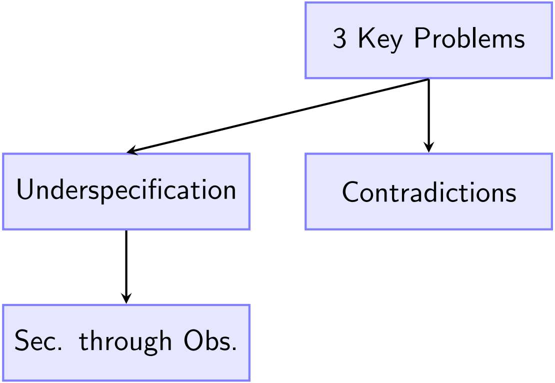
3 Key Problems

Underspecification

Motivation



Motivation



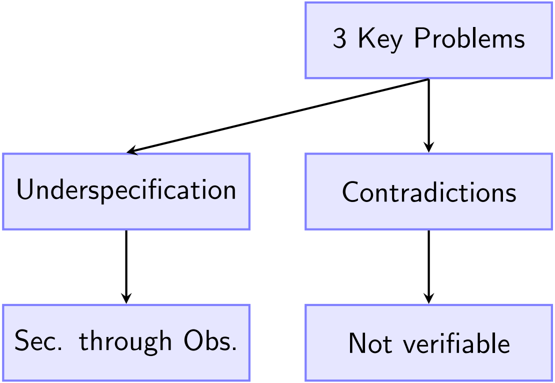
3 Key Problems

Underspecification

Contradictions

Sec. through Obs.

Motivation


3 Key Problems

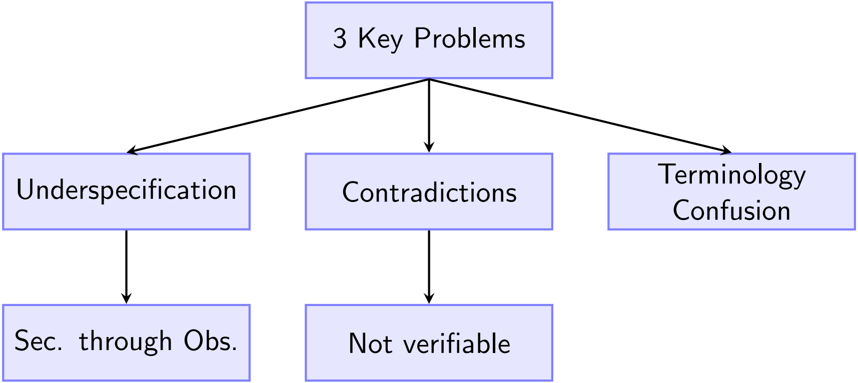
Underspecification

Contradictions

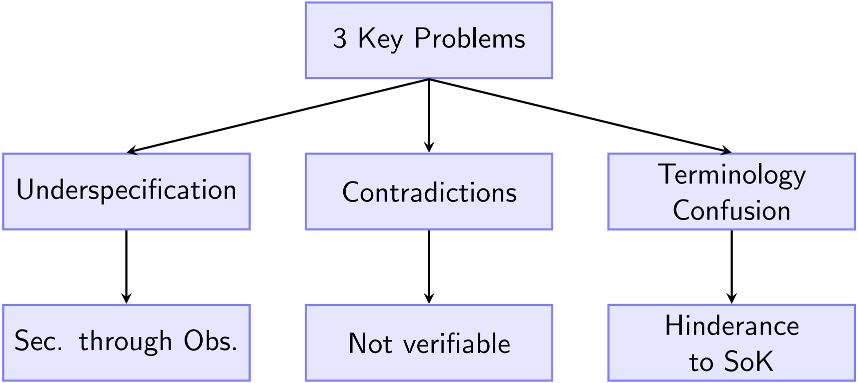
Sec. through Obs.

Not verifiable

Motivation


3 Key Problems
Underspecification
Contradictions
Terminology Confusion
Sec. through Obs.
Not verifiable

Motivation



3 Key Problems

Underspecification

Contradictions

Terminology Confusion

Sec. through Obs.

Not verifiable

Hinderance to SoK

Ex.1 Underspecification and Contradictions[1]

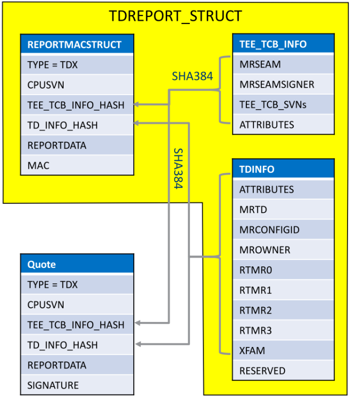
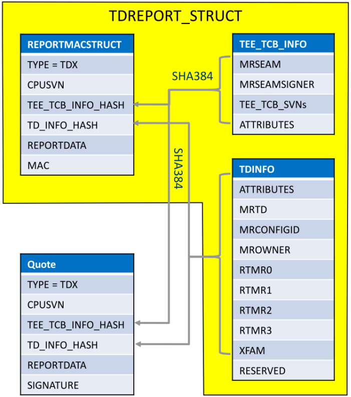


Figure 10.1: TDX Measurement Reporting

- TEE_TCB_INFO hash

[1] Intel, *Architecture Specification: Intel® Trust Domain Extensions (Intel® TDX) Module*, 2020

Ex.1 Underspecification and Contradictions[1]



TDREPORT_STRUCT

REPORTMACSTRUCT
- TYPE = TDX
- CPUSVN
- TEE_TCB_INFO_HASH
- TD_INFO_HASH
- REPORTDATA
- MAC

TEE_TCB_INFO
- MRSEAM
- MRSEAMSIGNER
- TEE_TCB_SVNs
- ATTRIBUTES

TDINFO
- ATTRIBUTES
- MRTD
- MRCONFIGID
- MROWNER
- RTMR0
- RTMR1
- RTMR2
- RTMR3
- XFAM
- RESERVED

Quote
- TYPE = TDX
- CPUSVN
- TEE_TCB_INFO_HASH
- TD_INFO_HASH
- REPORTDATA
- SIGNATURE

SHA384

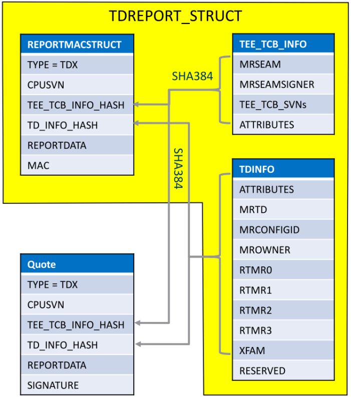
SHA384

Figure 10.1: TDX Measurement Reporting

- TEE_TCB_INFO hash
- TDINFO hash

[1] Intel, *Architecture Specification: Intel® Trust Domain Extensions (Intel® TDX) Module*, 2020

Ex.1 Underspecification and Contradictions[1]



TDREPORT_STRUCT

REPORTMACSTRUCT
TYPE = TDX
CPUSVN
TEE_TCB_INFO_HASH
TD_INFO_HASH
REPORTDATA
MAC

TEE_TCB_INFO
MRSEAM
MRSEAMSIGNER
TEE_TCB_SVNs
ATTRIBUTES

SHA384

SHA384

TDINFO
ATTRIBUTES
MRTD
MRCONFIGID
MROWNER
RTMR0
RTMR1
RTMR2
RTMR3
XFAM
RESERVED

Quote
TYPE = TDX
CPUSVN
TEE_TCB_INFO_HASH
TD_INFO_HASH
REPORTDATA
SIGNATURE

Figure 10.1: TDX Measurement Reporting

- TEE_TCB_INFO hash
- TDINFO hash
- Only description of Quote

[1] Intel, *Architecture Specification: Intel® Trust Domain Extensions (Intel® TDX) Module*, 2020

Ex.1 Underspecification and Contradictions[2]

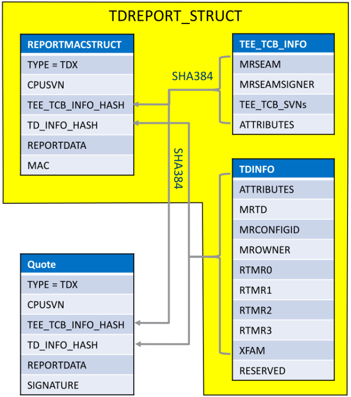


Figure 10.1: TDX Measurement Reporting

- TEE_TCB_INFO hash
- TDINFO hash
- Only description of Quote

- Issues reported to Intel in April 2021[1]

[1] Sardar, Musaev, and Fetzer, "Demystifying Attestation in Intel Trust Domain Extensions via Formal Verification", 2021

[2] Intel, *Architecture Specification: Intel® Trust Domain Extensions (Intel® TDX) Module*, 2020

Ex.1 Underspecification and Contradictions[2]

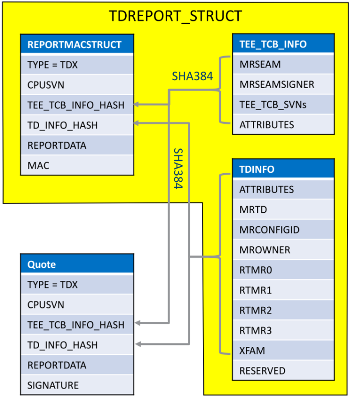


Figure 10.1: TDX Measurement Reporting

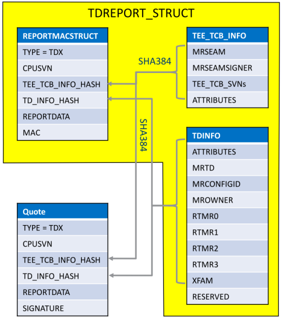
- TEE_TCB_INFO hash
- TDINFO hash
- Only description of Quote

- Issues reported to Intel in April 2021[1]
- Intel acknowledged and promised to update specifications

[1] Sardar, Musaev, and Fetzer, "Demystifying Attestation in Intel Trust Domain Extensions via Formal Verification", 2021

[2] Intel, *Architecture Specification: Intel® Trust Domain Extensions (Intel® TDX) Module*, 2020

Comparison



TDREPORT_STRUCT

REPORTMACSTRUCT
TYPE = TDX
CPUSVN
TEE_TCB_INFO_HASH
TD_INFO_HASH
REPORTDATA
MAC

TEE_TCB_INFO
MRSEAM
MRSEAMSIGNER
TEE_TCB_SVNs
ATTRIBUTES

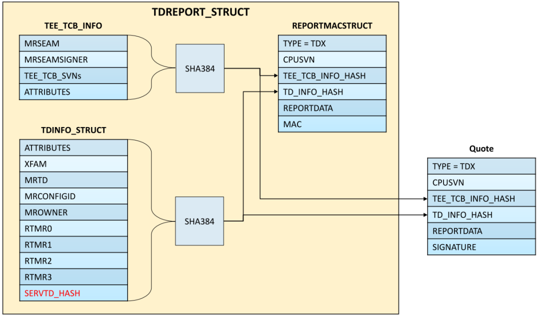
SHA384

TDINFO
ATTRIBUTES
MRTD
MRCONFIGID
MROWNER
RTMR0
RTMR1
RTMR2
RTMR3
XFAM
RESERVED

SHA384

Quote
TYPE = TDX
CPUSVN
TEE_TCB_INFO_HASH
TD_INFO_HASH
REPORTDATA
SIGNATURE

Figure 10.1: TDX Measurement Reporting



TDREPORT_STRUCT

TEE_TCB_INFO
MRSEAM
MRSEAMSIGNER
TEE_TCB_SVNs
ATTRIBUTES

SHA384

REPORTMACSTRUCT
TYPE = TDX
CPUSVN
TEE_TCB_INFO_HASH
TD_INFO_HASH
REPORTDATA
MAC

TDINFO_STRUCT
ATTRIBUTES
XFAM
MRTD
MRCONFIGID
MROWNER
RTMR0
RTMR1
RTMR2
RTMR3
SERVTD_HASH

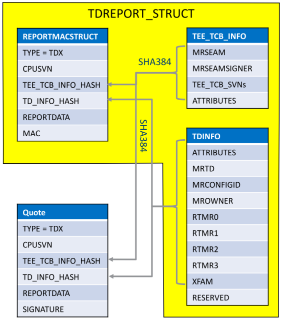
SHA384

Quote
TYPE = TDX
CPUSVN
TEE_TCB_INFO_HASH
TD_INFO_HASH
REPORTDATA
SIGNATURE

Figure 12.1: UPDATED: TD Measurement Reporting

- TEE_TCB_INFO still same

Comparison



TDREPORT_STRUCT

REPORTMACSTRUCT: TYPE = TDX, CPUSVN, TEE_TCB_INFO_HASH, TD_INFO_HASH, REPORTDATA, MAC

TEE_TCB_INFO: MRSEAM, MRSEAMSIGNER, TEE_TCB_SVNs, ATTRIBUTES

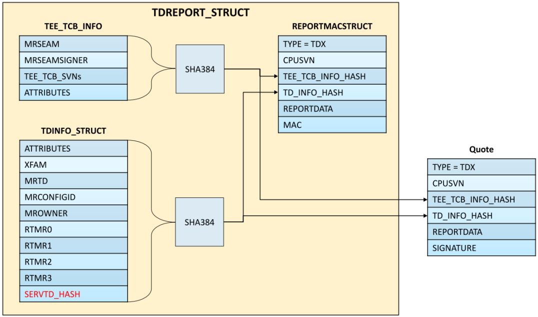
TDINFO: ATTRIBUTES, MRTD, MRCONFIGID, MROWNER, RTMR0, RTMR1, RTMR2, RTMR3, XFAM, RESERVED

Quote: TYPE = TDX, CPUSVN, TEE_TCB_INFO_HASH, TD_INFO_HASH, REPORTDATA, SIGNATURE

SHA384

SHA384

Figure 10.1: TDX Measurement Reporting



TDREPORT_STRUCT

TEE_TCB_INFO: MRSEAM, MRSEAMSIGNER, TEE_TCB_SVNs, ATTRIBUTES

REPORTMACSTRUCT: TYPE = TDX, CPUSVN, TEE_TCB_INFO_HASH, TD_INFO_HASH, REPORTDATA, MAC

TDINFO_STRUCT: ATTRIBUTES, XFAM, MRTD, MRCONFIGID, MROWNER, RTMR0, RTMR1, RTMR2, RTMR3, SERVTD_HASH

SHA384

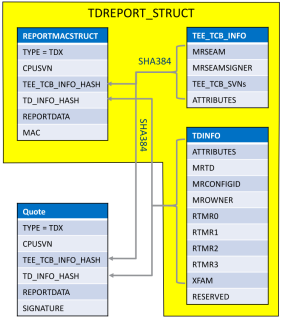
SHA384

Quote: TYPE = TDX, CPUSVN, TEE_TCB_INFO_HASH, TD_INFO_HASH, REPORTDATA, SIGNATURE

Figure 12.1: UPDATED: TD Measurement Reporting

- TEE_TCB_INFO still same
- Quote still same

Comparison



TDREPORT_STRUCT

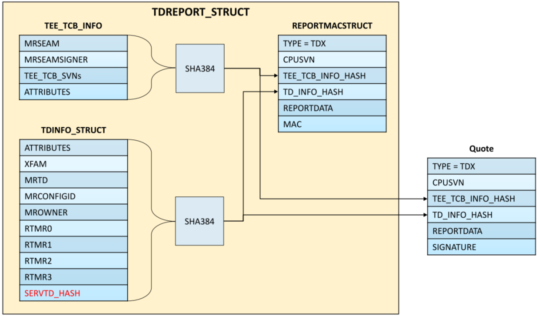
REPORTMACSTRUCT
TYPE = TDX
CPUSVN
TEE_TCB_INFO_HASH
TD_INFO_HASH
REPORTDATA
MAC

TEE_TCB_INFO
MRSEAM
MRSEAMSIGNER
TEE_TCB_SVNs
ATTRIBUTES

TDINFO
ATTRIBUTES
MRTD
MRCONFIGID
MROWNER
RTMR0
RTMR1
RTMR2
RTMR3
XFAM
RESERVED

Quote
TYPE = TDX
CPUSVN
TEE_TCB_INFO_HASH
TD_INFO_HASH
REPORTDATA
SIGNATURE

Figure 10.1: TDX Measurement Reporting



TDREPORT_STRUCT

TEE_TCB_INFO
MRSEAM
MRSEAMSIGNER
TEE_TCB_SVNs
ATTRIBUTES

REPORTMACSTRUCT
TYPE = TDX
CPUSVN
TEE_TCB_INFO_HASH
TD_INFO_HASH
REPORTDATA
MAC

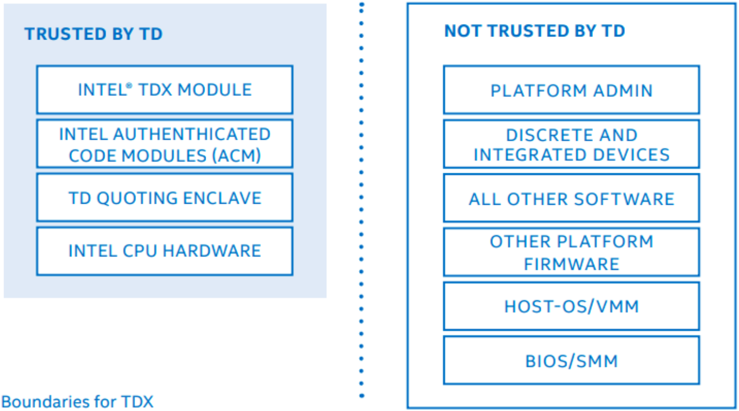
TDINFO_STRUCT
ATTRIBUTES
XFAM
MRTD
MRCONFIGID
MROWNER
RTMR0
RTMR1
RTMR2
RTMR3
SERVTD_HASH

Quote
TYPE = TDX
CPUSVN
TEE_TCB_INFO_HASH
TD_INFO_HASH
REPORTDATA
SIGNATURE

Figure 12.1: UPDATED: TD Measurement Reporting

- TEE_TCB_INFO still same
- Quote still same
- TDINFO still inconsistent

Ex.2 Underspecification[3]



TRUSTED BY TD

- INTEL® TDX MODULE
- INTEL AUTHENTHICATED CODE MODULES (ACM)
- TD QUOTING ENCLAVE
- INTEL CPU HARDWARE

NOT TRUSTED BY TD

- PLATFORM ADMIN
- DISCRETE AND INTEGRATED DEVICES
- ALL OTHER SOFTWARE
- OTHER PLATFORM FIRMWARE
- HOST-OS/VMM
- BIOS/SMM

Figure 5.1. Trust Boundaries for TDX

Unclear how secure channel is established with PCE during provisioning

[3] Intel, *Intel ® Trust Domain Extensions*, 2020

Ex.3 Terminology Confusion

- Payload

Ex.3 Terminology Confusion

- Payload
- Target entity

Ex.3 Terminology Confusion

- Payload
- Target entity
- Prover

Ex.3 Terminology Confusion

- Payload

- Target entity
- Prover
- Attester

Ex.3 Terminology Confusion

- Payload
- Target entity
- Prover
- Attester
- Attestation agent

Ex.3 Terminology Confusion

- Payload
- Target entity
- Prover
- Attester
- Attestation agent
- Attestation service

Ex.3 Terminology Confusion

- Payload

- Target entity
- Prover
- Attester
- Attestation agent
- Attestation service

- Challenger

Ex.3 Terminology Confusion

- Payload

- Target entity
- Prover
- Attester
- Attestation agent
- Attestation service

- Challenger
- Appraiser

Ex.3 Terminology Confusion

- Payload

- Target entity
- Prover
- Attester
- Attestation agent
- Attestation service

- Challenger
- Appraiser
- Relying party

Ex.3 Terminology Confusion

- Payload

- Target entity
- Prover
- Attester
- Attestation agent
- Attestation service

- Challenger
- Appraiser
- Relying party
- Verifier

Outline

1. Motivation
   - Example 1
   - Example 2
   - Example 3
2. Goals
3. Related Work
   - RATS
   - Neimi et al.
4. Approach

Goals

- Attempt consensus in terminology between standardization bodies and academia

Goals

- Attempt consensus in terminology between standardization bodies and academia
- Define concepts formally (or at least precisely) in the context of CC

Goals

- Attempt consensus in terminology between standardization bodies and academia
- Define concepts formally (or at least precisely) in the context of CC
  - Vendor solutions (Intel SGX & TDX, Arm CCA, AMD SEV SNP, IBM PEF, ...)

Goals

- Attempt consensus in terminology between standardization bodies and academia
- Define concepts formally (or at least precisely) in the context of CC
  - Vendor solutions (Intel SGX & TDX, Arm CCA, AMD SEV SNP, IBM PEF, ...)
  - Frameworks on top of vendor solutions (SCONE[4], Gramine, ...)

[4] Arnautov et al., "SCONE: Secure Linux Containers with Intel SGX", 2016

Goals

- Attempt consensus in terminology between standardization bodies and academia
- Define concepts formally (or at least precisely) in the context of CC
  - Vendor solutions (Intel SGX & TDX, Arm CCA, AMD SEV SNP, IBM PEF, ...)
  - Frameworks on top of vendor solutions (SCONE[4], Gramine, ...)
- Benefits

[4] Arnautov et al., "SCONE: Secure Linux Containers with Intel SGX", 2016

Goals

- Attempt consensus in terminology between standardization bodies and academia
- Define concepts formally (or at least precisely) in the context of CC
  - Vendor solutions (Intel SGX & TDX, Arm CCA, AMD SEV SNP, IBM PEF, ...)
  - Frameworks on top of vendor solutions (SCONE[4], Gramine, ...)
- Benefits
  - Architects, developers: better understanding, deal with heterogeneity (Derek Miller)

[4] Arnautov et al., "SCONE: Secure Linux Containers with Intel SGX", 2016

Goals

- Attempt consensus in terminology between standardization bodies and academia
- Define concepts formally (or at least precisely) in the context of CC
  - Vendor solutions (Intel SGX & TDX, Arm CCA, AMD SEV SNP, IBM PEF, ...)
  - Frameworks on top of vendor solutions (SCONE[4], Gramine, ...)
- Benefits
  - Architects, developers: better understanding, deal with heterogeneity (Derek Miller)
  - Vendors: wider use of technology

[4] Arnautov et al., “SCONE: Secure Linux Containers with Intel SGX”, 2016

Goals

- Attempt consensus in terminology between standardization bodies and academia
- Define concepts formally (or at least precisely) in the context of CC
  - Vendor solutions (Intel SGX & TDX, Arm CCA, AMD SEV SNP, IBM PEF, ...)
  - Frameworks on top of vendor solutions (SCONE[4], Gramine, ...)
- Benefits
  - Architects, developers: better understanding, deal with heterogeneity (Derek Miller)
  - Vendors: wider use of technology
  - Researchers: scientific well-founded answers and comparisons, SoK, interdisciplinary opportunities (SE+Sec.+FM),

[4] Arnautov et al., "SCONE: Secure Linux Containers with Intel SGX", 2016

Goals

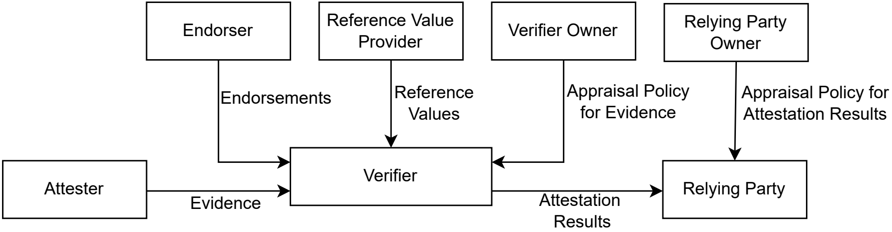
- Attempt consensus in terminology between standardization bodies and academia
- Define concepts formally (or at least precisely) in the context of CC
  - Vendor solutions (Intel SGX & TDX, Arm CCA, AMD SEV SNP, IBM PEF, ...)
  - Frameworks on top of vendor solutions (SCONE[4], Gramine, ...)
- Benefits
  - Architects, developers: better understanding, deal with heterogeneity (Derek Miller)
  - Vendors: wider use of technology
  - Researchers: scientific well-founded answers and comparisons, SoK, interdisciplinary opportunities (SE+Sec.+FM),
  - Legal: compliance with regulations (transparency obligations)

[4] Arnautov et al., "SCONE: Secure Linux Containers with Intel SGX", 2016

Outline

1. Motivation
   - Example 1
   - Example 2
   - Example 3
2. Goals
3. Related Work
   - RATS
   - Neimi et al.
4. Approach

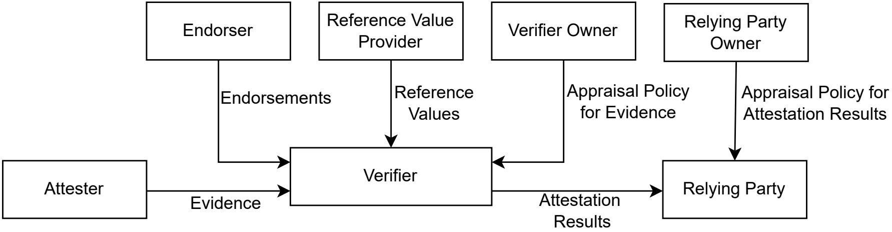
RATS[5]



- Claim: neutral to processor architecture

[5] Birkholz et al., *Remote Attestation Procedures Architecture*, 2022

RATS[6]

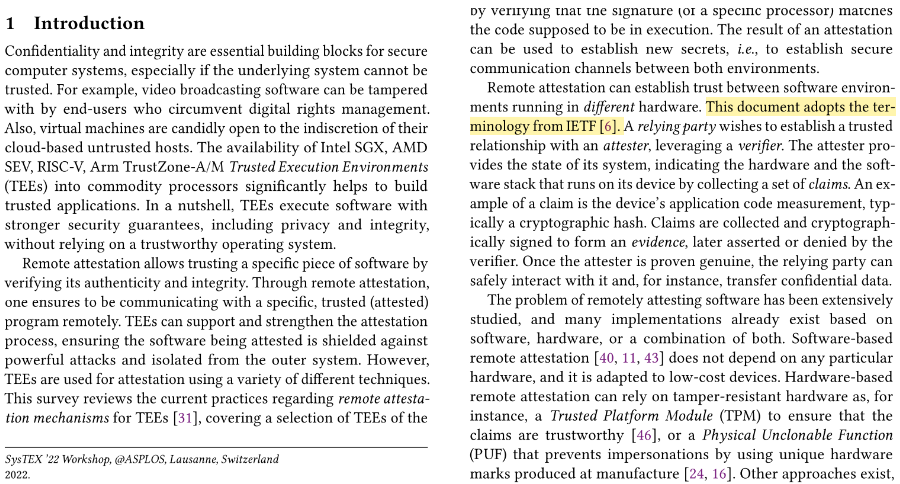


- Claim: neutral to processor architecture
- Terminology, topological patterns, and even architecture primarily ‘taken’ from DICE[5]

[5] Trusted Computing Group, *DICE Attestation Architecture*, 2021

[6] Birkholz et al., *Remote Attestation Procedures Architecture*, 2022

Use of RATS[7]



1 Introduction

Confidentiality and integrity are essential building blocks for secure computer systems, especially if the underlying system cannot be trusted. For example, video broadcasting software can be tampered with by end-users who circumvent digital rights management. Also, virtual machines are candidly open to the indiscretion of their cloud-based untrusted hosts. The availability of Intel SGX, AMD SEV, RISC-V, Arm TrustZone-A/M *Trusted Execution Environments* (TEEs) into commodity processors significantly helps to build trusted applications. In a nutshell, TEEs execute software with stronger security guarantees, including privacy and integrity, without relying on a trustworthy operating system.

Remote attestation allows trusting a specific piece of software by verifying its authenticity and integrity. Through remote attestation, one ensures to be communicating with a specific, trusted (attested) program remotely. TEEs can support and strengthen the attestation process, ensuring the software being attested is shielded against powerful attacks and isolated from the outer system. However, TEEs are used for attestation using a variety of different techniques. This survey reviews the current practices regarding *remote attestation mechanisms* for TEEs [31], covering a selection of TEEs of the

SysTEX '22 Workshop, @ASPLOS, Lausanne, Switzerland
2022.

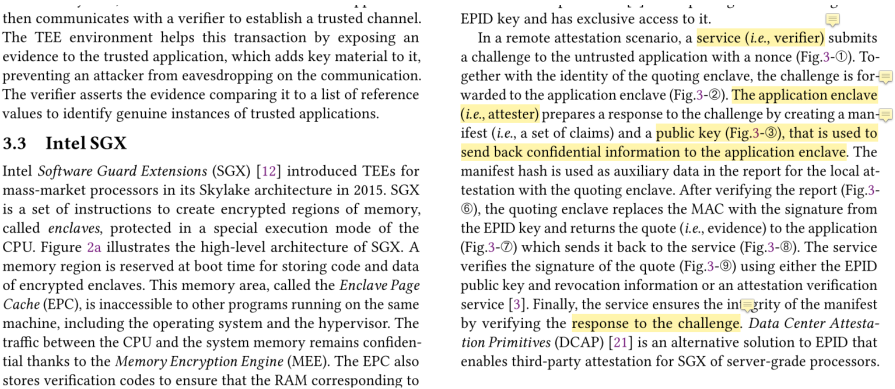
by verifying that the signature (of a specific processor) matches the code supposed to be in execution. The result of an attestation can be used to establish new secrets, *i.e.*, to establish secure communication channels between both environments.

Remote attestation can establish trust between software environments running in *different* hardware. This document adopts the terminology from IETF [6]. A *relying party* wishes to establish a trusted relationship with an *attester*, leveraging a *verifier*. The attester provides the state of its system, indicating the hardware and the software stack that runs on its device by collecting a set of *claims*. An example of a claim is the device's application code measurement, typically a cryptographic hash. Claims are collected and cryptographically signed to form an *evidence*, later asserted or denied by the verifier. Once the attester is proven genuine, the relying party can safely interact with it and, for instance, transfer confidential data.

The problem of remotely attesting software has been extensively studied, and many implementations already exist based on software, hardware, or a combination of both. Software-based remote attestation [40, 11, 43] does not depend on any particular hardware, and it is adapted to low-cost devices. Hardware-based remote attestation can rely on tamper-resistant hardware as, for instance, a *Trusted Platform Module* (TPM) to ensure that the claims are trustworthy [46], or a *Physical Unclonable Function* (PUF) that prevents impersonations by using unique hardware marks produced at manufacture [24, 16]. Other approaches exist,

[7] Ménétrey et al., "An Exploratory Study of Attestation Mechanisms for Trusted Execution Environments", 2022

(Mis-)Use of RATS[8]



then communicates with a verifier to establish a trusted channel. The TEE environment helps this transaction by exposing an evidence to the trusted application, which adds key material to it, preventing an attacker from eavesdropping on the communication. The verifier asserts the evidence comparing it to a list of reference values to identify genuine instances of trusted applications.

3.3 Intel SGX

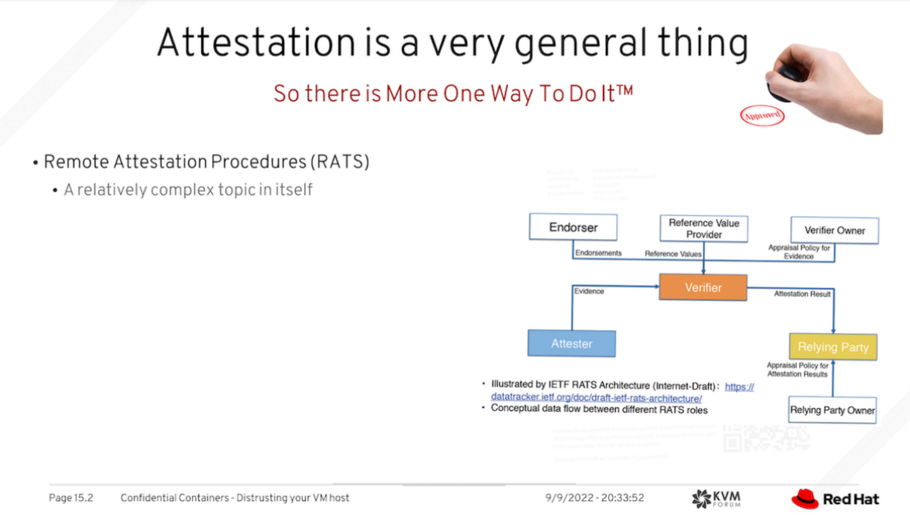
Intel *Software Guard Extensions* (SGX) [12] introduced TEEs for mass-market processors in its Skylake architecture in 2015. SGX is a set of instructions to create encrypted regions of memory, called *enclaves*, protected in a special execution mode of the CPU. Figure 2a illustrates the high-level architecture of SGX. A memory region is reserved at boot time for storing code and data of encrypted enclaves. This memory area, called the *Enclave Page Cache* (EPC), is inaccessible to other programs running on the same machine, including the operating system and the hypervisor. The traffic between the CPU and the system memory remains confidential thanks to the *Memory Encryption Engine* (MEE). The EPC also stores verification codes to ensure that the RAM corresponding to

EPID key and has exclusive access to it.

In a remote attestation scenario, a service (*i.e.*, verifier) submits a challenge to the untrusted application with a nonce (Fig.3-①). Together with the identity of the quoting enclave, the challenge is forwarded to the application enclave (Fig.3-②). The application enclave (*i.e.*, attester) prepares a response to the challenge by creating a manifest (*i.e.*, a set of claims) and a public key (Fig.3-③), that is used to send back confidential information to the application enclave. The manifest hash is used as auxiliary data in the report for the local attestation with the quoting enclave. After verifying the report (Fig.3-⑥), the quoting enclave replaces the MAC with the signature from the EPID key and returns the quote (*i.e.*, evidence) to the application (Fig.3-⑦) which sends it back to the service (Fig.3-⑧). The service verifies the signature of the quote (Fig.3-⑨) using either the EPID public key and revocation information or an attestation verification service [3]. Finally, the service ensures the integrity of the manifest by verifying the response to the challenge. *Data Center Attestation Primitives* (DCAP) [21] is an alternative solution to EPID that enables third-party attestation for SGX of server-grade processors.

[8] Ménétrey et al., "An Exploratory Study of Attestation Mechanisms for Trusted Execution Environments", 2022

Problem: complexity of RATS (slide from Dinechin[9])



Attestation is a very general thing

So there is More One Way To Do It™

- Remote Attestation Procedures (RATS)
  - A relatively complex topic in itself

- Illustrated by IETF RATS Architecture (Internet-Draft): https://datatracker.ietf.org/doc/draft-ietf-rats-architecture/
- Conceptual data flow between different RATS roles

Page 15.2 Confidential Containers - Distrusting your VM host 9/9/2022 - 20:33:52 KVM FORUM Red Hat

[9] Dinechin, *Five Big Problems with Confidential Containers*, 2022

Adapted from RATS[10]

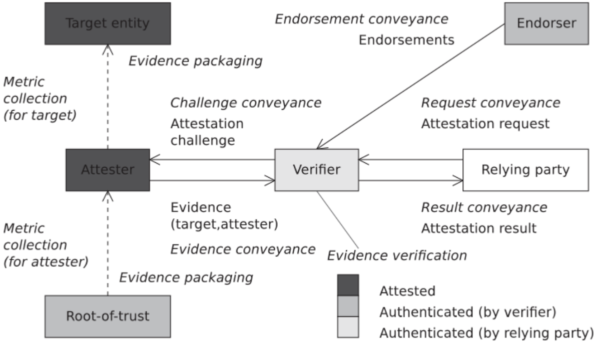


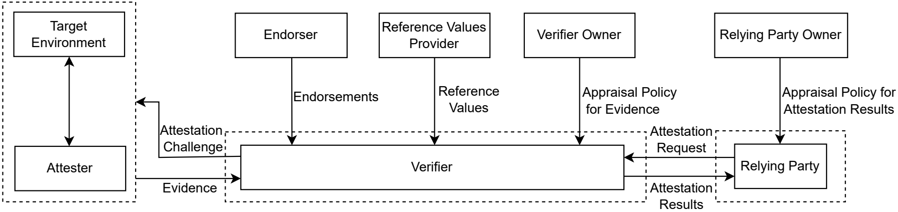
Fig. 1. Roles, messages and *processes* in the architectural model used in this article (adapted from the TCG and RATS architectures [20, p. 12][13, p. 8]. As discussed in the text, many variations are possible.

[10] Niemi, Sovio, and Ekberg, "Towards Interoperable Enclave Attestation: Learnings from Decades of Academic Work", 2022

Outline

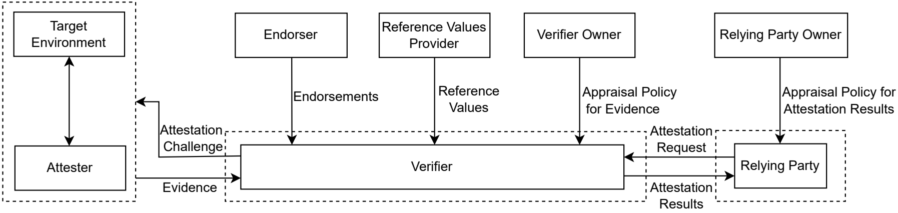
1. Motivation
   - Example 1
   - Example 2
   - Example 3
2. Goals
3. Related Work
   - RATS
   - Neimi et al.
4. Approach

Towards Precise Definitions



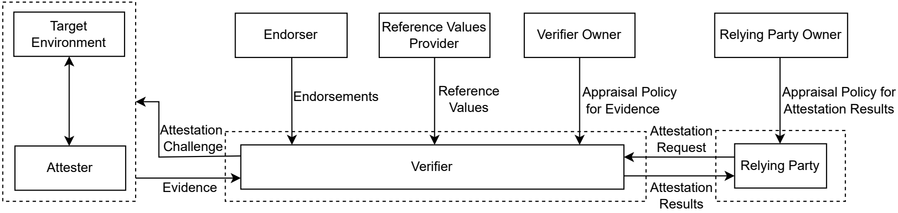
- Attester: minimal number of certifiers

Towards Precise Definitions



- Attester: minimal number of certifiers
- Local vs. Remote: on same/different platform

Towards Precise Definitions



- Attester: minimal number of certifiers
- Local vs. Remote: on same/different platform
- Verifying Relying Party

Key References

Arnautov, Sergei et al. "SCONE: Secure Linux Containers with Intel SGX". In: *USENIX Symposium on Operating Systems Design and Implementation*. 2016, pp. 689–703.

Birkholz, Henk et al. *Remote Attestation Procedures Architecture*. Tech. rep. August. Internet Engineering Task Force, 2022, pp. 1–56. URL: https://datatracker.ietf.org/doc/html/draft-ietf-rats-architecture-21.

Dinechin, Christophe de. *Five Big Problems with Confidential Containers*. Sept. 2022. URL: https://static.sched.com/hosted_files/kvmforum2022/f9/Five%20Big%20Problems%20with%20Confidential%20Containers%20%E2%80%93%C2%A0KVM%20Forum%202022.pdf.

Intel. *Architecture Specification: Intel® Trust Domain Extensions (Intel® TDX) Module*. Sept. 2020. URL: https://software.intel.com/content/dam/develop/external/us/en/documents/intel-tdx-module-1eas.pdf.

— .*Intel ® Trust Domain Extensions*. Aug. 2020. URL: https://cdrdv2.intel.com/v1/dl/getContent/690419.

Ménétrey, Jämes et al. "An Exploratory Study of Attestation Mechanisms for Trusted Execution Environments". In: 2022.

Niemi, Arto, Sampo Sovio, and Jan Erik Ekberg. "Towards Interoperable Enclave Attestation: Learnings from Decades of Academic Work". In: *Conference of Open Innovation Association, FRUCT*. Vol. 2022-April. IEEE Computer Society, 2022, pp. 189–200. ISBN: 9789526924472. DOI: 10.23919/FRUCT54823.2022.9770907. URL: https://ieeexplore.ieee.org/document/9770907.

Sardar, Muhammad Usama, Saidgani Musaev, and Christof Fetzer. "Demystifying Attestation in Intel Trust Domain Extensions via Formal Verification". In: *IEEE Access* (2021). URL: https://www.researchgate.net/publication/351699567_Demystifying_Attestation_in_Intel_Trust_Domain_Extensions_via_Formal_Verification.

Trusted Computing Group. *DICE Attestation Architecture*. Tech. rep. 2021. URL: https://trustedcomputinggroup.org/wp-content/uploads/DICE-Attestation-Architecture-r23-final.pdf%20https://trustedcomputinggroup.org/resource/dice-attestation-architecture/.