SoK: Attestation in Confidential Computing

Muhammad Usama Sardar\textsuperscript{1} \quad Thomas Fossati\textsuperscript{2} \quad Simon Frost\textsuperscript{2}

\textsuperscript{1}TU Dresden
\textsuperscript{2}Arm Ltd.

March 29, 2023
Outline

1. Problem Statement
2. Contributions
3. Summary
Confidential Computing

Relying Party
Confidential Computing

Relying Party \rightarrow \text{Workload}
Confidential Computing

Relying Party \rightarrow Workload

SW
Confidential Computing

Relying Party

Workload

SW

HW and FW
Attestation

Relying Party \rightarrow \text{Attestation request} \rightarrow \text{Attestor}
Attestation

Relying Party

Attestation request

Evidence

Attester
Problem Statement

Holistic view of attestation
Problem Statement

Holistic view of attestation

TEE-agnostic attestation architecture
Problem Statement

- Holistic view of attestation
- TEE-agnostic attestation architecture
- Mappings to attestation architecture
Problem Statement

- Holistic view of attestation
- TEE-agnostic attestation architecture
- Mappings to attestation architecture
- Formal specs
Outline

1 Problem Statement

2 Contributions
   - Holistic View
   - TEE-agnostic Architecture
   - Mappings
   - Formal Specs
   - Design and Security Issues: TDX
   - Design and Security Issues: SCONE

3 Summary
Contributions

- Holistic View
- TEE-agnostic Architecture
- Mappings
- Formal Specs
- Design and Security Issues: TDX
- Design and Security Issues: SCONE
Holistic View of Attestation

- Initialization
- Provisioning
- Attestation Protocol
- Trustworthy Operations

Increasing frequency
2 Contributions

- Holistic View
- TEE-agnostic Architecture
- Mappings
- Formal Specs
- Design and Security Issues: TDX
- Design and Security Issues: SCONE
Attestation Architecture

- Limitations of RATS\(^1\)

---

Attestation Architecture

• Limitations of RATS\(^1\)
  • **Local attestation** out of scope (cannot express Intel’s attestation mechanisms)

---

Attestation Architecture

• Limitations of RATS\(^1\)
  • **Local attestation** out of scope (cannot express Intel’s attestation mechanisms)
  • Cannot express **anonymous** attestation (Intel EPID)

---

Attestation Architecture

• Limitations of RATS
  • Local attestation out of scope (cannot express Intel’s attestation mechanisms)
  • Cannot express anonymous attestation (Intel EPID)
  • Various ambiguities, e.g., role vs. entity

---

Attestation Architecture

- Limitations of RATS\(^1\)
  - **Local attestation** out of scope (cannot express Intel’s attestation mechanisms)
  - Cannot express *anonymous* attestation (Intel EPID)
  - Various ambiguities, e.g., role vs. entity

- **Errata** submitted for RATS

Attestation Architecture

- Limitations of RATS\(^1\)
  - **Local attestation** out of scope (cannot express Intel’s attestation mechanisms)
  - Cannot express **anonymous** attestation (Intel EPID)
  - Various **ambiguities**, e.g., role vs. entity

- **Errata** submitted for RATS

- **TEE-agnostic architecture**

Outline

2 Contributions

- Holistic View
- TEE-agnostic Architecture

Mappings

- Formal Specs
- Design and Security Issues: TDX
- Design and Security Issues: SCONE
Main Groups for Attestation

- **Frameworks**
  - (SCONE, Gramine, MAA, Veraison, ...)

- **Vendor solutions**
  - (Intel SGX, Intel TDX, AMD SEV-SNP, IBM PEF, ...)

- **Architecture lead solutions**
  - (Arm CCA, RISC-V, ...)

Muhammad Usama Sardar
SoK: Attestation in CC
March 29, 2023
### Overview of Related Work

<table>
<thead>
<tr>
<th>Related work</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>IETF RATS$^2$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ménétrey et al.$^3,4$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Niemi et al.$^5$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---


## Overview of Related Work

<table>
<thead>
<tr>
<th>Related work</th>
<th>Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>IETF RATS(^2)</td>
<td>Co-developed with DICE(^3)</td>
</tr>
<tr>
<td>Ménétrey et al.(^4,5)</td>
<td>Use RATS</td>
</tr>
<tr>
<td>Niemi et al.(^6)</td>
<td>Adapted from RATS</td>
</tr>
</tbody>
</table>


## Overview of Related Work

<table>
<thead>
<tr>
<th>Related work</th>
<th>Architecture</th>
<th>Mapping to group 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>IETF RATS²</td>
<td>Co-developed with DICE³</td>
<td>No</td>
</tr>
<tr>
<td>Ménétrey et al.⁴,⁵</td>
<td>Use RATS</td>
<td>Inaccurate for SGX</td>
</tr>
<tr>
<td>Niemi et al.⁶</td>
<td>Adapted from RATS</td>
<td>Very high level for SGX</td>
</tr>
</tbody>
</table>


## Overview of Related Work

<table>
<thead>
<tr>
<th>Related work</th>
<th>Architecture</th>
<th>Mapping to group 1</th>
<th>Mapping to group 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>IETF RATS(^2)</td>
<td>Co-developed with DICE(^3)</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Ménétrey et al.(^4,(^5))</td>
<td>Use RATS</td>
<td>Inaccurate for SGX</td>
<td>No</td>
</tr>
<tr>
<td>Niemi et al.(^6)</td>
<td>Adapted from RATS</td>
<td>Very high level for SGX</td>
<td>High level summary for CCA</td>
</tr>
</tbody>
</table>


## Overview of Related Work

<table>
<thead>
<tr>
<th>Related work</th>
<th>Architecture</th>
<th>Mapping to group 1</th>
<th>Mapping to group 2</th>
<th>Mapping to group 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>IETF RATS(^2)</td>
<td>Co-developed with DICE(^3)</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Ménétrey et al.(^4,(^5)</td>
<td>Use RATS</td>
<td>Inaccurate for SGX</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Niemi et al.(^6)</td>
<td>Adapted from RATS</td>
<td>Very high level for SGX</td>
<td>High level summary for CCA</td>
<td>No</td>
</tr>
</tbody>
</table>


Arm CCA Attestation Architecture Overview

Realm instance (RIM, REM)

RAK

RMM

Attesting Environment

Target Environment

Monitor Security Domain (System Boot State, CCA Parameters, pub(RAK))

Platform Evidence Binding

HES CPAK

Attesting Environment Platform Attester

RAK

RMM

Attesting Environment Realm Attester

Remote Evidence (Platform Evidence, Realm Evidence)

Verifier

Platform Evidence

Muhammad Usama Sardar
Outline

2 Contributions

- Holistic View
- TEE-agnostic Architecture
- Mappings
- **Formal Specs**
  - Design and Security Issues: TDX
  - Design and Security Issues: SCONE
Arm CCA Evidence Generation

- RMM
  - RAK

Attestation Request including challenge

- Prepare Realm claims-set including challenge and pub(RAK)
- Sign claims-set using RAK to form Realm Evidence

- [Platform Evidence, Realm Evidence]

- Realm

Verifier

- pub(CPAK)

- challenge

- [Platform Evidence, Realm Evidence]
Formal Analysis in ProVerif

• Assumptions
  • Verifier has **preconfigured pub(CPAK)** for signature verification
  • **Secure channel** between HES and RMM to transport the RAK key pair

• Integrity of Platform and Realm Evidence

\[
\text{query } data : \text{bitstring} ;
\]
\[
\text{event } (\text{accepted}(data)) \implies \text{inj-event } (\text{sent}(data)).
\]
Outline

2 Contributions

- Holistic View
- TEE-agnostic Architecture
- Mappings
- Formal Specs
- Design and Security Issues: TDX
- Design and Security Issues: SCONE
Claimed TCB

Legend
- Entity on Intel key server
- X.509 certs
- CRLs
- Entity on platform
- custom format cert-like structure

Figure 5.1. Trust Boundaries for TDX
Figure 5.1. Trust Boundaries for TDX

Figure: Old

Figure: Updated
SVN for TD?

- TD (Static and runtime measurements)
- TDX Module (SVN, measurements)
- TD QE (QE SVN, QE MRSIGNER, QE MRENCLAVE)
- PCE (PCE SVN)
- CPU HW and FW (CPU SVN)
- TDK
- AK
- AK cert
- PCK
- MAC Keys

Steps:
1. QE REPORT
2. AK cert
3. TDREPORT
4. TD Quote
Missing Specs

Provisioning phase

Structure of Remote Evidence (TD Quote)
Missing Specs

- Provisioning phase
- Structure of Remote Evidence (TD Quote)
- Structure of AK cert
Missing Specs

- Provisioning phase
- Structure of Remote Evidence (TD Quote)
- Structure of AK cert
- KDF for Local Evidence
Contributions

- Holistic View
- TEE-agnostic Architecture
- Mappings
- Formal Specs
- Design and Security Issues: TDX
- Design and Security Issues: SCONE
Order of QE selection

Chosen based on platform capabilities (not by app owner)

• Perspective 1
  1. DCAP QE (qe3)
  2. SCONE QE + EPID QE
  3. EPID QE

• Perspective 2
  1. DCAP QE (qe3)
  2. EPID QE
  3. SCONE QE (can use only if platform ID is known)

• Perspective 3
  • Everything (out of EPID, DCAP, SCONE Quote) that Platform 1 supports is sent to the CAS. So order is not important. CAS decides based on the policy.
    • food for thought: what do we gain?
    • unnecessary overhead without any apparent gain
Phases of SCONE Attestation

1. SCONE CLI (Platform 4) verifies CAS remotely
2. SCONE CLI (Platform 3) verifies CAS remotely
3. CAS verifies App enclave (Platform 1) remotely

Initialization
1. QE REPORT
2. AK cert

Attestation Protocol
3. REPORT
4. Quote

Muhammad Usama Sardar
SoK: Attestation in CC
March 29, 2023
When is a property attested?

![Diagram showing the attestation process between SCONE Runtime and CAS]

- auth data
  - version 4B
- upper protocol version 4B
- lower protocol version 4B
- client cert
  - pub(EK)
- nonce (from TLS) 32B

- SHA256

- Report data
  - 32B
  - 0x00 32B

- SPKI (RFC 5280)
- export-keying-material (RFC 5705)
Outline

1  Problem Statement

2  Contributions
   • Holistic View
   • TEE-agnostic Architecture
   • Mappings
   • Formal Specs
   • Design and Security Issues: TDX
   • Design and Security Issues: SCONE

3  Summary
Challenges

ca. 1500 pages of specs of TDX
Challenges

ca. 1500 pages of specs of TDX

Inherits specs from SGX (SDM alone ca. 5000 pages)
Challenges

- ca. 1500 pages of specs of TDX
- Inherits specs from SGX (SDM alone ca. 5000 pages)
- Specs in natural language
Challenges

- ca. 1500 pages of specs of TDX
- Inherits specs from SGX (SDM alone ca. 5000 pages)
- Specs in natural language
- Closed-source nature of SCONE
Take-home

- Towards TEE-agnostic verification infrastructure for transparency and interoperability
Take-home

- Towards TEE-agnostic *verification infrastructure* for transparency and interoperability

- **TDX**: how do we precisely express trust boundaries?
Take-home

- Towards **TEE-agnostic verification infrastructure** for transparency and interoperability

- **TDX**: how do we precisely express trust boundaries?

- **SCONE**: when do we say that something is attested?
Take-home

• Towards TEE-agnostic verification infrastructure for transparency and interoperability

• **TDX**: how do we precisely express trust boundaries?

• **SCONE**: when do we say that something is attested?

• Lots of work required for precise specification and standardization for understanding underlying assumptions
Take-home

• Towards **TEE-agnostic verification infrastructure** for transparency and interoperability

• **TDX**: how do we precisely express trust boundaries?

• **SCONE**: when do we say that something is attested?

• Lots of work required for **precise** specification and standardization for understanding underlying assumptions
  • Integration with TLS (**RA-TLS**)
Take-home

- Towards TEE-agnostic verification infrastructure for transparency and interoperability

- **TDX**: how do we precisely express trust boundaries?

- **SCONE**: when do we say that something is attested?

- Lots of work required for precise specification and standardization for understanding underlying assumptions
  - Integration with TLS (**RA-TLS**)
  - Integration with vTPM
Key References


Call to Action

- Get involved: https://github.com/CCC-Attestation/formal-spec-TEE
- Additional information: link here
- Specify your attestation designs using presented architecture and proposed formalism