Agenda

• Document Status
  – Changes from the last version

• Issue Update
Draft Status Update

• v03 published draft-ietf-teep-architecture-03
  – No more SD. Removed dependency on a SD.
  – Added TEEP Agent in TEE as an explicit entity
  – TEEP Broker vs. TEEP Agent vs. Agent clarified
• Additional issues resolved and updates proposed from an interim meeting but not reflected into draft update yet
Issues Closed

**Before IETF 104**

<table>
<thead>
<tr>
<th>Issue #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#3</td>
<td>TA Packaging and Distribution</td>
</tr>
<tr>
<td>#8</td>
<td>Multiple vs Single TEE in Device</td>
</tr>
<tr>
<td>#52</td>
<td>Session Based TA Provisioning &amp; Management</td>
</tr>
<tr>
<td>#57</td>
<td>Agent and Broker used concurrently</td>
</tr>
</tbody>
</table>

**After IETF 104**

<table>
<thead>
<tr>
<th>Issue #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#3</td>
<td>TA Packaging and Distribution</td>
</tr>
<tr>
<td>#8</td>
<td>Multiple vs Single TEE in Device</td>
</tr>
<tr>
<td>#52</td>
<td>Session Based TA Provisioning &amp; Management</td>
</tr>
<tr>
<td>#57</td>
<td>Agent and Broker used concurrently</td>
</tr>
</tbody>
</table>

**Issues Ready To Be Closed**

<table>
<thead>
<tr>
<th>Issue #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#7</td>
<td>Security Domain Clarification</td>
</tr>
<tr>
<td>#10</td>
<td>TEE signing first</td>
</tr>
<tr>
<td>#57</td>
<td>Agent and Broker used concurrently</td>
</tr>
</tbody>
</table>

---

Before IETF 104:

- **TEEP Architecture Diagram**: #39 by hannestshofenig was closed on Dec 10, 2018
- **Figure one arrow directionality correct?**: #36 by nicorwing was closed 16 days ago
- **Device administrator vs Device owner**: #29 by nicorwing was closed 16 days ago
- **Terminology for “Agent”**: #16 by hannestshofenig was closed on Nov 7, 2018
- **Applicability of TPMs to TEEP**: #15 by hannestshofenig was closed on Oct 22, 2018
- **Every Rich App Talks to TAM?**: #12 by hannestshofenig was closed on Nov 7, 2018
- **Attestation Agility**: #6 by hannestshofenig was closed on Nov 7, 2018
- **Option to not use secure boot**: #5 by hannestshofenig was closed on Nov 7, 2018
- **Algorithm Agility and Longer Key Sizes**: #4 by hannestshofenig was closed on Nov 7, 2018

After IETF 104:

- **Issue #**: #3
  - **Description**: TA Packaging and Distribution
- **Issue #**: #8
  - **Description**: Multiple vs Single TEE in Device
- **Issue #**: #52
  - **Description**: Session Based TA Provisioning & Management
- **Issue #**: #57
  - **Description**: Agent and Broker used concurrently
## GitHub Open Issues

<table>
<thead>
<tr>
<th>Issue #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#9</td>
<td>Install TA in a single pass</td>
</tr>
<tr>
<td>#11</td>
<td>Role of Client App</td>
</tr>
<tr>
<td>#13</td>
<td>Support for TA-to-TA dependency</td>
</tr>
<tr>
<td>#14</td>
<td>Multiple TAMs for a single Client App</td>
</tr>
<tr>
<td>#17</td>
<td>Capabilities of Attestation Mechanism</td>
</tr>
<tr>
<td>#30</td>
<td>Cardinality of Key Pair and Certificate</td>
</tr>
<tr>
<td>#31</td>
<td>SEED for TAM protocol</td>
</tr>
<tr>
<td>#32</td>
<td>Trust Anchor Lifecycle Management</td>
</tr>
<tr>
<td>#34</td>
<td>Dependencies between Client App &amp; TA</td>
</tr>
<tr>
<td>#35</td>
<td>Coordinate TA updates with Client App</td>
</tr>
<tr>
<td>#37</td>
<td>Sample Device Setup flow</td>
</tr>
<tr>
<td>#38</td>
<td>Trust Anchor Fingerprint</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Issue #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#51</td>
<td>Trust anchor format in a separate draft</td>
</tr>
<tr>
<td>#53</td>
<td>Editorial: regular operating system</td>
</tr>
<tr>
<td>#54</td>
<td>Editorial: regular/normal/typical OS</td>
</tr>
<tr>
<td>#55</td>
<td>Editorial: untrusted vs client app</td>
</tr>
<tr>
<td>#56</td>
<td>Editorial: device user – a human being</td>
</tr>
<tr>
<td>#58</td>
<td>Figure 6: difference btwn “device secure storage” and “device TEE” not clear</td>
</tr>
<tr>
<td>#59</td>
<td>Agent distribution</td>
</tr>
<tr>
<td>#62</td>
<td>Editorial: some SD ref still remains</td>
</tr>
<tr>
<td>#63</td>
<td>Clarification of location of keys, certs, CA</td>
</tr>
<tr>
<td>#64</td>
<td>End-to-end security for IP protection</td>
</tr>
</tbody>
</table>
Interim Meeting and Editor Working Session Update

• Interim meeting and working session on 5/17/2019
  – Symantec office, virtual meeting, and author / chairs working session
• Issues discussed and proposed resolutions
  – Security Domain (SD) resolution
  – Add TEEP Agent into architecture diagram
  – Terminology alignment (TEEP Broker, TAM Broker...)
  – Interaction flow and protocol specification completeness check
    • APIs between TEEP Broker and TEEP Agent
    • Interfaces between TEEP Broker and TAM Broker (Transport protocol APIs)
    • Call out functionality support need in architecture doc
  – TA distribution by a Client App
  – SP to TEE end-to-end security for personalization data
Security Domain Resolution (Issue #7)

- Issue: meaning and purpose of SD in TEEP
  - Is it a management component?
  - Is it an isolation mechanism?
  - Is it a key provisioning mechanism?
  - Is it necessary?

- Resolution:
  - TEEP doesn’t expose SD management APIs
  - Make SD implementation dependent if a TEE needs to use it under the cover
    - An implementation may still carry implicit SD information when an underlying TEE assumes a concept of SD

- Status
  - Architecture doc updated. No SD required for support.
  - Protocol doc will update schema to reflect this change
    - Device State Information (DSI)
TEEP Agent Added in Architecture Diagram (Issue #16, #57)
Terminology Alignment
(Issue #11, #16, #57)

1. Is name TAM Broker good?
2. Should it be formulated in Architecture doc?
Architecture and Protocol Spec Scope
(Issue #11)

2. TEEP Agent API in Arch doc? *Editors proposed “yes”*

4. Client App to TEEP Broker interface: *out of scope*

5. Is TAM API needed in Arch doc? *TBD*
TA Binary in a Client App Installation Implications (Issue #11)

- A Client App or Installer calls TEEP Broker to initiate TA installation
- TEEP Broker receives TA Binary from the Client App
- TEEP Broker calls “Request TA” API to TEEP Agent, including TA ID + TA hash
- TEEP Agent constructs a TAM Response Message back to TEEP Broker
  - A TA ID + TA hash will be sent to TAM so that TAM can make a policy decision if that TA can be allowed into a device
- TEEP Broker will send to TEEP Agent two pieces of information: InstallTA Message and TA binary if needed by a TEE

```
1. ReqTA(TA ID, hasBinary)
2. ReqTA
3. TA exist? Y/N
4. TAM URL
5. Connect
6. ProcTEEPMsg(Query)
7. Create QueryResp:
   { ReqedTA:
     {TA ID, TA Hash} }
8. QueryResp
9. InstallTA
10. ProcTEEPMsg(Install TA, [TA])
11. ProcTEEPMsg(TA, TAM URL)
```
End-to-End Security between SP and TEE (Issue #64)

• Requirement: yes

• Example case
  – An AI model provider for IoT devices wants to protect its IP. It shares TA with a manufacture, which hosts a TAM, to devices. However, it cannot share algorithms used in TA to the TAM.

• Proposal
  – A different data TAM for personalization data where the SP hosts this TAM itself
#13: Is it in scope: TA depends on another TA?

- Discussed in IETF 104
  - Concerns
    - Complex: very deep dependency
    - Circular dependency
  - Recommendation
    - Defer dependencies to SUIT manifest

- Status
  - Doc needs to be updated to reflect this, and then be closed
#14: Multiple TAMs for single Client App?

- Discussed in IETF 104
  - TAM is associated with a TA, not a Client App
  - A Client App may depend on multiple TAs
    - Two different TAs could be from different TAMs when multiple third party TAs are used by a Client App
    - However, a SP will typically provide the TAs of their own or work to acquire those third party TA binary to supply to a TAM on its choice.
  - Resolution
    - A Client App manifest file can contain all TAMs it may use to get TAs, normally just one
    - Go with simple case that single TAM is contacted by a TEEP Broker for a Client App
    - A SP provides each TAM that it places in the Client App’s manifest all the TAs that the app requires, so any TAM can provide all the TA’s
    - For third party TAs that a Client App may depend on, a TAM can reach out to the original TAMs for those third party TAs that it is missing, but this would be a TAM implementation specialization
- Status
  - Doc update to add that TAM URL decided by TEEP Agent, not TEEP Broker
#17 Capabilities of the Attestation Mechanism

• Changes made to:
  – Define attestation
  – Describe assumptions required for an attestation
  – Identify the need to support both proprietary and standard attestation signatures

• Status
  – Proposed format of attestation may need more work & discussion
  – New issue #12 about alignment with RATS
Attestation Structure

- Attestation Type
- Signature Type
- Version Number
- Manufacturer and Device Unique Identifiers
- TEE Manufacturer and TEE Type and Version Numbers
- Nonce And/or Timestamp
- Claims based on Attestation Type In Header
- Optional Claims required by TEE Type or required by Requestor

IETF105
Attestation Work still to Complete

- Update format based on feedback
- Provide clear direction for the mapping of Device, TEE, and TA attributes in the format
- Provide formats for TEEP standard claims
- Provide examples of real attestation (suggest SGX and ARM TZ)
#32 / #51 Trust Anchor Update

• Trust Anchor update must be considered for the completeness of the Trust Anchor lifecycle management

• Two options
  – Part of architecture draft, synchronized with the SUIT definitions
  – A separate draft work for the full definition of the Trust Anchor lifecycle (creation/provisioning, use, update)

• Current preference
  – Defer complete definition for a separate draft document, but provide basic definitions aligned to SUIT and the use of the Trust Anchors in the architecture document.
  – A solution discussed was to use a system Manager TA pre-installed in a TEE for check and update of Trust Anchors

• A related question
  – Trust Anchor format: leave it to TEE implementation or define it in TEEP?
    • Trust anchors could be inside TEE or stored outside of TEE
  – If defined in TEEP a very comprehensive document with many implementation options must be provided (including fuses, one-time-programmable bits (OTP), locked flash, battery backed RAM, PUFs, etc)
#9: Install TA in Single Pass?

• Discussed in IETF 104
  – Not always
  – Flow update per Hackathon feedback
    • Initial TAM GET call is necessary
      – Only provide device signing key information to a trusted TAM, not others
    • Optimize to do this Single Pass for a device that has had cached TAM information
  – David T new draft content
    • To be merged back to the core protocol document

• Status
  – Need update both architecture and protocol doc
#10: Local TEE Signing First

• Issue
  – One proposal was put forward to make the TEE connect to the TAM using an attestation of the platform and include any “installTA” requests in the message
  – The objection was stated as: Local TEE signing first would leak the TEE signing key to potentially unknown TAM

• Resolution
  – A TAM round trip is still needed unless a TAM certificate is cached. Otherwise a TEEP Agent will not initiate a signing; it may only return TAM URL that it trusts to install a TA.
  – Protocol doc will elaborate this flow. Old flow removed from the Arch doc.

• Status
  – Ready to be closed for Architecture Doc.
Issue #52: Alternate Session based TA Provisioning

• Issue
  – Anders suggested use an alternative protocol approach
    • Negotiate a session key first, and then use that session key for future attestation
    • Use a binary protocol to TEE and a conversion with JSON

• Responses
  – Dramatic change to the protocol with a session negotiation binary flow
  – Binary protocol vs. JSON / CBOR protocol
  – IP patented

• Status
  – Lack of support to make this change. The filer closed the issue.
Q&A

Thank you!