TEEP Architecture
draft-ietf-teep-architecture-13

Dave Thaler (presenting)

Ming Pei, David Wheeler, Hannes Tschofenig

Timeline

- JAN, 2020: WGLC completed
- FEB, 2020: TEEP virtual interim meeting
- APR, 2020: Additional reviews done by Russ Housley and Daniel Migault
- JUL, 2020: IETF 108
  - Two action items were agreed on in the meeting:
    - Move broker architecture section from transport spec to architecture doc (DONE)
    - Add “Unneeded TAs” point next to “Requested TAs” in claims discussion (see next slide)
  - Minutes say:

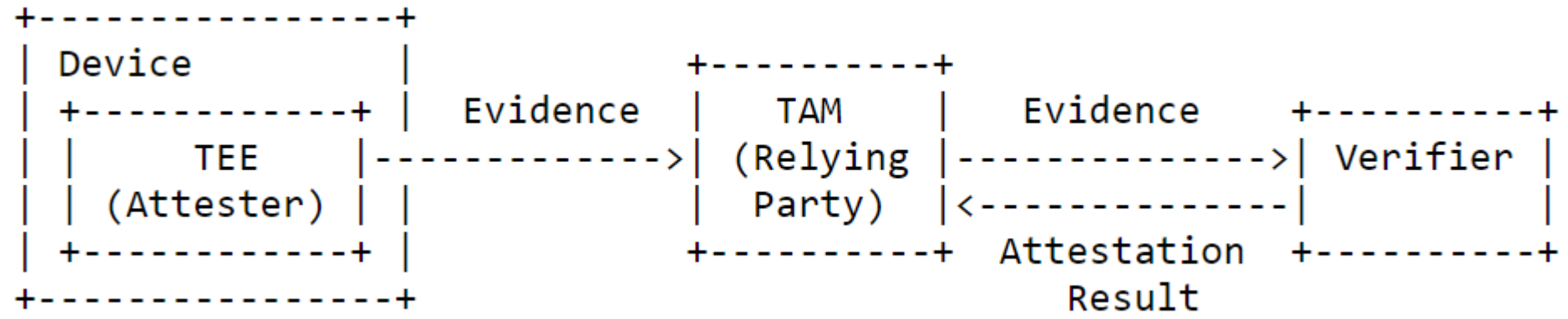
    “2 - Architecture document: Dave Thaler; noted two issues to be resolved, but those can be addressed during IETF 108, so August 2020 is still realistic.

    Hannes: suggests moving to September

    Dave T: fine; it can always be submitted to the IESG earlier if it's done earlier.”

#208: “What goes in claims vs TEEP protocol fields”

- Oct. 16 list post with above title made proposal with rationale



- ***Claims*** *should be used for fields that are important to the* ***Verifier****, whereas* ***protocol fields*** *should be used for information that is destined for the* ***TAM*** *and not important to the Verifier.*
  - Requested & Unneeded TAs are info used by the TAM not Verifier
- No objections raised, so “Requested TAs“ (and “Unneeded TAs”) removed from claims discussion

#182: Use of symmetric keys

- Addressed Tiru's feedback about clarifying use of symmetric keys:
  - "In the context of TEEP, symmetric algorithms are used for encryption and integrity protection of TA binaries and personalization data whereas the asymmetric algorithms are used for signing messages and managing symmetric keys."

- Tiru commented "Update looks good" and closed the issue ☺

#206: Remaining comments from Russ

- Augmented confidential cloud computing with example of use:
  - A tenant can store sensitive data, **such as customer details or credit card numbers**, in a TEE in a cloud computing server such that only the tenant can access the data, preventing the cloud hosting provider from accessing the data.
- Augmented “Compromised REE” security considerations:
  - “We have already seen examples of attacks on the public Internet of billions of compromised devices being used to mount DDoS attacks. A compromised REE can be used for such an attack but it cannot tamper with the TEE's code or data in doing so. A compromised REE can, however, launch DoS attacks against the TEE.”
- Accepted other editorial nits
- Chairs verified and closed the issue ☺

PR #212: “Trusted Applications vs Trusted Components”

- Oct. 26 list post with above title made proposal with rationale
- “TA” defined as an app that “runs” in a TEE, meaning code not data-only
- Already allowed for TEEP protocol to install “personalization data” that is required by, but not bundled with, code
- Other text only talked about installing “TAs” (not data-only) so inconsistent
- One sentence already used “trusted components” to mean TA OR Personalization Data
- Proposed adding “Trusted Components” to terminology section and updating text that should support both
- No objections raised, so done in draft -13 and similar in TEEP protocol spec

Issue #213/PR #216: UnrequestTA

- **New** issue found during Hackathon implementing IETF 108 consensus
- IETF 108 discussion: added way for TEEP protocol to tell TAM which TA's are no longer needed so the TAM can choose to remove them
- Issue: no way to communicate that to the TEEP Agent
- Fix: add UnrequestTA abstract API like existing RequestTA abstract API

  "UnrequestTA: A notification from an REE application (e.g., an installer, or an Untrusted Application) that it no longer depends on a given Trusted Component, which may or may not already be installed in the TEE. For example, if the Untrusted Application is uninstalled, the uninstaller might invoke this conceptual API."
- Affects architecture spec, transport spec, and protocol spec

Issue #214/PR #215: Obsolete paragraph

- New editorial-only issue
- Problem: last paragraph below is about security domains, but all other security domain text was previously removed

  "The Trusted Execution Environment Provisioning (TEEP) protocol addresses the following problems:

  ...

  - A TA developer wants to define the relationship between cooperating TAs under the TA developer's control, and specify whether the TAs can communicate, share data, and/or share key material."
- Fix: delete the paragraph

Issue #217 / PR #218: Applicability text? (1/2)

- TEEP protocol won't be used for everything:

  1) GSMA standardized a TEEP-like protocol for managing trusted applets in SIM chips, in wide use by mobile operators

  2) GlobalPlatform standardized OTrP for managing trusted apps in secure elements

  3) "Classic" use of SGX, and use of REE TA Store in OP-TEE/TrustZone don't need TEE to "install" or "enumerate" apps, contrary to what one might assume based on SGX and TrustZone text in the doc

- Propose adding paragraph explaining that TEEP is applicable to TEEs that install/enumerate apps ***in*** a TEE, where a domain-specific protocol standard is not already in use
  - E.g., OP-TEE/TrustZone's Secure TA store, use of SGX for a LibOS style use (see IETF 106 presentation)

Issue #217 / PR #218: Applicability text? (2/2)

- Intro section:
  - For TEEs that simply verify and load signed TA's from an untrusted filesystem, classic application distribution protocols can be used without modification. The above problems require a new protocol, i.e., the TEEP protocol, for TEEs that can install and enumerate TAs in a TEE-secured location and where another domain-specific protocol standard (e.g., [GSMA], [OTRP]) that meets the needs is not already in use.
- SGX section:
  - As long as signed files (TAs and/or Personalization Data) are installed into an untrusted filesystem and trust is verified by the TEE at load time, classic distribution mechanisms can be used. Some uses of SGX, however, allow a model where an unmodified TA can be dynamically installed into an SGX enclave that provides a runtime platform for such TAs. The TEEP protocol can be used in such cases, where the runtime platform could include a TEEP Agent.
- TrustZone section:
  - TEE OS's (e.g., OP-TEE) that support loading and verifying signed TAs from an untrusted filesystem can, like SGX, use classic file distribution mechanisms. If secure TA storage is used (e.g., a Replay-Protected Memory Block device) on the other hand, the TEEP protocol can be used to manage such storage.