Standardization Work for RATS

Standardization work under the IETF Remote ATtestation ProcedureS (RATS) Working Group is near completion, and multiple Internet-Drafts documents defining RATS Architecture, Reference Interaction Models, Entity Attestation Token (EAT) and such are already available.[1] Since it is inappropriate to use Internet-Drafts as reference material, we will cite these as "work in progress" sources here. This document summarizes the Remote Attestation concepts our project group has examined so far in our study of these Drafts, and highlights some questions arising about their implementation on our TEE-equipped Boards at this stage of our study.

RATS ARCHITECTURE

The underlying concept of Remote Attestation Procedures involves a peer called an "A*ttester*" which produces supposedly accurate information about itself ("*evidence*"), to enable a remote peer called the "R*elying Party*" to decide whether to consider that attester a trustworthy peer or not. Another vital Party, the *Verifier*, appraises evidence through *appraisal policies* and creates *Attestation Results* to support the Relying Parties in their decision process. It is interesting to note the distinction in this context between Trust, which is a choice one makes about another system, and Trustworthiness, which is a quality about the other system that can be used in making one's decision to trust it or not. An attester makes *claims* about its trustworthiness, which form part of the evidence which must be appraised (compared against a set of "*known good*" values) in order to determine the extent to which the attester is considered trustworthy. [2]

=> Terminology

The IETF RATS Architecture Internet-Draft defines a set of useful Terminology that provides well-understood meanings for themes common across Remote Attestation Procedures such as roles, device composition, topological models and appraisal. We feel it is important to list the over-arching definitions here as they will be vital for semantic interoperability across solutions[2] :

- ***Appraisal Policy for Evidence:*** A set of rules that informs how a Verifier evaluates the validity of information about an Attester.

- ***Appraisal Policy for Attestation Results:*** A set of rules that direct how a Relying Party uses the Attestation Results regarding an Attester generated by the Verifiers.

- ***Attestation Result:*** The output generated by a Verifier, typically including information about an Attester, where the Verifier vouches for the validity of the results

- ***Attester:*** A role performed by an entity (typically a device) whose Evidence must be appraised in order to infer the extent to which the Attester is considered trustworthy, such as when deciding whether it is authorized to perform some operation

- ***Claim:*** A piece of asserted information, often in the form of a name/value pair.

- ***Endorsement***: A secure statement that an Endorser vouches for the integrity of an Attester's various capabilities such as Claims collection and Evidence signing

- ***Endorser:*** An entity (typically a manufacturer) whose Endorsements help Verifiers appraise the authenticity of Evidence

- ***Evidence:*** A set of information about an Attester that is to be appraised by a Verifier. Evidence may include configuration data, measurements, telemetry, or inferences.

- ***Reference Value Provider:*** An entity (typically a manufacturer) whose Reference Values help Verifiers appraise the authenticity of Evidence.

- ***Reference Values:*** A set of values against which values of Claims can be compared as part of applying an Appraisal Policy for Evidence. Reference Values are sometimes referred to in other documents as known-good values, golden measurements, or nominal values, although those terms typically assume comparison for equality, whereas here Reference Values might be more general and be used in any sort of comparison.

- ***Relying Party:*** A role performed by an entity that depends on the validity of information about an Attester, for purposes of reliably applying application specific actions.

- ***Relying Party Owner:*** An entity (typically an administrator), that is authorized to configure Appraisal Policy for Attestation Results in a Relying Party

- ***Verifier:*** A role performed by an entity that appraises the validity of Evidence about an Attester and produces Attestation Results to be used by a Relying Party

- ***Verifier Owner:*** An entity (typically an administrator), that is authorized to configure Appraisal Policy for Evidence in a Verifier

=> Architectural Overview

The RATS Architecture Internet-Draft mentions some reference use-cases for RATS to provide motivation for aspects of the architecture presented. Comments on the various use-cases or possible implementations for our LPC55S69 boards will form the subject of a different document. Regardless of protocol or use cases, figure 1 shows the conceptual data flow between different roles in Remote Attestation Procedures. The communication between the Attester, the Verifier, and the Relying Party can actually be implemented under various topological models, which are discussed later in this document.

To asses the trustworthiness of an Attester, the Verifier applies an Appraisal Policy for Evidence using the Evidence, and any *Endorsements* from Endorsers. The resulting *Attestation Results* are relayed to the Relying Party, which applies its own Appraisal Policy for Attestation Results to determine which degree to trust the attester.

```
************   *************    ************    *****************
* Endorser *   * Reference *    * Verifier *    * Relying Party *
************   * Value     *    *  Owner   *    *  Owner        *
   |           * Provider  *    ************    *****************
   |           *************          |                 |
   |                  |               |                 |
   |Endorsements      |Reference      |Appraisal        |Appraisal
   |                  |Values         |Policy           |Policy for
   |                  |               |for              |Attestation
   .-----------.      |               |Evidence         |Results
               |      |               |                 |
               |      |               |                 |
               v      v               v                 |
             .-----------------------------.              |
      .----->|          Verifier         |------.       |
      |      '---------------------------'      |       |
      |                                         |       |
      |                              Attestation|       |
      |                              Results    |       |
      | Evidence                                |       |
      |                                         |       |
      |                                         v       v
.----------.                                .---------------.
| Attester |                                | Relying Party |
'----------'                                '---------------'
```

Figure 1: Conceptual Data Flow [3]

The Appraisal Policy for Evidence might be configured in the Verifier by the Verifier Owner, or be procured via a different mechanism such as being sent by an Endorser along with the Endorsements. The Appraisal Policy for Attestation Results is configured and/or programmed into the Relying Party by the Relying Party Owner. During the appraisal processes, values of claims about the attester are compared with the constraints defined by the appraisal policy.

This could involve tests of equality, range checks, membership checks or any other comparison of claim values against Reference Values. These Reference Values might be configured as a part of the appraisal policy itself, or obtained via a different mechanism such as through an Endorsement. The Data Format and Semantics of these Reference Values are are specific to claims and implementations across solutions [4].

=> Attester and Layered Attestation Environments

The RATS Architecture Draft defines a flexible architecture specification for the Attester allowing for more complex implementations depending on different use-case scenarios. Figure 2 shows the underlying architectural structure. The Attester must consist of at least one Target Environment, and an Attesting Environment. The latter collects values and information about the Target Environment by reading system registers and variables, calling into subsystems and taking measurements on code or memory. These values are represented in *Claims* about the Target Environment which are then properly formatted by the Attesting Environment, and turned into *Evidence*, typically by using cryptographic signing or cipher

```
  .--------------------------------.
  |                                |
  |            Verifier            |
  |                                |
  '--------------------------------'
                          ^
                          |
.-------------------------|----------.
|                         |          |
|   .----------------.    |          |
|   | Target         |    |          |
|   | Environment    |    |          |
|   |                |    | Evidence |
|   '----------------'    |          |
|                   |     |          |
|                   |     |          |
|          Collect  |     |          |
|           Claims  |     |          |
|                   |     |          |
|                   v     |          |
|                 .-------------.    |
|                 | Attesting   |    |
|                 | Environment |    |
|                 |             |    |
|                 '-------------'    |
|               Attester             |
'------------------------------------'
```

Figure 2: Attester Environments [5]

algorithms. Depending on implementation, the Attesting and Target Environments may be combined, or there may be multiple of each within Composite Devices.

The Attester may contain one or more nested or cascading staged environments, where each environment has the responsibility of measuring the next environment before the next environment is started. A chained Attestation Evidence is produced with a component typically serving as the root of trust. Figure 3 extracted from the Draft provides an example of such a layered Attestation Environment. The number of layers may vary across implementations and one Attestation Environment may have multiple Target Environments to measure.

Attesting Environments typically exist in trusted Execution Environments (TEEs), embedded Secure Elements (eSEs), and BIOS Firmware. Considering an Attester Device with BIOS written to ROM as the root of trust, an attesting environment in the BIOS would be responsible for ensuring the integrity of the bootloader. The bootloader in turn would contain an Attesting environment for which the Target Environment would be the Kernel. In this context, it is important to ensure that the bootloader attesting environment is not able to alter any Claims about the bootloader itself. This is done either by the BIOS Attesting Environment signing those claims about the bootloader, or storing them in an untamperable manner.

The final Evidence generated and sent to the Verifier by the Attesting Environment in the bootloader would contain a set of Claims about the bootloader measured and signed by the BIOS, and one set of claims about the Kernel measured and signed by the bootloader. This structure could be extended further by making the Kernel contain another Attesting Environment for some application as the Target Environment, which would result in an additional set of claims about the Application measured and signed by the Kernel. The purpose of presenting this example architectural structure here is to help understand the

```
    .----------.                    .----------.
    |          |                    |          |
    | Endorser |------------------->| Verifier |
    |          |    Endorsements    |          |
    '----------'  for A, B, and C   '----------'
                                          ^
.------------------------------------.    |
|                                    |    |
|   .---------------------------.    |    |
|   | Target                    |    |    | Layered
|   | Environment               |    |    | Evidence
|   | C                         |    |    |   for
|   '---------------------------'    |    | B and C
|           Collect |                |    |
|           claims  |                |    |
|   .---------------|-----------.    |    |
|   | Target        v           |    |    |
|   | Environment .-----------. |    |    |
|   | B           | Attesting | |    |    |
|   |             |Environment|-----------'
|   |             |     B     | |    |
|   |             '-----------' |    |
|   |                     ^     |    |
|   '---------------------|-----'    |
|           Collect |     | Evidence |
|           claims  v     |  for B   |
|                 .-----------.      |
|                 | Attesting |      |
|                 |Environment|      |
|                 |     A     |      |
|                 '-----------'      |
|                                    |
'------------------------------------'
```

Figure 3: Layered Attester[6]

flexibility of layered attestation as defined in the Draft and how it can be used to create more complex Attestation structures depending on different solutions.

Citiations

1. Remote Attestation Procedures (RATS), Active Internet-Drafts. Retrieved: *https://datatracker.ietf.org/wg/rats/documents/*

2. H. Birkholz, D. Thaler, M. Richardson, N. Smith, W. Pan. Remote Attestation Procedures Architecture, draft-ietf-rats-architecture-07, p.4-5. Retrieved: https://datatracker.ietf.org/doc/draft-ietf-rats-architecture/?include_text=1

3. H. Birkholz, D. Thaler, M. Richardson, N. Smith, W. Pan. Remote Attestation Procedures Architecture, draft-ietf-rats-architecture-07, p9., Retrieved: https://datatracker.ietf.org/doc/draft-ietf-rats-architecture/?include_text=1

4. H. Birkholz, D. Thaler, M. Richardson, N. Smith, W. Pan. Remote Attestation Procedures Architecture, draft-ietf-rats-architecture-07, p10., Retrieved: https://datatracker.ietf.org/doc/draft-ietf-rats-architecture/?include_text=1

5. H. Birkholz, D. Thaler, M. Richardson, N. Smith, W. Pan. Remote Attestation Procedures Architecture, draft-ietf-rats-architecture-07, p11., Retrieved: https://datatracker.ietf.org/doc/draft-ietf-rats-architecture/?include_text=1

6. H. Birkholz, D. Thaler, M. Richardson, N. Smith, W. Pan. Remote Attestation Procedures Architecture, draft-ietf-rats-architecture-07, p12., Retrieved: https://datatracker.ietf.org/doc/draft-ietf-rats-architecture/?include_text=1