Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 13, 2019

H. Birkholz
Fraunhofer SIT
M. Wiseman
GE Global Research
H. Tschofenig
ARM Ltd.
N. Smith
Intel

March 12, 2019

Architecture and Reference Terminology for Remote Attestation Procedures
draft-birkholz-rats-architecture-01

Abstract

Remote ATtestation ProcedureS (RATS) architecture facilitates the attestation of device characteristics that, in general, are based on specific trustworthiness qualities intrinsic to a device or service. It includes trusted computing functionality provided by device hardware and software that allows trustworthiness qualities to be asserted and verified as part of, or pre-requisite to, the device's normal operation. The RATS architecture maps corresponding attestation functions and capabilities to specific RATS Roles. The goal is to enable an appropriate conveyance of evidence about device trustworthiness via network protocols. RATS Roles provide the endpoint context for understanding the various interaction semantics of the attestation lifecycle. The RATS architecture provides the building block concepts, semantics, syntax and framework for interoperable attestation while remaining hardware-agnostic. This flexibility is intended to address a significant variety of use-cases and scenarios involving interoperable attestation. Example usages include, but are not limited to: financial transactions, voting machines, critical safety systems, network equipment health, or trustworthy end-user device management. Existing industry attestation efforts may be helpful toward informing RATS architecture. Such as: Remote Integrity VERification (RIVER), the creation of Entity Attestation Tokens (EAT), software integrity Measurement And ATtestation (MAAT)

Commented [DT1]: Need to fix grammar

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute

Copyright Notice

Table of Contents

1. Introduction

In general, this document provides normative guidance how to use, create or adopt network protocols that facilitate remote attestation procedures. The RATS Architecture anticipates broad deployment contexts that range from IoT to Cloud and Edge ecosystems. The foundation of the RATS architecture is the specification of RATS Roles that can be chained via RATS Interactions and - as a result - may be composed into use-case specific Remote Attestation Procedures. RATS Actors establish an ecosystem neutral context where RATS Roles are hosted and where a variety of Remote Attestation Procedure interactions are defined independent of specific conveyance protocols or message formats. In summary, the goal of the RATS Architecture is to enable interoperable interaction between the RATS Roles. Hence, the RATS Architecture is designed to enable interoperability via well-defined semantics of the information model (attestation assertions/claims), associated with RATS Roles following a conveyance model (RATS Interactions) that may be used to compose domain-specific remote attestation solutions.

1.1. What is Remote Attestation

Unfortunately, the term Attestation itself is an overloaded term. In consequence, the term Remote Attestation covers a spectrum of meanings. The common denominator encompasses the creation, conveyance, and appraisal of evidence pertaining to the trustworthiness characteristics of the creator of the evidence. In essence, RATS are used to enable the assessment of the trustworthiness of a communication partner.

1.2. The purpose of RATS Architecture and Terminology

To consolidate the utilization of existing and emerging network protocols in the context of RATS, this document provides a detailed definition of Attestation Terminology that enables interoperability between different types pf RATS. Specifically, this document illustrates and remediates the impedance mismatch of terms related to Remote Attestation Procedures used in different domains today. As an additional contribution, new terms defined by this document provide a common basis that simplifies future work on RATS in the IETF and beyond.

Commented [DT2]: typo

1.3. Requirements notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. RATS Architecture

One of the goals of the RATS Architecture is to provide the building blocks - the roles defined by the RATS Architecture - to enable the composition of service-chains/hierarchies and work-flows that can create and appraise evidence about the trustworthiness of devices and services.

The RATS Architecture is based on the use-cases defined in [I-D.richardson-rats-usecases].

The RATS architecture specifies:

- The building blocks to create remote attestation procedures applicable Actors, Roles, Duties, and Interactions,

Commented [DT3]: can't parse grammar

- Mandatory and optional trust relationships between its Roles, that may assume a Root-of-Trust context,

- The interaction between Roles that reside on separate Actors and interact via network protocols,

- Protocol/message framing that allows for well-defined and opaque payloads,

- The means to prove, preserve and convey trust properties, such as identity, varacity, freshness, or provenance, and

Commented [DT4]: typo: veracity

- Primitives necessary for the construction of interoperable attestation payloads.

3. Architectural Components

The basic architectural components defined in this document are:

- RATS Roles

- RATS Actors

- RATS Duties

- RATS Interactions

The following sub-section define and elaborate on these terms:

3.1. RATS Roles

A Role in the context of usage scenarios for remote attestation procedures is providing a service to other Roles. Roles are building blocks that can be providers and consumers of information. In the RATS architecture, devices or services can take on RATS roles. They are composites of internal functions (RATS Duties) and external functions (RATS Interactions) that facilitate a required (sometimes optional) task in a remote attestation procedure.

The base set of RATS roles is:

Claimant: The producer of trustworthiness assertions pertaining to an Attester; that may or not have a root-of-trust for measurement.

It is not guaranteed that a Verifier Role can appraise the output of a Claimant via reference values (in contrast to the output of an Attester).

Examples of Claimant assertions include: * The hardware, firmware and software components of the Attester. * The manufactuer of Attester components. * The Attester's current configuration. * The Attester's current location - e.g. GPS coordinates. * The method by which binding of an attester to an RTR. * The identifier(s) available for identifying and authenticating the Attester - e.g. Universal Entity ID (UEID).

Commented [DT5]: Looks like a formatting issue, I expect these were meant to be bullets

Commented [DT6]: typo

Commented [DT7]: nit: "e.g." should always be followed by a comma (Chicago Manual of Style section 5.202, and similar in other style guides)

Commented [DT8]: Undefined acronym. Expand on first use (it's expanded below but not here)

Typically, claimant role are taken on by RATS Actors that supply chain entities (SCE). Various assertions (often represented as Claims or Trusted Claims Sets, e.g. [I-D.mandyam-eat] or [I-D.tschofenig-rats-psa-token]).

Commented [DT9]: Fix grammar

Commented [DT10]: Can't parse grammar

Attester: The producer of attestation evidence that has a root of trust for reporting (RTR) and implements a conveyance protocol, authenticates using an attestation credential, consumes assertions about itself and presents it to a consumer of evidence (e.g. a relying party or a verifier). Every output of an attester can be appraised via reference values.

Commented [DT11]: Undefined term. Define it or cite a reference for its definition

Commented [DT12]: What does "it" refer to here? ("assertions" is plural not singular so can't tell if that's what was meant)

Authentication Checker: The consumer of signed assertions such as trusted claim sets or attestation evidence that assesses the trustworthiness or other trust relationships of the information consumed via trusted third parties or external trust authorities, such as a privacy certificate authority. In certain environments, an Authentication Checker can assess a system's trustworthiness via external trust anchors, implicitly.

Verifier: The consumer of attestation evidence that has a root of trust for verification (RTV), implements conveyance protocols, appraises attestation evidence against reference values or policies, and makes verification results available to relying parties.

Commented [DT13]: What is the difference between "verification" and "authentication" (as done by an Authentication Checker)? Clarify.

Relying Party: The consumer and assessor of verifier or Authentication Checker results for the purpose of improved risk management, operational efficiency, security, privacy (natural or legal person) or safety. The verifier and/or authentication checker roles and the relying party role may be tightly integrated.

4. RATS Actors

RATS Actors may be any entity, such as an user, organization, execution environment, device or service provider, that takes on (implements) one or more RATS Roles and performs RATS Duties and/or RATS Interactions. RATS Interactions occur between RATS Actors. The methods whereby RATS Actors are identified, discovered, and

Commented [DT14]: Typo: "a"

connectivity established are out-of-scope for this architecture. In contrast, if multiple RATS Roles reside on a single RATS Actor, the definition of RATS Interactions is out-of-scope of the RATS architecture, if no network protocols are required.

```
                +------------------+   +------------------+
                |      Actor 1     |   |      Actor 2     |
                |  +------------+  |   |  +------------+  |
                |  |            |  |   |  |            |  |
                |  |   Role 1   |<-|---|->|   Role 2   |  |
                |  |            |  |   |  |            |  |
                |  +------------+  |   |  +------------+  |
                |                  |   |                  |
                |  +-----+------+  |   |  +-----+------+  |
                |  |            |  |   |  |            |  |
                |  |   Role 2   |<-|---|->|   Role 3   |  |
                |  |            |  |   |  |            |  |
                |  +------------+  |   |  +------------+  |
                |                  |   |                  |
                +------------------+   +------------------+
```

Figure 1: RATS Actor-Role Interactions

RATS Actors have the following properties: * Multiplicity - Multiple instances of RATS Actors that possess the same RATS Roles can exist. * Decomposability - A singleton RATS Actor possessing multiple RATS Roles can be separated into multiple RATS Actors. RATS Interactions may occur between them. * Composablility - RATS Actors possessing different RATS Roles can be combined into a singleton RATS Actor possessing the union of RATS Roles. RATS Interactions between combined RATS Actors ceases.

Commented [DT15]: Looks like a formatting issue

Interactions between RATS Roles belonging to the same RATS Actor are generally believed to be uninteresting. Actor operations that apply resiliency, scaling, load balancing or replication are generally believed to be uninteresting.

4.1. RATS Duties

A RATS Role can take on one ore more duties. RATS Duties are role-internal functions that do not require interaction with other RATS Roles. In general, and RATS Duties are typically associated with a RATS Role. The list presented in this document is exhaustive. Also, there can be usage scenario where RATS Duties are associated with other RATS Roles than illustrated below:

Commented [DT16]: typo

4.1.1. Attester Duties

- Acquisition or collection of assertions about itself

Commented [DT17]: Grammar mismatch between this bullet and later ones. Be consistent in tense. “Acquire or collect” would match “Provide or create” and “Create” in the other bullets. Same issue in the bullets in the next couple sections too.

- Provide or create proof that an assertion is bound to the Attester
- Create Evidence from assertion bundles via roots-of-trust

Commented [DT18]: Can’t understand this bullet yet. Maybe it requires background definitions that aren’t present until later in the doc, in which case maybe the sections are in the wrong order?

4.1.2. Verifier Duties

- Acquisition and storage of assertion semantics

Commented [DT19]: No idea what this means

- Acquisition and storage of appraisal policies
- Verification of Attester Identity (attestation provenance)
- Comparing assertions or evidence with reference values according to appraisal policies
- Validate authentication information based on public keys, signatures, secrets that are shielded, or secrets that are access restricted via protection profiles

Commented [DT20]: Undefined term

4.1.3. Claimant Duties

- Hardens the device or service that implements the Attester role
- Provisions device identities and/or key material accessible to the Attester role
- Evaluates trustworthiness during manufacturing, supply chain and onboarding
- Produces trustworthiness assertions applicable to the Attestor role

Commented [DT21]: typo

- Embeds trustworthiness assertions about the Attester role in the device or service during manufacturing, supply chain or onboarding

4.1.4. Relying Party Duties

- Evaluate assertions/evidence locally as far as possible

Commented [DT22]: Not true in all use cases. Elaborate on what you mean by this. Some use cases may simply rely on the verifier to do all evaluation and only check whether it was verified or not.

- Compare trust policies to attestation-results based on assertions or evidence
- Enforce policies or create input for risk engines

4.1.5. RATS Interactions

The flow of information between RATS Roles located on RATS Actors compose individual remote attestation procedures. The RATS Architecture provides a set of standard interactions between the RATS Roles defined in this document in order to enable this composability. In this section, common interactions between roles are specified. This list of interactions is not exhaustive, but provides the basis to create various standard RATS.

Every RATS Interaction specified below is based on the information flow between two RATS Roles defined above. Every RATS Interaction is conducted via an Interconnect between corresponding RATS Roles that RATS Actors take on. If more than one RATS Role resides on the same RATS Actor, a network protocol might not be required. If RATS Roles are collapsed into a singular RATS Actor in this way, the method of conveying information is out-of-scope of this document. If network protocols are used to convey corresponding information between RATS Roles (collapsed on a singular RATS Actor or not), the definitions and requirements defined in this document apply.

Commented [DT23]: Undefined term. (not defined until the next paragraph)

In essence, an Interconnect is an abstract "distance-less" channel between RATS Actors that can range from General Purpose Input Output (GPIO) interfaces to the Internet.

Attester/Verifier: The most basic RATS interaction is between the creator of evidence (Attester) and its complementary remote attestation service (Verifier). In order to convey evidence (or assertions that are not accompanied by a proof of their validity) this RATS Interaction is required.

Commented [DT24]: Typo: creater

Commented [DT25]: I think this word choice is bad. If the Verifier is the RP's verifier rather than the Attester's verifier, then this wording is bad.

Commented [DT26]: Is this a definition of the word "evidence" or an alternative thing that is not "evidence"?

Attester/Relying-Party: A Relying Party typically requires external help to either validate authentication information or to appraise evidence presented by an Attester. In most cases, a Relying Party requires a corresponding Verifier to process the assertions/evidence received. In consequence, (a subset of) the information received by an Attester must be relayed securely to a Verifier.

Relying-Party/Verifier: Typically, trusted assertions or evidence are conveyed from an Attester to a Relying Party. In an open ecosystem, such as the Internet, the appraisal of the evidence presented by an Attester provided in order to assess its trustworthiness requires a remote attestation service. Hence, either the RATS roles of Verifier and Relying Party are collapsed and compose a single RATS Actor, or - if they reside on separate RATS Actors - a Relying Party requires appropriate configuration or a discovery/join/rendezvous service to initiate a RATS Interaction with an appropriate and trusted Verifier.

Commented [DT27]: This term is bad since "trusted" is not useful without saying whom it's trusted by. What's trusted by the RP and what's trusted by the Attester can be different. Hence clarify and don't use ambiguous terms.

Attestation information originating from an Attester that is relayed via a Relying Party must be protected from replay or relay attacks, accordingly. In a closed ecosystem, trustworthiness with respect to the Attester can be achieved via a simply query to the Verifier. In an open ecosystem, the information conveyed in this interaction can include integrity measurements of every distinguishable software component that has been executed since its last boot cycle.

Commented [DT28]: I think you need to define what you mean by this term.

Commented [DT29]: Typo: simple

In the scope of RATS, this interaction encompasses the largest variety of information conveyed.

Claimant/Verifier: The intended operational state an Attester is intended to be in, is defined by the supply chain entities that manufacture and maintain the Attestor. In order to appraise trusted assertions or evidence conveyed by the Attester, every distinguishable system component the Attester is composed of can provide trusted assertions or evidence about its trustworthiness. A corresponding verifier that is tasked with assessing the trustworthiness of the Attester potentially requires a multitude of sources of reference values according to policies and the information provided. As Relying Parties often have to discover an appropriate Verifier, a Verifier has to obtain and potentially store appropriate reference values in order to asses assertions or evidence about trustworthiness.

Commented [DT30]: typo

Commented [DT31]: Unclear whether these are positioned as two different things, or one is a synonym for the other

Commented [DT32]: Typo: assess

Claimant/Attester: To enable RATS, trustworthy assertions have to be embedded in an Attester by its manufactorer. In some cases this involves various types of roots of trust. In other cases shielded pre-master secrets in combination with key derivation functions (KDF) provide this binding of trusted information to an Attester. A supply chain entity can embed additional trusted assertions to an Attester. These assertion can also be used to assert the trustworthiness on behalf of a separate RATS Actor or they can originate from an external entity (e.g. a security certification authority).

Commented [DT33]: Either I don't understand the definition of Claimant or I disagree with the paragraph here. To me, if the Attester generates evidence, sends that evidence to a server (which I'm guessing the term Claimant would apply to), gets back a token that it can supply to a Relying Party, then that would be Claimant/Attester interaction. And this is at runtime, not manufacturing time, and the server may be under the control of the device owner, not the device manufacturer and so limiting this discussion to be about what the manufacturer or supply chain does would be wrong.

Commented [DT34]: typo

Commented [DT35]: Undefined term (citation?)

Commented [DT36]: Fix grammar

5. Application of RATS

Attester are typically composite devices (in the case of atomically integrated devices that would result in a composite device with one component) or services. Services are software components - e.g. a daemon, a virtual network function (vnf) or a network security function (nsf) - that can reside on one or more Attester and are not necessarily bound to a specific set of hardware devices.

Commented [DT37]: Fix grammar

Commented [DT38]: Fix grammar

Relevant decision-factors that influence the composition of RATS Roles on RATS Actors, which result in specific work-flows are (amongst others):

- which RATS Role (or correspondingly, which RATS Actore that is taking on specific RATS roles) is triggering a Remote Attestation Procedure

Commented [DT39]: typo

- which entities are involved in a Remote Attestation Procedure (e.g. the Attester itself, trusted third parties, specific trust anchors, or other sources of assertions)

- the capabilities of the protocols used (e.g. challenge-response based, RESTful, or uni-directional)

- the security requirements and security capabilities of systems in a domain of application

- the risks and corresponding threats that are intended to be mitigated

5.1. Trust and Trustworthiness

[RFC4949] provides definitions that highlight the difference between a "trusted system" and a "trustworthy system". The following definitions exclude the explicit specialization of concepts that are "environmental disruption" as well as "human user and operator errors".

A trusted system in the context of RATS "operates as expected, according to design and policy, doing what is required and not doing other things" [RFC4949]. A trustworthy system is a system "that not only is trusted, but also warrants that trust because the system's behavior can be validated in some convincing way, such as through formal analysis or code review" [RFC4949].

The goal of RATS is to convey information about system component characteristics, such as integrity or authenticity, that can be appraised in a convincing way.

RATS require trust relationships with third parties that qualify assertions about, for example, origin of data, the manufacturer or the capabilities of a system, or the origination of attestation evidence (attestation provenance). Without trusted authorities (e.g. a certificate authority) it is virtually impossible to assess the level of assurance (or resulting level of confidence, correspondingly) of information produced by RATS. Trusting a system does not make it trustworthy. Assessing trustworthiness requires the conveyance of evidence that a system is a trustworthy system, which has to originate from the system itself and has to be convincing. If the convincing information is not originating from the system itself, it comprises trusted claim sets and not evidence. In essence, the attestation provenance of attestation evidence is the system that intends to present its trustworthiness in a believable manner.

The essential basis for trust in the information created via RATS are roots of trust.

Roots of trust are defined by the NIST special publication 800-164 draft as "security primitives composed of hardware, firmware and/or software that provide a set of trusted, security-critical functions. They must always behave in an expected manner because their misbehavior cannot be detected. As such, RoTs need to be secured by their design. Hardware RoTs are preferred over software RoTs due to their immutability, smaller attack surface, and more reliable behavior."

If the root of trust involved is a root of trust for measurement (RTM), the producer of information takes on the role of a asserter. An asserter can also make use of a root of trust for integrity (RTI) in order to increase the level of assurance in the assertions produced. If the root of trust involved is a root of trust for reporting (RTR), the producer of information takes on the role of an attester.

Commented [DT40]: typo “an”

5.2. Claims and Evidence

The RATS asserter role produces measurements about the system's characteristics in the form of signed (sometimes un-signed) claim sets in order to convey information. A secret signing key is required for this procedure, which is typically stored in a shielded location that can be trusted, for example, via a root of trust for storage (RTS).

The RATS attester role produces signed attestation evidence in order to convey information. The secret key required for this procedure is stored in a shielded location that only allows access to that key, if a specific operational state of the system is met. The trust with respect to this origination is based on a root of trust for reporting.

5.3. RATS Information Flows

There are six roles defined in the RATS architecture. iFigure 2 provides a simplified overview of the RATS Roles defined above,

Commented [DT41]: typo

illustrating a general Interconnect in the center that facilitates all RATS Interactions.

```
          +------------+                     +------------------+
          |            |                     |                  |
          |  Attester  |                  +->|  Verifier        |
          |            |                  |  |                  |
          +------------+                  |  +------------------+
                ^                         |
                |                         |
                |     +----------------+  |
                +---->|                |<-+
                      |  Interconnect  |
                +---->|                |<-+
                |     +----------------+  |
                v                         |
          +------------+                  |  +------------------+
          |            |                  |  |                  |
          |  Claimant  |                  +->|  Relying Party   |
          |            |                     |                  |
          +------------+                     +------------------+
```

Figure 2: Overall Relationships of Roles in the RATS Architecture

6. Exemplary Composition of Roles

In order to provide an intuitive understanding how the roles used in RATS can be composed into work-flows, this document provides a few example work-flows. Boxes in the following examples that include more than one role are systems that take on more than one role.

6.1. Conveyance of Trusted Claim Sets Validated by Signature

If there is a trust relationship between a trusted third party that can assert that signed claims created by a claimant guarantee a trustworthy origination of claim, the work-flow depicted in Figure 3 can facilitate a trust-based implicit remote attestation procedure. The information conveyed are signed claim sets that are trusted via an authoritative third party. In this work-flow claim emission is triggered by the claimant. Variations based on requests emitted by the relying party can be easily facilitated by the same set of roles.

Commented [DT42]: Neither of the diagrams below match the relationships I'm most familiar with, so I can't tell whether the definitions are confusing or the pictures don't apply to me.

In the pictures I would draw, there's Attester<->Relying Party communication.
On the Attester side, the Attester may or may not (depending on use case/protocol implementation) first hit a server (maybe called a Claimant?) to get a token that it includes in its communication with the RP. That is, either the communication to the RP includes evidence signed by the Attester's own RoT, or it sends such evidence up to its server, and gets a signed statement that roots to the server.
On the RP side, the RP may or may not (depending on use case/protocol implementation) first hit a server (maybe called a Verifier?) to convert whatever the server supplies, into a statement signed by its own server, and use that for local decision making. That is, the Verifier may be local or remote, but in either case, the RP is acting on information from an Attester.

```
                                                  +-----------+
            +------------+  +----------------+    |           |
            |            |  |                |    |  Relying  |
            |  Claimant  |->|  Interconnect  |--->|  Party    |
            |            |  |                |    |           |
            +------------+  +----------------+    +-----------+
```

Figure 3: Conveyance of Trusted Claim Sets Validated by Signature

6.2. Conveyance of Attestation Evidence Appraised by a Verifier

If there is trust in the root of trust for reporting based on the assertions of a trusted third party, the work-flow depicted in Figure 4 can facilitate an evidence-based explicit remote attestation procedure. The information conveyed is signed attestation evidence that is created by the trusted verifier. In this work-flow claims do not necessarily have to be signed and the work-flow is triggered by the attestor that aggregates claims from a root of trust of measurement. Variations based on requests emitted by the verifier can be easily facilitated by the same set of roles.

Commented [DT43]: typo

```
    +------------------+                      +----------------------+
    |                  |                      |  +----------------+  |
    |  +------------+  |  +----------------+  |  |                |  |
    |  |            |  |  |                |  |  |                |  |
    |  |  Attester  |--+->|  Interconnect  |--+->|   Verifier     |  |
    |  |            |  |  |                |  |  |                |  |
    |  +------------+  |  +----------------+  |  +----------------+  |
    |        ^         |                      |          |           |
    |        |         |                      |          v           |
    |        |         |                      |   +-----------+      |
    |  +-----+------+  |                      |   |           |      |
    |  |            |  |                      |   |  Relying  |      |
    |  |  Claimant  |  |                      |   |  Party    |      |
    |  |            |  |                      |   |           |      |
    |  +------------+  |                      |   +-----------+      |
    |                  |                      |                      |
    +------------------+                      +----------------------+
```

Figure 4: Conveyance of Attestation Evidence Appraised by a Verifier

7. The Scope of RATS

During its evolution, the term Remote Attestation has been used in multiple contexts and multiple scopes and in consequence accumulated various connotations with slightly different semantic meaning.

Correspondingly, Remote Attestation Procedures (RATS) are employed in various usage scenarios and different environments.

In order to better understand and grasp the intent and meaning of specific RATS in the scope of the security area - including the requirements that are addressed by them - this document provides an overview of existing work, its background, and common terminology. As the contribution, from that state-of-the-art a set of terms that provides a stable basis for future work on RATS in the IETF is derived.

In essence, a prerequisite for providing an adequate set of terms and definitions for the RATS architecute is a general understanding and a common definitions of "what" RATS can accomplish "how" RATS can to be used.

Commented [DT44]: typo

Please note that this section is still missing various references and is considered "under construction". The majority of definitions is still only originating from IETF work. Future iterations will pull in more complementary definitions from other SDO (e.g. Global Platform, TCG, etc.) and a general structure template to highlight semantic relationships and capable of resolving potential discrepancies will be introduced. A section of context awareness will provide further insight on how Attestation procedures are vital to ongoing work in the IETF (e.g. I2NSF & tokbind). The definitions in the section about RATS are still self-describing in this version. Additional explanatory text will be added to provide more context and coherence.

Commented [DT45]: fix grammar

7.1. The Lying Endpoint Problem

A very prominent goal of RATS is to address the "lying endpoint problem". The lying endpoint problem is characterized as a condition of a Computing Context where the information or behavior embedded, created, relayed, stored, or emitted by the Computing Context is not "correct" according to expectations of the authorized system designers, operators and users. There can be multiple reasons why these expectations are incorrect, either from malicious Activity, unanticipated conditions or accidental means. The observed behavior, nevertheless, appears to be a compromised Computing Context.

Commented [DT46]: Term is not defined until 8.1. Don't use undefined terms.

Commented [DT47]: I think it's important to call out that the fact that some information is wrong does not necessarily mean that the device has been compromised, only that that piece of information is wrong. For example, since location is not an inherent aspect of a device, it has to get the location from somewhere else. If it gets the wrong information, and then reports that's what it knows, it is not compromised per se, it's just reporting incorrect information it was given. So is repeating incorrect information, without knowing it's incorrect, really called "lying"?

- Attempts to "scrub" the data or "proxy" control elements implies the existence of a more fundamental trusted endpoint that is operating correctly. Therefore, Remote Attestation - the technology designed to detect and mitigate the "lying endpoint problem" - must be trusted to behave correctly independent of other controls.

Consequently, a "lying endpoint" cannot also be a "trusted system".

Remote Attestation procedures are intended to enable the consumer of information emitted by a Computing Context to assess the validity and integrity of the information transferred. The approach is based, for example, on the assumption that if attestation evidence can be provided in order to prove the integrity of every software instance installed involved in the activity of creating the emitted information in question, the emitted information can be considered valid and integer.

Commented [DT48]: This is exactly why extrinsic information like location has to be treated differently, since the creation of the value is not done in the device/service that's being attested.

In contrast, such Evidence has to be impossible to create if the software instances used in a Computing Context are compromised. Attestation activities that are intended to create this Evidence therefore also provide guarantees about the validity of the Evidence they can create.

7.1.1. How the RATS Architecture Addresses the Lying Endpoint Problem

RATS imply the involvement of at least two players (roles) who seek to overcome the lying endpoint problem. The Verifier wishes to consume application data supplied by a Computing Context. But before application data is consumed, the Verifier obtains Attestation Evidence about the Computing Context to assess likelihood of poisoned data due to endpoint compromise or failure. Remote Attestation argues that a systems's integrity characteristics should not be believed until rationale for believability is presented to the relying party seeking to interact with the system.

Commented [DT49]: typo

An Interconnect defines an untrusted channel between subject and object wherein the rationale for believability is securely exchanged. The type of interconnect technology could vary widely, ranging from GPIO pins, to a PC peripheral IO bus, to the Internet, to a direct physical connection, to a wireless radio-receiver association, or to a world wide mesh of peers. In other words, virtually every kind communication path could be used as the "Interconnect" in RATS. In fact, a single party could take on all roles at the same time (e.g. Self Encrypting Devices).

Commented [DT50]: I think the use of "the" (and similarly the depiction in Figure 4) is bad in that it implies that the same interconnect technology is used for all communication between all roles. This is not the case.

Attestation evidence can be thought of as the topics of the exchange that is created the operational primitives of a root of trust for reporting. Evidence may be structured in an interoperable format called claims that may include references to the claimants which are asserting the claims. RATS aims to define "interoperable Remote Attestation" such that evidence can be created and consumed by different ecosystem systems and can be securely exchanged by a broad set of network protocols.

8. RATS Terminology

This document relies on terminology found in [RFC4949]. This document presumes the reader is familiar with the following terms.

- Cryptography
- Entity (System entity)
- Identity
- Object
- Principal
- Proof-of-possession protocol
- Security environment (Environment)
- Security perimeter
- Subject
- Subsystem
- System
- Target-of-Evaluation (TOE)
- Trusted Computing Base (TCB)
- Trusted Platform Module (TPM)
- Trusted (Trustworthy) system
- Verification

Terminology defined by this document is preceded by a dollar sign ($) to distinguish it from terms defined elsewhere and as a way to disambiguate term definition from explanatory text.

Terms defined by this document that are subsequently used by this document are distinguished by capitalizing the first letter of the term (e.g. Term or First_word Second_word).

8.1. Computing Context

This section introduces the term Computing Context in order to specialize the notions of environment and endpoint to terminology that has relevance to trusted computing. Attestation is a discipline of trusted computing.

A Computing Context could refer to a large variety of endpoints. Examples include but are not limited to: the compartmentalization of physical resources, the separation of software instances with different dependencies in dedicated containers, and the nesting of virtual components via hardware-based and software-based solutions. The number of approaches and techniques to construct an endpoint continuously changes with new innovation. Hence, it isn't a goal of this document to define remote attestation for a fixed set of endpoints. Rather, it attempts to define endpoints conceptually and rely on Claims management as a way to clarify the details and specific attributes of conceptual endpoints.

Computing Contexts may be recursive in nature in that it could be composed of a system that is itself a composite of subsystems. In consequence, a system may be composed of other systems that may be further composed of one or more Computing Contexts capable of taking on the RATS roles. The scope and application of these roles can range from:

o Continuous mutual Attestation procedures of every subsystem inside a composite device, to

o Sporadic Remote Attestation of unknown parties via heterogeneous Interconnects.

Analogously, the increasing number of features and functions that constitute components of a device start to blur the lines that are required to categorize each solution and approach precisely. To address this increasingly challenging categorization, the term Computing Context defines the characteristics of the (sub)systems that can take on the role of an Attester and/or the role of a Verifier. This approach is intended to provide a stable basis of definitions for future solutions that continuous to remain viable long-term.

$ Computing Context : An umbrella term that combines the scope of the definitions of endpoint [ref NEA], device [ref 1ar], and thing [ref t2trg], including hardware-based and software-based sub-contexts that constitute independent, isolated and distinguishable slices of a Computing Context created by compartmentalization mechanisms, such as Trusted Execution Environments (TEE), Hardware Security Modules (HSM) or Virtual Network Function (VNF) contexts.

8.1.1. Characteristics of a Computing Context

While the semantic relationships highlighted above constitute the fundamental basis to provide a define Computing Context, the following list of object characteristics is intended to improve the application of the term and provide a better understanding of its meaning:

$ Computing Context Characteristics: A representation of the identity, composition, configuration and state of a Computing Context.

Computing context characteristics provide the following: * An independent environment in regard to executing and running software, * An isolated control plane state (by potentially interacting with other Computing Contexts), * A dedicated management interface by which control plane behavior can be effected, * Unique identification towards reliable disambiguation within a given scope.

Commented [DT51]: Formatting problem

Computing context characteristics do not necessarily include a network interface with associated network addresses (as required by the definition of an endpoint) - although it is very likely to have (access to) one.

[Issue: This conclusion could be incorrect] In contrast, a container [ref docker, find a more general term here] context is not a distinguishable isolated slice of an information system and therefore is not an independent Computing Context. [more feedback on this statement is required as the capabilities of docker-like functions evolve continuously]

Commented [DT52]: Yes I think this is incorrect and this paragraph should just be deleted

Examples include: a smart phone, a nested virtual machine, a virtualized firewall function running distributed on a cluster of physical and virtual nodes, or a trust-zone.

Commented [DT53]: Undefined term

8.1.2. Computing Context Semantic Relationships

Computing Contexts may relate to other Computing Contexts that are decomposable in a variety of ways.

o Singleton,

o Tuples (e.g. 2-tuple, n-tuple),

- Nested,

Commented [DT54]: Is an app running on an OS considered to be a “nested” relationship? (Personally, I find this bullet list to be more confusing than helpful)

- Clustered (homogeneous),

- Grouped (heterogenous).

The scope of Computing Context encompasses a broad spectrum of systems including, but not limited to:

- An information system,

- An object,

- A composition of objects,

- A system component,

- A system sub-component,

- A composition of system sub-components,

- A system entity,

- A composition of system entities.

A Computing Context may be realized in a variety of ways including, but not limited to:

- A process, thread or task as defined by an operating system,

- A privileged operating system task, interrupt handler or event handler,

- A virtual machine,

- A virtual machine monitor,

- A processor mode (e.g. system management mode),

- A co-processor,

- A peripheral device,

- A secure element,

- A trusted execution environment,

- A controller, sensor, actutor, switch, router or gateway,

Commented [DT55]: typo

- An FPGA,
- An ASIC,
- A memory resource,
- A storage resource.

Analogously, a computing sub-context is a decomposition of a Computing Context; a subsystem is a decomposition of a system; a sub-component is a decomposition of a component; and a peer node is a decomposition of a node cluster.

A formal semantic relationship is therefore expressed using an information model that captures interactions, relationships, bindings and interfaces among systems, subsystems, system components, system entities or objects.

[Issue: A tangible relationship to an information model is required here] An information model that richly captures Computing Context semantics is therefore believed to be relevant if not fundamental to Remote Attestation.

8.1.3. Computing Context Identity

The identity of a Computing Context implies there is a binding operation between an identifier and the Computing Context.

$ Computing Context Identity: Computing Context Identity provides the basis for associating attestation Evidence about a particular Computing Context to create believable knowledge about attestation provenance.

Confidence in the identity assurance level [NIST SP-800-63-3] or the assurance levels for identity authentication [RFC4949] is a property of the identifier uniqueness properties and binding operation veracity. Such properties impact the trustworthiness of associated attestation Evidence.

8.2. Remote Attestation Concepts

Attestation Evidence created by RATS is a form of telemetry about a computing environment that enables better security risk management through disclosure of security properties of the environment. Attestation may be performed locally (within the same computing environment) or remotely (between different computing environments). The exchange of attestation evidence can be formalized to include well-defined protocol, message syntax and semantics.

8.3. Core RATS Terminology

Commented [DT56]: This section needs to be way earlier in the document. That is, define terms before first use in the doc.

$ Attestation: The creation of evidence by the Attester based on measurements or other claimant output.

A form of telemetry involving the delivery of Claims describing various security properties of a Computing Context by an Attester, such that the Claims can be used as Evidence toward convincing a Verifier regarding trustworthiness of the Computing Context.

Commented [DT57]: This looks like a definition, but for what term?

$ Conveyance: The transfer of Evidence from the Attester to the Verifier.

$ Verification: The appraisal of Evidence by the Verifier who evaluates it against a reference policy. See also RFC4949 [1].

$ Remote Attestation: A procedure involving Attestation, Conveyance and Verification.

8.4. RATS Information Model Terminology

Evidence conveyed to a Verifier by an Attester is structured to facilitate syntactic and semantic interoperability. An information model defines the tag namespaces used to create tag-value pairs containing discrete bits of Evidence.

$ Evidence: A set of Measurements, quality metrics, quality procedures or assurance criteria about an Computing Context's behavioral, operational and intrinsic characteristics.

$ Claim: Structured Evidence asserted about a Computing Context. It contains metadata that informs regarding the type, class, representation and semantics of Evidence information. A Claim is represented as a name-value pair consisting of a Claim Name and a Claim Value [RFC7519]. In the context of SACM, a Claim is also specialized as an attribute-value pair that is intended to be related to a statement [I-D.ietf-sacm-terminology].

Commented [DT58]: Does evidence consist of a set of claims? Or does a claim consist of a set of evidence? I have a hard time understanding the difference without an example.

$ Attestable Claim: Structured Evidence including one or more Claims that are asserted by a Claimant (Note: an Attester role doubles as a Claimant role). An Attestable Claim has the following structure:

Commented [DT59]: I'm still confused about this role and so I can't follow what is meant in this section

1. A Claim or Claims.

2. A Claimant identity.

3. Proof of Claimant identity.

4. Proof the Claimant intended to make these Claims.

Note: Proofs of Claims assertions may be separated from the Claim itself. For example, a secure transport over which Claims are conveyed where Claimant's signing key integrity protects the transport payload could be used as proof of Claim assertion. Alternatively, each Claim could be separately signed by a Claimant.

$ Attested (Asserted) Claim: An Attestable Claim where the proof elements are populated.

$ Evidence (Claims) Creation: Instantiation of Attested Claims by a Claimant.

$ Evidence (Claims) Collection: Assembling of Attested Claims by an Attester for the purpose of Conveyance.

$ Verified (Valid) Claim: An Attested Claim where the proof elements have been verified by a Verifier according to a policy that identifies trusted Claimants and/or trusted Evidence values.

8.5. RATS Work-Flow Terminology

This section introduces terms and definitions that are required to illustrate the scope and the granularity of RATS workflows in the domain of security automation. Terms defined in the following sections will be based on this workflow-related definitions.

In general, RATS are composed of iterative activities that can be conducted in intervals. It is neither a generic set of actions nor simply a task, because the actual actions to be conducted by RATS can vary significantly depending on the protocols employed and types of Computing Contexts involved.

$ Activity: A sequence of actions conducted by Computing Contexts that compose a Remote Attestation procedure. The actual composition of actions can vary, depending on the characteristics of the Computing Context they are conducted by/in and the protocols used to utilize an Interconnect. A single Activity provides only a minimal amount of semantic context, e.g.defined by the Activity's requirements imposed upon the Computing Context, or via the set of actions it is composed of. Example: The Conveyance of cryptographic Evidence or the appraisal of Evidence via imperative guidance.

Commented [DT60]: This term is confusing. Why do we need it at all? I think the architecture could be described much more simply with fewer terms, without loss of flexibility. Activity vs Task vs Action seems like gratuitous confusion, to me.

Commented [DT61]: space

$ Task: A unit of work to be done or undertaken.

In the scope of RATS, a task is a procedure to be conducted. Example: A Verifier can be tasked with the appraisal of Evidence originating from a specific type of Computing Contexts providing appropriate identities.

$ Action: The accomplishment of a thing usually over a period of time, in stages, or with the possibility of repetition.

In the scope of RATS, an action is the execution of an operation or function in the scope of an Activity conducted by a Computing Context. A single action provides no semantic context by itself, although it can limit potential semantic contexts of RATS to a specific scope. Example: Signing an existing public key via a specific openssl library, transmitting data, or receiving data are actions.

$ Procedure: A series of actions that are done in a certain way or order.

In the scope of RATS, a procedure is a composition of activities (sequences of actions) that is intended to create a well specified result with a well established semantic context. Example: The activities of Attestation, Conveyance and Verification compose a Remote Attestation procedure.

8.6. RATS Reference Use Cases

Commented [DT62]: Why do you need this section if this document already references MCR's use cases document? Shouldn't use cases just be in one document, not split/duplicated in two?

A "lying endpoint" is not trustworthy.

This document provides NNN prominent examples of use cases Attestation procedures are intended to address:

- Verification of the source integrity of a Computing Context via data integrity proofing of installed software instances that are executed, and

- Verification of the identity proofing of a Computing Context.

8.6.1. Use Case A

8.6.2. Use Case B

8.7. RATS Reference Terminology

$ Attestable Computing Context: A Computing Context where a Claimant is able to create Claims, an Attester is able to Attest those Claims and a Verifier is able to verify the Claims.

$ Attestation Identity: An identity that refers to an Attester.

$ Attestation Identity Credential: A credential used to authenticate an Attestation Identity.

$ Attestation Identity Key (AIK): An Attestation Identity Credential in the form of an asymmetric cryptographic key where the AIK private key is protected by a Computing Context with protection properties that are stronger than the Computing Context about which the AIK attests. A root-of-trust Computing Context normally protects AIK private keys.

$ Claimant Identity: An identity that refers to an Claimant.

Commented [DT63]: typo

$ Claimant Identity Credential: A credential used to authenticate a Claimant Identity.

$ Measurements / Integrity Measurements: Metrics of Computing Context characteristics (i.e. composition, configuration and state) that affect the confidence in the trustworthiness of a Computing Context. Digests of integrity Measurements can be stored in shielded locations (e.g. a PCR of a TPM).

$ Reference Integrity Measurements: Signed Measurements about a Computing Context's characteristics that are provided by a vendor or manufacturer and are intended to be used as declarative guidannce [I-D.ietf-sacm-terminology] (e.g. a signed CoSWID).

$ Root-of-trust: The Computing Context that protects the following where no other Computing Context is expected to provide its Attestation Evidence: + Attestation Evidence. + AIKs. + Code used during the collection and reporting of Attestation Evidence.

Commented [DT64]: Looks like a formatting error

$ Root-of-trust-for-measurement (RTM): A trusted Computing Context where a Claimant creates integrity Measurements and other Evidence about a Computing Context where no other Computing Context is expected to provide its Attestation Evidence.

$ Root-of-trust-for-reporting (RTR): A trusted Computing Context where an Attester stages reporting of Claims where no other Computing Context is expected to provide its Attestation Evidence.

$ Root-of-trust-for-storage (RTS): A trusted Computing Context where a Claimaint or Attester stores Claims, Evidence, credentials or policies associated with Attestation where no other Computing Context is expected to provide its Attestation Evidence.

Commented [DT65]: typo

Commented [DT66]: I think this should also say "or Verifier" since the policy (e.g., reference values) typically also need to be protected by a root of trust.

$ Trustworthy Computing Context: A Computing Context that guarantees trustworthy behavior and/or composition (with respect to certain declarative guidance and a scope of confidence). A trustworthy Computing Context is a trustworthy system.

<NMS: is this necessary?> Trustworthy Statement: Evidence conveyed by a Computing Context that is not necessarily trustworthy. [update with tamper related terms]

Commented [DT67]: Can't understand this

8.8. Interpretations of RFC4949 Terminology for Attestation

Commented [DT68]: Personally, I find this section too wordy to go in an architecture document.

Assurance: An attribute of an information system that provides grounds for having confidence that the system operates such that the system's security policy is enforced [RFC4949] (see Trusted System below).

In common criteria, assurance is the basis for the metric level of assurance, which represents the "confidence that a system's principal security features are reliably implemented".

Commented [DT69]: This phrase sounds like a meaningless tautology due to the apparently circular statement.

The NIST Handbook [get ref from 4949] notes that the levels of assurance defined in Common Criteria represent "a degree of confidence, not a true measure of how secure the system actually is. This distinction is necessary because it is extremely difficult-and in many cases, virtually impossible-to know exactly how secure a system is."

Historically, assurance was well-defined in the Orange Book [http://csrc.nist.gov/publications/history/dod85.pdf] as "guaranteeing or providing confidence that the security policy has been implemented correctly and that the protection-relevant elements of the system do, indeed, accurately mediate and enforce the intent of that policy. By extension, assurance must include a guarantee that the trusted portion of the system works only as intended."

Confidence: The definition of correctness integrity in [RFC4949] notes that "source integrity refers to confidence in data values". Hence, confidence in an Attestation procedure is referring to the degree of trustworthiness of an Attestation Activity that produces Evidence (Attester), of an Conveyance Activity that transfers Evidence (interconnect), and of a Verification Activity that appraises Evidence (Verifier), in respect to correctness integrity.

Commented [DT70]: typo

Correctness: The property of a system that is guaranteed as the result of formal Verification activities.

Correctness integrity: The property that the information represented by data is accurate and consistent.

Data Integrity: (a) The property that data has not been changed, destroyed, or lost in an unauthorized or accidental manner. (See: data integrity service. Compare: correctness integrity, source integrity.)

(b) The property that information has not been modified or destroyed in an unauthorized manner.

Entity: A principal, Subject, relying party or stake holder in an Attestation ecosystem.

Identity: The set of attributes that distinguishes a principal.

Identifier: The set of attributes that distinguishes an object.

Identity Proofing: A vetting process that verifies the information used to establish the identity of a system entity.

(Information) System: An organized assembly of computing and communication resources and procedures - i.e., equipment and services, together with their supporting infrastructure, facilities, and personnel - that create, collect, record, process, store, transport, retrieve, display, disseminate, control, or dispose of information to accomplish a specified set of functions.

Object: A system component that contains or receives information.

Source Integrity: The property that data is trustworthy (i.e., worthy of reliance or trust), based on the trustworthiness of its sources and the trustworthiness of any procedures used for handling data in the system.

Subject: A Computing Context acting in accordance with the interests of a principal.

Subsystem: A collection of related system components that together perform a system function or deliver a system service.

System Component: An instance of a system resource that (a) forms a physical or logical part of the system, (b) has specified functions and interfaces, and (c) is extant (e.g., by policies or specifications) outside of other parts of the system. (See: subsystem.)

An identifiable and self-contained part of a $Target-of-Evaluation.

Token: A data structure suitable for containing Claims.

Trusted (Trustworthy) System: A system that operates as expected, according to design and policy, doing what is required - despite environmental disruption, human user and operator errors, and attacks by hostile parties - and not doing other things.

Verification: (a) The process of examining information to establish the truth of a claimed fact or value.

(b) The process of comparing two levels of system specification for proper correspondence, such as comparing a security model with a top-level specification, a top-level specification with source code, or source code with object code.

8.9. Building Block Vocabulary (Not in RFC4949)

[working title, pulled from various sources, vital]

Attribute: TBD

Characteristic: TBD

Context: TBD

Endpoint: TBD

Environment: TBD

Manifest: TBD

Telemetry: An automated communications process by which data, readings, Measurements and Evidence are collected at remote points and transmitted to receiving equipment for monitoring and analysis. Derived from the Greek roots tele = remote, and metron = measure.

Commented [DT71]: No idea why this sentence is relevant to this document.

9. IANA considerations

This document will include requests to IANA:

- o first item
- o second item

10. Security Considerations

There are always some.

11. Acknowledgements

Maybe.

12. Change Log

No changes yet.

13. References

13.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

13.2. Informative References

[I-D.ietf-sacm-terminology] Birkholz, H., Lu, J., Strassner, J., Cam-Winget, N., and A. Montville, "Security Automation and Continuous Monitoring (SACM) Terminology", draft-ietf-sacm-terminology-16 (work in progress), December 2018.

[I-D.mandyam-eat] Mandyam, G., Lundblade, L., Lundblade, L., Ballesteros, M., and J. O'Donoghue, "The Entity Attestation Token (EAT)", draft-mandyam-eat-01 (work in progress), November 2018.

[I-D.richardson-rats-usecases] Richardson, M., "Use cases for Remote Attestation common encodings", draft-richardson-rats-usecases-00 (work in progress), March 2019.

[I-D.tschofenig-rats-psa-token]
Tschofenig, H., Frost, S., Brossard, M., and A. Shaw, "Arm's Platform Security Architecture (PSA) Attestation Token", draft-tschofenig-rats-psa-token-00 (work in progress), March 2019.

[RFC4949] Shirey, R., "Internet Security Glossary, Version 2", FYI 36, RFC 4949, DOI 10.17487/RFC4949, August 2007, <https://www.rfc-editor.org/info/rfc4949>.

[RFC7519] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015, <https://www.rfc-editor.org/info/rfc7519>.

13.3. URIs

[1] https://tools.ietf.org/html/rfc4949

Authors' Addresses

Henk Birkholz
Fraunhofer SIT
Rheinstrasse 75
Darmstadt 64295
Germany

Email: henk.birkholz@sit.fraunhofer.de

Monty Wiseman
GE Global Research
USA

Email: monty.wiseman@ge.com

Hannes Tschofenig
ARM Ltd.
110 Fulbourn Rd
Cambridge CB1 9NJ
UK

Email: hannes.tschofenig@gmx.net

Ned Smith
Intel Corporation
USA

Email: ned.smith@intel.com