

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 27, 2019

H. Birkholz
Fraunhofer SIT
M. Wiseman
GE Global Research
H. Tschofenig
ARM Ltd.
N. Smith
Intel
October 24, 2018

Architecture and Reference Terminology for Remote Attestation Procedures
draft-birkholz-rats-architecture-00

Abstract

Remote ATtestation ProcedureS (RATS), such as Remote Integrity VERification (RIVER), the creation of Entity Attestation Tokens (EAT), software integrity Measurement And ATtestation (MAAT), or the attestation of device characteristics, in general, are based on specific operations and qualities provided by hardware and software. The RATS architecture maps corresponding functions and operation capabilities to specific RATS roles. The goal is to enable an appropriate conveyance of believable evidence about device health or trusted claims about device capabilities via network protocols. The flows of data between these roles depend on the composition of RATS roles and their location with respect to devices or services. The RATS architecture provides these roles as building blocks to enable suitable composition, while remaining hardware-agnostic. This flexibility is intended to address a significant majority of use cases or usage scenarios in the domain of RATS. Examples include, but are not limited to: financial transactions, voting machines, critical safety systems, network equipment health, or trustworthy end-user device management.

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 working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any

time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 27, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](https://trustee.ietf.org/license-info) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

- 1. Introduction 3
- 1.1. Requirements notation 3
- 2. RATS Architecture 3
- 2.1. Roles, Devices, and Services 4
- 2.2. Trust and Trustworthiness 5
- 2.3. Claims and Evidence 6
- 2.4. RATS Roles 6
- 2.5. Exemplary Composition of Roles 8
- 2.5.1. Conveyance of Trusted Claim Sets Validated by Signature 8
- 2.5.2. Conveyance of Attestation Evidence Appraised by a Verifier 9
- 2.6. The Scope of RATS 9
- 2.6.1. The Lying Endpoint Problem 10
- 2.6.2. How the RATS Architecture Addresses the Lying Endpoint Problem 11
- 3. RATS Terminology 11
- 3.1. Computing Context 12
- 3.1.1. Characteristics of a Computing Context 13
- 3.1.2. Computing Context Semantic Relationships 14
- 3.1.3. Computing Context Identity 16
- 3.2. Remote Attestation Concepts 16
- 3.3. Core RATS Terminology 16
- 3.4. RATS Information Model Terminology 17
- 3.5. RATS Work-Flow Terminology 18
- 3.6. RATS Reference Use Cases 19

3.6.1. Use Case A 19

3.6.2. Use Case B 19

3.7. RATS Reference Terminology 19

3.8. Interpretations of RFC4949 Terminology for Attestation . 21

3.9. Building Block Vocabulary (Not in RFC4949) 23

4. IANA considerations 23

5. Security Considerations 23

6. Acknowledgements 23

7. Change Log 24

8. References 24

8.1. Normative References 24

8.2. Informative References 24

8.3. URIs 24

Authors' Addresses 24

1. Introduction

This document provides normative guidance how to use, create or adopt network protocols that facilitate remote attestation procedures. The foundation of the RATS architecture are specific roles that can be chained and as a result compose remote attestation procedures. The term attestation, unfortunately, is an overloaded term. There are different interpretations, connotations and meanings to the term attestation and therefore also to terms related to attestation. In consequence, this document also provides a detailed definition of Attestation Terminology. The intent is to illustrate and remediate the impedance mismatch of terms related to Remote Attestation Procedures used in different domains today. New terms defined by this document provide a consolidated basis to support future work on RATS in the IETF and beyond.

1.1. 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 RFC 2119, BCP 14 [RFC2119].

2. RATS Architecture

The goal of the RATS architecture is to provide the building blocks - the roles defined by the RATS architecture - to enable the composition of service-chains and work-flows to create and appraise evidence about the trustworthiness of devices and services.

The RATS architecture does not prescribe specific payload definitions, role composition, or protocol use. However, it imposes requirements on payload definitions, interfaces, and network

protocols with respect to proofs of freshness, attestation provenance, and required operational primitives that are available to devices and services taking on RATS roles. For brevity, the term "system" is a synonym for "device or service" in this document.

2.1. Roles, Devices, and Services

In the RATS architecture, devices or services can take on RATS roles. In this context, devices are typically composite devices (in the case of atomically integrated devices that would result in a composite device with one component). 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 devices and are not necessarily bound to a specific set of devices.

Devices or Services (Systems) can take on one or more RATS roles either by separate functions or via a collapsed functions that take on multiple RATS roles. Systems that take on RATS roles:

- o are consumer and/or producer of role-specific information, and
- o can be chained to compose specific work-flows.

Systems can be distinguished on the management plane via identity documents (which includes specific claim sets about device characteristics, see [RFC4949](#)) or via trusted claim sets (e.g. the Entity Attestation Token) and can be addressed by network protocols via IP addresses. RATS can be used in environments without network protocols and RATS roles can be used to design work-flows in these domains, correspondingly. However, the primary focus of the RATS architecture is to facilitate network protocols between RATS roles that convey information via the Internet Protocol.

Relevant decision-factors that influence the composition of RATS roles on systems and resulting work-flows are (amongst others):

- o which role (or correspondingly, which system that is taking on specific RATS roles) is triggering a Remote Attestation Procedure
- o 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)
- o the capabilities of the protocols used (e.g. challenge-response based, RESTful, uni-directional)
- o the security requirements and security capabilities of systems in a domain of application

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

2.2. 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.

2.3. 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.

2.4. RATS Roles

There are six roles defined in the RATS architecture. iFigure 1 provides a simplified overview of the roles defined in this section.

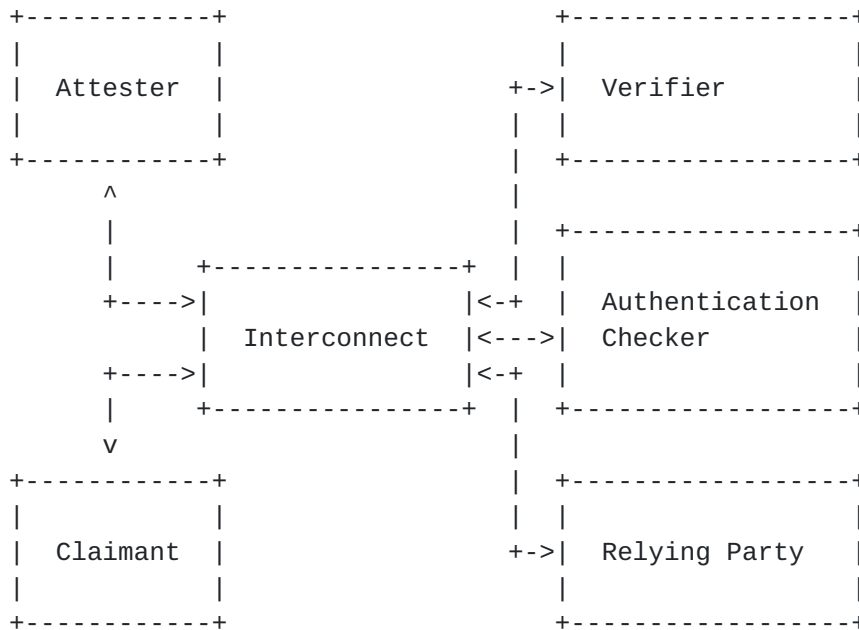


Figure 1: Overall Relationships of Roles in the RATS Architecture

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.

Claimant: The producer of measurements or assertions to certain properties regarding the trustworthiness of a system's characteristics that has a root of trust for measurement. It is not guaranteed that a verifier can appraise the output of a claimant via reference values. Examples of claim output include: the binding of an attester to an RTR, GPS coordinates set of integrity measurements, or an Universal Entity ID (UEID).

Interconnect: A communication channel or secure path between systems that take on RATS roles. Attestation evidence, for example, can be conveyed from an attester to a verifier via an interconnect. Examples include: GPIO pins, an USB link, or the Internet.

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.

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 and implements a conveyance protocol, appraises attestation evidence against reference values or policies and makes verification results available to relying parties.

2.5. 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.

2.5.1. 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 2 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.

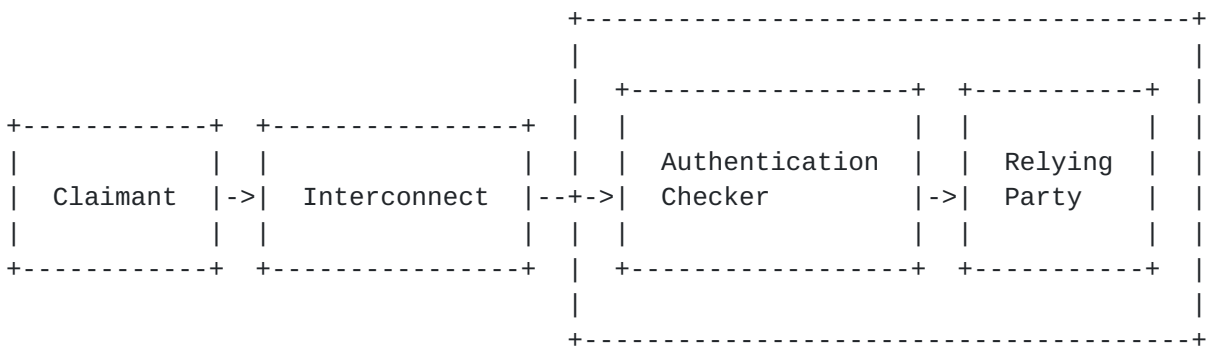


Figure 2: Conveyance of Trusted Claim Sets Validated by Signature

2.5.2. 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 3 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.

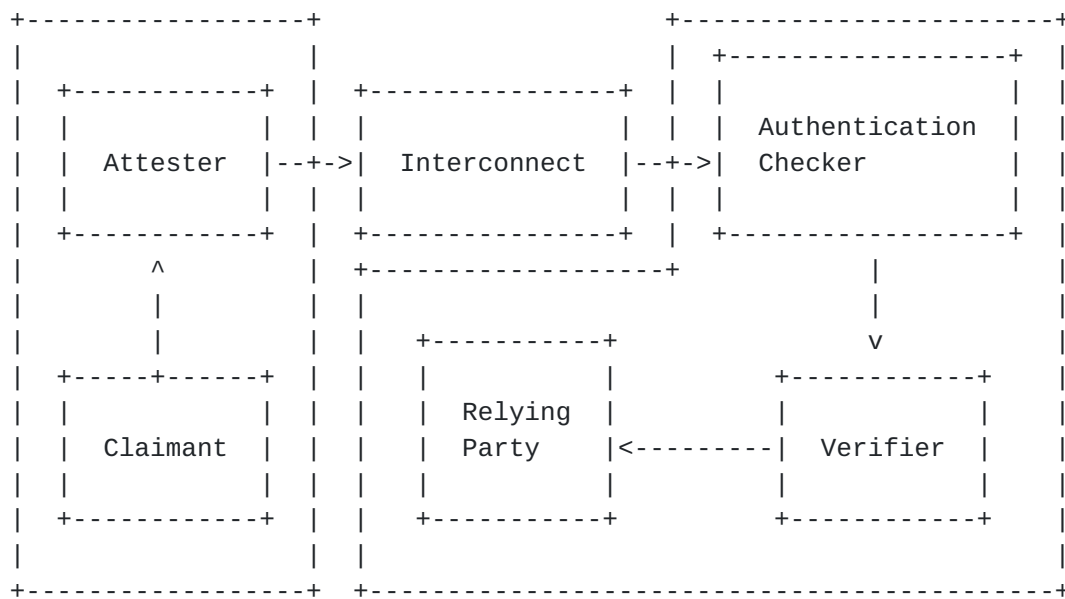


Figure 3: Conveyance of Attestation Evidence Appraised by a Verifier

2.6. 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 architecture is a general understanding and a common definitions of "what" RATS can accomplish "how" RATS can to be used.

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.

2.6.1. 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.

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.

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.

2.6.2. 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.

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).

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.

3. RATS Terminology

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

- o Cryptography
- o Entity (System entity)
- o Identity

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

3.1. 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.

3.1.1. 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.

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]

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.

3.1.2. 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),
- o Nested,
- o Clustered (homogeneous),
- o Grouped (heterogenous).

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

- o An information system,
- o An object,
- o A composition of objects,

- o A system component,
- o A system sub-component,
- o A composition of system sub-components,
- o A system entity,
- o A composition of system entities.

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

- o A process, thread or task as defined by an operating system,
- o A privileged operating system task, interrupt handler or event handler,
- o A virtual machine,
- o A virtual machine monitor,
- o A processor mode (e.g. system management mode),
- o A co-processor,
- o A peripheral device,
- o A secure element,
- o A trusted execution environment,
- o A controller, sensor, actuator, switch, router or gateway,
- o An FPGA,
- o An ASIC,
- o A memory resource,
- o 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.

3.1.3. 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.

3.2. 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.

3.3. Core RATS Terminology

\$ 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.

\$ 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.

3.4. 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](#)].

\$ 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:

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.

3.5. 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.

\$ 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.

3.6. RATS Reference Use Cases

A "lying endpoint" is not trustworthy.

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

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

3.6.1. Use Case A

3.6.2. Use Case B

3.7. 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.
 - \$ 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 guidance [[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.
 - \$ 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 Claimant or Attester stores Claims, Evidence, credentials or policies associated with Attestation where no other Computing Context is expected to provide its Attestation Evidence.
 - \$ 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]

3.8. Interpretations of RFC4949 Terminology for Attestation

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".

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.

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.

3.9. 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.

4. IANA considerations

This document will include requests to IANA:

- o first item
- o second item

5. Security Considerations

There are always some.

6. Acknowledgements

Maybe.

7. Change Log

No changes yet.

8. References

8.1. 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>>.

8.2. 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-15 (work in progress), June 2018.

[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>>.

8.3. 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

