Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 3, 2019

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

July 02, 2018

Reference Terminology for Remote Attestation Procedures
draft-birkholz-attestation-terminology-02

Abstract

This document is intended 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 attestation procedures in the IETF and beyond.

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 January 3, 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 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 2
1.1. Requirements notation 4
2. Basic Roles of RATS 4
3. Computing Context 4
3.1. Formal Semantic Relationships 5
3.2. Characteristics of a Computing Context 6
4. Computing Context Identity 7
5. Attestation Workflow 7
6. Reference Use Cases 8
6.1. The Lying Endpoint Problem 10
6.2. Who am I a talking to? 11
7. Trustworthiness 11
8. Remote Attestation 12
8.1. Building Block Terms 12
9. IANA considerations 13
10. Security Considerations 13
11. Acknowledgements 13
12. Change Log 13
13. References 14
13.1. Normative References 14
13.2. Informative References 14
Authors' Addresses 14

1. Introduction

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 intend 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.

The primary application of RATS is to increase the trust and confidence in the integrity of the object characteristics and properties of a system entity that is intended to interact and

exchange data with other system entities remotely. How an objects's characteristics are attested remotely and which characteristics are actually chosen to be attested varies with the requirements of the use cases, or -- in essence -- depends on the risk that is intended to be mitigated via RATS. Effectively, RATS are a vital tool to be used to increase the confidence in the level of trust of a system that is supposed to be a trusted system.

In the remainder of this document a system that is capable to provide an appropriate amount of information about its integrity is considered to be a trustworthy system - or simply trustworthy.

The primary characteristics of a trustworthy system are commonly based on information about the integrity of its intended composition, its enrolled and subsequently installed software components, and the scope of known valid states that a trustworthy system is supposed to operate in.

It is important to note that the activity of attestation itself in principle only provides the evidence that proves the integrity of a (subset) of a system's object characteristics. The provided evidence is used as a basis for further activities. Specific RATS define the higher semantic context about how the evidence is utilized and what RATS actually can accomplish; and what they cannot accomplish, correspondingly. Hence, this document is also intended to provide a map of terms, concepts and applications that illustrate the ecosystem of current applications of RATS.

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

Please note that this document is still missing multiple reference 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.

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. Basic Roles of RATS

The use of the term Remote Attestation Procedures always implies the involvement of at least two parties that each take on a specific role in corresponding RATS - the Attestor role and the Verifier role. Depending on the object characteristics attested and the nature of the parties, information is exchanged via specific types of Interconnects between them. The type of interconnect ranges from GIO pins, to a bus component, to the Internet, or from a direct physical connection, to a wireless association, to a world wide mesh of peers. In other words, virtually every kind communication path (Interconnect) can be used by system entities that take on the role of Attestor and Verifier (in fact, a single party can take on both roles at the same time, but there is only a limited use to this architecture).

Attestor: The role that designates the subject of the remote attestation. A system entity that is the provider of evidence takes on the role of an Attestor.

Verifier: The role that designates the system entity that is the appraiser of the evidence provided by the Attestor. A system entity that is the consumer of evidence takes on the role of a Verifier.

Interconnect: A channel of communication between Attestor and Verifier that enables the appraisal of evidence created by the Attestor by a remote Verifier.

3. Computing Context

This section introduces the term Computing Context in order to simplify the definition of RATS terminology.

The number of approaches and solutions to create things that provide the same capabilities as a "simple physical device" continuously increases. 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.

System entities are composed of system entities. In essence, every physical or logical device is a composite of system entities. In consequence, a composite device also constitutes a system entity. Every component in that composite is a potential Computing Context capable of taking on the roles of Attestor or Verifier. The scope and application of these roles can range from:

- o continuous mutual attestation procedures of every system entity 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 system entities that can take on the role of an Attestor 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. Formal Semantic Relationships

The formal semantic relationship of a Computing Context and the definitions provided by RFC 4949 is a as follows.

The scope of the term computing context encompasses

- o an information system,

- o an object and in consequence a system component or a composite of system sub-components, and

- o a system entity or a composite of system entities.

Analogously, a sub-context is a subsystem and as with system components, computing contexts can be nested and therefore be

physical system components or logical ("virtual") system (sub-)components.

The formal semantic relationship is based on the following definitions from RFC 4949.

(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.

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

System Component: A collection of system resources that (a) forms a physical or logical part of the system, (b) has specified functions and interfaces, and (c) is treated (e.g., by policies or specifications) as existing independently of other parts of the system. (See: subsystem.)

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

System Entity: An active part of a system - [...] (see: subsystem) - that has a specific set of capabilities.

3.2. 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:

A computing context:

- o provides its own independent environment in regard to executing and running software,

- o provides its own isolated control plane state (by potentially interacting with other Computing

- o Contexts) and may provide a dedicated management interface by which control plane behavior can be effected,

- o can be identified uniquely and therefore reliably differentiated in a given scope, and

- o does not necessarily has to include a network interface with associated network addresses (as required, e.g. by the definition of an endpoint) - although it is very likely to have (access to) one.

In contrast, a docker [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.

4. Computing Context Identity

The identity of a Computing Context provides the basis for creating evidence about data origin authenticity. Confidence in the identity assurance level [NIST SP-800-63-3] or the assurance levels for identity authentication [RFC4949] impacts the confidence in the evidence an Attestor provides.

5. Attestation Workflow

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 on 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 piece 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.

6. Reference Use Cases

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.

These use case summary highlighted above is based in the following terms defined in RFC4949 and complementary sources of terminology:

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 (Attestor), of an conveyance activity that transfers evidence (interconnect), and of a verification activity that appraises evidence (Verifier), in respect to correctness integrity.

Identity: Defined by [RFC4949] as the collective aspect of a set of attribute values (i.e., a set of characteristics) by which a system user or other system entity is recognizable or known. (See: authenticate, registration. Compare: identifier.)

There are different scopes an identity can apply to:

Singular identity: An identity that is registered for an entity that is one person or one process.

Shared identity: An identity that is registered for an entity that is a set of singular entities in which each member is authorized to assume the identity individually, and for which the registering system maintains a record of the singular entities that comprise the set. In this case, we would expect each member entity to be registered with a singular identity before becoming associated with the shared identity.

Group identity: An identity that is registered for an entity that is a set of entities (2) for which the registering system does not maintain a record of singular entities that comprise the set.

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

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.

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.

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.

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.

Forward Authenticity (FA): A property of secure communication protocols, in which later compromise of the long-term keys of a data origin does not compromise past authentication of data from that origin. FA is achieved by timely recording of assessments of the authenticity from entities (via "audit logs" during "audit sessions") that are authorized for this purpose, in a time frame much shorter than that expected for the compromise of the long-term keys.

6.1. The Lying Endpoint Problem

A very prominent goal of attestation procedures - and therefore a suitable example used as reference in this document - is to address the "lying endpoint problem".

Information created, relayed, or, in essence, emitted by a computing context does not have to be correct. There can be multiple reasons why that is the case and the "lying endpoint problem" represents a scenario, in which the reason is the compromization of computing contexts with malicious intend. A compromised computing context could try to "pretend" to be integer, while actually feeding manipulated information into a security domain, therefore compromising the effectiveness of automated security functions. Attestation - and remote attestation procedures specifically - is an approach intended to identify compromised software instances in computing contexts.

Per definition, a "lying endpoint" cannot be "trusted system".

Trusted 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.

Remote attestation procedures are intended to enable the consumer of information emitted by an computing context to assess the validity and integrity of the information transferred. The approach is based on the assumption that if 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 to also provide guarantees about the validity of the evidence they can create.

6.2. Who am I a talking to?

[working title, write up use case here, ref teep requirements]

7. Trustworthiness

A "lying endpoint" is not trustworthy.

Trusted 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.

Trustworthy: pull in text here

8. Remote Attestation

Attestation: An object integrity authentication facilitated via the creation of a claim about the properties of an Attestor, such that the claim can be used as evidence.

Conveyance: The transfer of evidence from the Attestor to the Verifier.

Verification: The appraisal of evidence by evaluating it against declarative guidance.

Remote Attestation: A procedure composed of the activities attestation, conveyance and verification.

8.1. Building Block Terms

[working title, pulled from various sources, vital]

Attestation Identity Key (AIK): A special purpose signature (therefore asymmetric) key that supports identity related operations. The private portion of the key pair is maintained confidential to the computing context via appropriate measures (that have a direct impact on the level of confidence). The public portion of the key pair may be included in AIK credentials that provide a claim about the computing context.

Claim: A piece of information asserted about a subject. 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].

Computing Context Characteristics: The composition, configuration and state of a computing context.

Evidence: A trustworthy set of claims about an computing context's characteristics.

Identity: A set of claims that is intended to be related to an entity. [merge with RFC4949 defintion above]

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).

Trustworthiness: The qualities of computing context characteristics that guarantee a specific behavior specified by declarative guidance. Trustworthiness is not an absolute property but defined with respect to a computing context, corresponding declarative guidance, and has a scope of confidence. A trusted system is trustworthy. [refactor defintion with RFC4949 terms]

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

Trustworthy Statement: evidence that trustworthy conveyed by a computing context that is not necessarily trustworthy. [update with tamper related terms]

9. IANA considerations

This document will include requests to IANA:

- first item
- 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>.

[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>.

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-14 (work in progress), December 2017.

[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>.

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