

anima Working Group
Internet-Draft
Intended status: Standards Track
Expires: 19 April 2026

M. Richardson
Sandelman Software Works
H. Birkholz
Fraunhofer SIT
Y. Deshpande
Arm
16 October 2025

Taxonomy of Composite Attesters
draft-richardson-rats-composite-attesters-02

Abstract

This document further refines different kinds of RFC 9334 Composite Attesters.

About This Document

This note is to be removed before publishing as an RFC.

Status information for this document may be found at
<https://datatracker.ietf.org/doc/draft-richardson-rats-composite-attesters/>.

Discussion of this document takes place on the rats Working Group mailing list (<mailto:rats@ietf.org>), which is archived at <https://mailarchive.ietf.org/arch/browse/rats/>. Subscribe at <https://www.ietf.org/mailman/listinfo/rats/>.

Source for this draft and an issue tracker can be found at <https://github.com/richardson/rats-composite-attesters>.

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 19 April 2026.

Copyright Notice

Copyright (c) 2025 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	2
1.1. Caveats of Current Definition	3
1.2. Terminology	3
1.3. Class 0 Composite Attester	4
1.4. Class 1 Composite Attester	5
1.5. Class 2 Composite/Hybrid Attester	7
1.6. Class 3B Composite Background-Check Attester	8
1.7. Class 3P Composite Passport-Model Attester	9
1.8. Class 4 Composite Attester	11
1.9. Class 5 Mixed Composite Attester	12
2. Attestation Results as Evidence	12
3. Privacy Considerations	12
4. Security Considerations	12
5. Nonce Architecture	13
6. IANA Considerations	13
7. Acknowledgements	13
8. Changelog	13
9. References	13
9.1. Normative References	13
9.2. Informative References	13
Authors' Addresses	14

1. Introduction

This document clarifies and extends the meaning of Composite Attester from [RFC9334], Section 3.3.

1.1. Caveats of Current Definition

[RFC9334], Section 3.3 says:

``` A composite device is an entity composed of multiple sub-entities such that its trustworthiness has to be determined by the appraisal of all these sub-entities.

Each sub-entity has at least one Attesting Environment collecting the Claims from at least one Target Environment. Then, this sub-entity generates Evidence about its trustworthiness; therefore, each sub-entity can be called an "Attester". Among all the Attesters, there may be only some that have the ability to communicate with the Verifier while others do not. ```

In this description, it was left vague as to whether or not each Attesting Environment signs the Evidence that it generates, and whether or not the Evidence is evaluated by a Verifier operated by the Lead Attester, or if it's passed by the Lead Attester along with the Evidence from the Lead Target Environment.

### 1.2. Terminology

**Lead Attester:** This term is from RFC9334, and includes the (Lead) Attesting Environment, and the (Lead) Target Environment.

**Target Environment:** This term is from RFC9334, this refers to the environment for which Evidence is gathered.

**Attesting Environment:** This term is from RFC9334, this refers to the thing which gathers the Evidence.

**Component:** This is the pieces which are attached to the Lead Attester. There are one to many of these, typically each with their own application specific processor.

**Component Evidence:** This is the Evidence that is collected by the Component Attesting Environment about the Component Target Environment.

**Component Attesting Environment:** This term is new, and refers to an Attesting Environment residing inside a component of the whole.

**Component Target Environment:** This term is new, and refers to an environment for which Evidence is collected.

**Verifier le petit:** (Or, Le Petit Verificateur). This is the

Verifier that examines the Component Evidence. This may treat the Lead Attester as a component.

Verifier la grand: (Or, La Grande Verificateur). This is the Verifier that examines the arrangement and relationships between Components.

### 1.3. Class 0 Composite Attester

In this first, somewhat degenerate scenario, the Lead Attester has access to the entire memory/environment of all of the components. Examples of situations like this include classic PCI-buses, ISA-buses, VME, S100/IEEE 696-1983. In these situations, secondary components might not boot on their own. (It might even be that the lead environment (the chassis) will place code into RAM for these systems, with no ROM at all)

In this case, it is possible for the Lead Attesting Environment to collect Claims about each of the components without the components having to have their own Attesting Environment.

There is no Verifier le petit, since there are no components that can create Evidence other than the Lead Attester.

At this Class, all of these components can be considered part of the same system. In the classic PCI or ISA environment, the components are hard drive interfaces, video interfaces, and network interfaces. For many such systems considering the system to be a composite is unnecessary additional complexity.

The benefit of applying the composite mechanism in this case is that it is no longer necessary to consider the exhaustive combinatorics of all possible components being attached to the lead attester. It is, for instance, already the case the reference values for a target environment may change depending upon how much memory is installed in the target environment.

In this degenerate, or Class \_0\_ Composite Attester, the Claims gathered about the components would be included in the Lead Attester's signed Evidence (such as an EAT), as sub-components in UCCS form [RFC9781]. The signature from the Lead Attester applies to all the Claims, but the Verifier can evaluate each component separately.

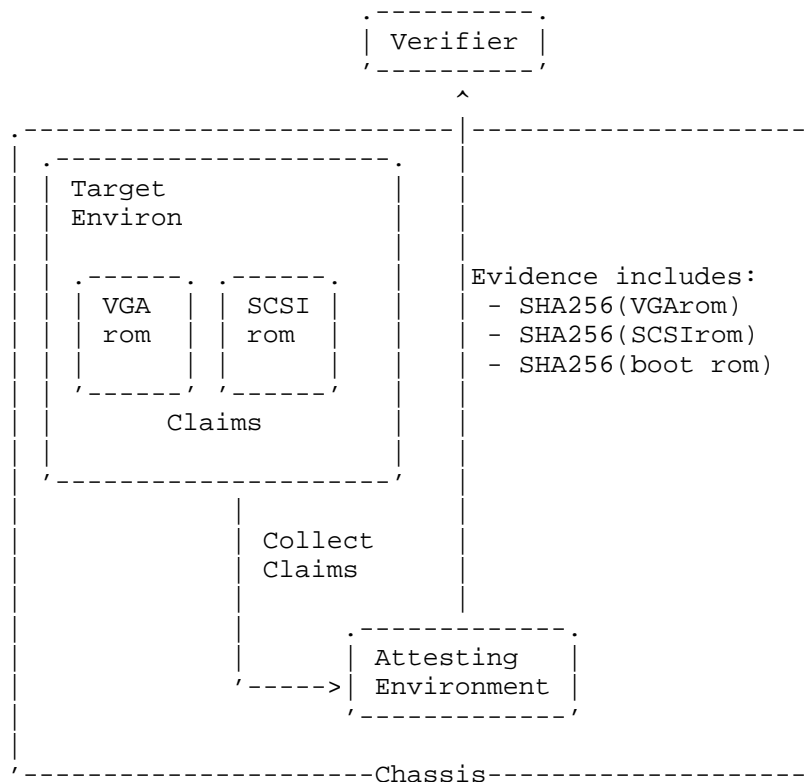


Figure 1: Class 0 Composite Attester

However, more modern buses like PCIe, InfiniBand, Thunderbolt, DisplayPort, USB, Firewire and others do not provided direct electrical access to target component system memory. While some seem to be very high speed serialized versions of the old I/O buses, there is a network-like protocol, and non-trivial deserialization occurs at each end. That implies that there can be mutable firmware in each component which mitigates access. That firmware itself might not be trustworthy. If it can even be seen by the Lead Attester, the mitigation mechanism can present whatever view the Lead Attester expects to see. So, a system with such interfaces would be a Class 1.

#### 1.4. Class 1 Composite Attester

In this Class, each component or slot has its own Attesting Environment and hence produces its own signed Evidence.

RFC 9334 gives the following example:

For example, a carrier-grade router consists of a chassis and multiple slots. The trustworthiness of the router depends on all its slots' trustworthiness. Each slot has an Attesting Environment, such as a TEE, collecting the Claims of its boot process, after which it generates Evidence from the Claims.

The Lead Attester simply relays the Evidence along with its own:

Among these slots, only a "main" slot can communicate with the Verifier while other slots cannot. However, other slots can communicate with the main slot by the links between them inside the router. The main slot collects the Evidence of other slots, produces the final Evidence of the whole router, and conveys the final Evidence to the Verifier. Therefore, the router is a composite device, each slot is an Attester, and the main slot is the lead Attester.

Note that the Lead Attester does not evaluate the Evidence, and does not run its own Verifier.

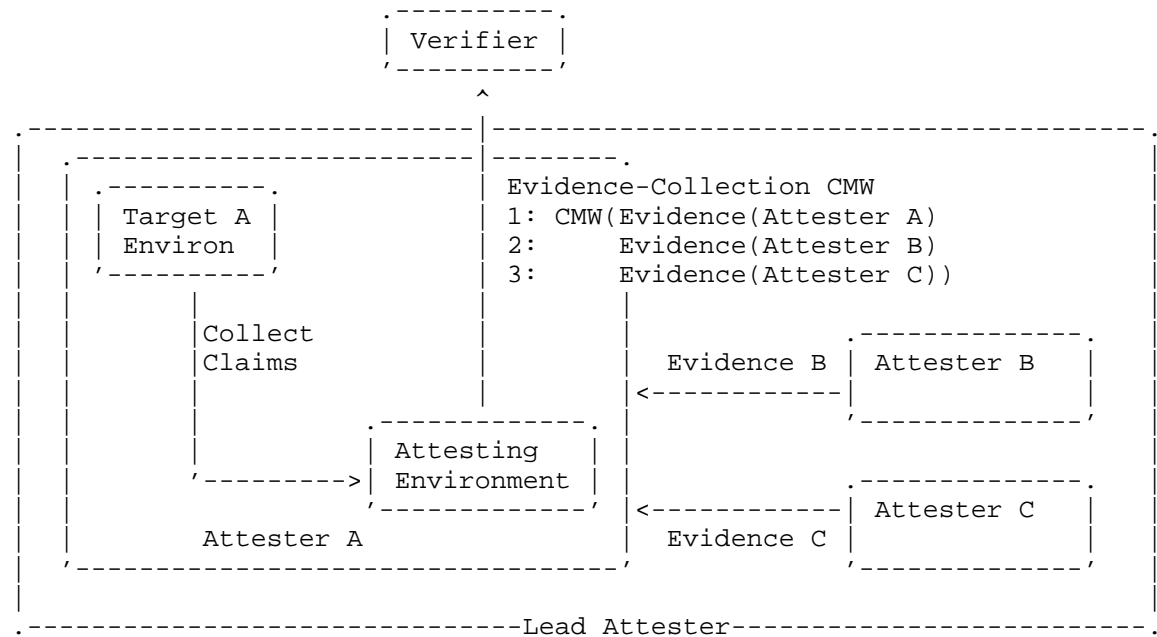


Figure 2: Class 1 Composite Attester

This diagram is intended to be identical to Figure 4 of [RFC9334], but has been stretched out to allow the relationship to other classes to be clearer.

### 1.5. Class 2 Composite/Hybrid Attester

In this scenario, the Components relay their Evidence to the Lead Attester. The Lead Attester operates a Verifier itself. It evaluates the Components' Evidence against Reference Values, Endorsements, etc. producing `_Attestation Results_`. These Attestation Results (or their selectively disclosed version: SD-CWT/SD-JWT) are then included as part of the Lead Attester's Evidence to it's Verifier.

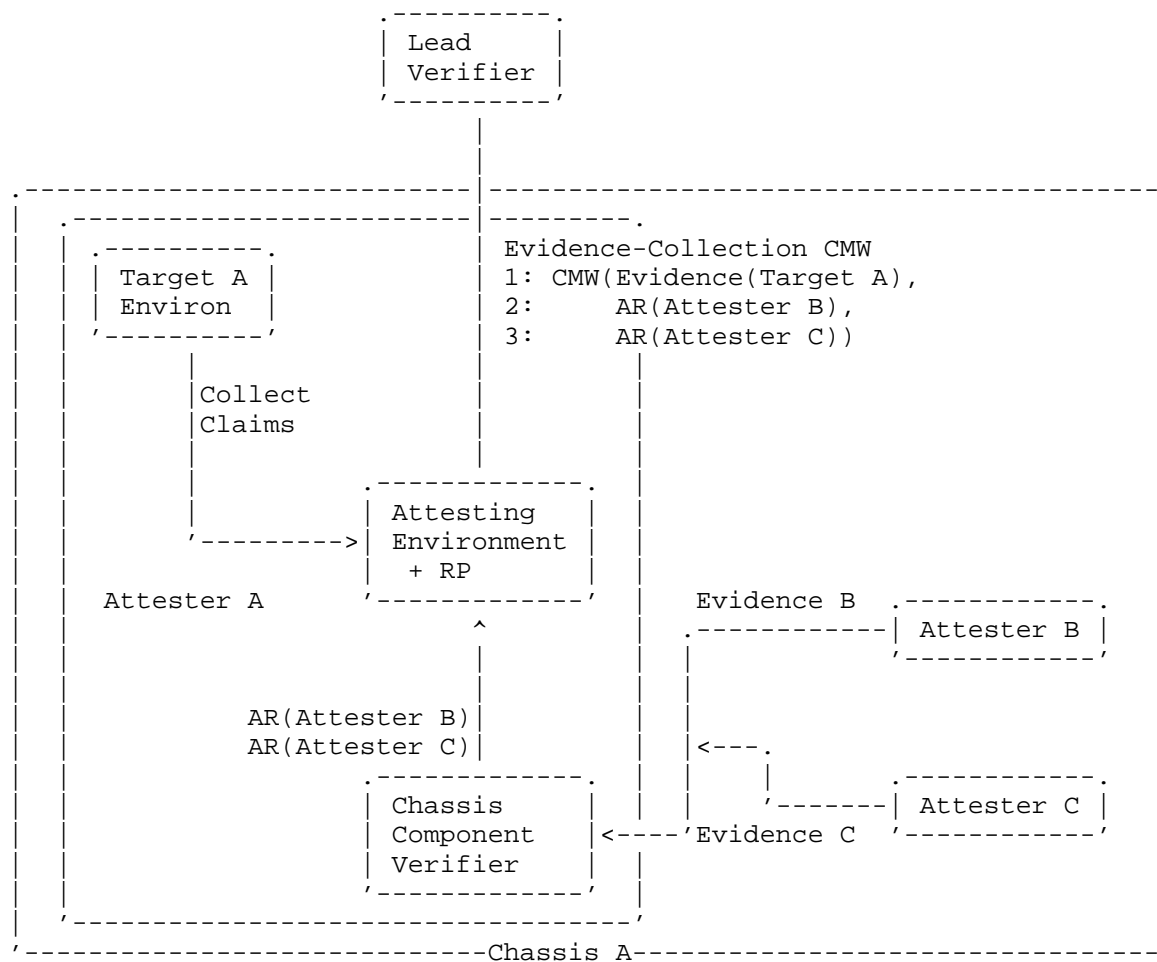


Figure 3: Class 2 Composite Attester

The Verifier’s signing credentials may be part of the same Attesting Environment as the Evidence signing credential used by the Lead Attesting environment. Or they could be in a different environment, such as in a different TEE.

1.6. Class 3B Composite Background-Check Attester

In this scenario, the Components relay their Evidence to the Lead Attester. The Lead Attester does not operates a Verifier itself.

Instead, the Lead Attester, conveys the Evidence to the Lead Verifier along with it’s own Evidence. The Component Evidence is not placed within the Lead Attester’s Evidence (DEBATE). The Lead Attester needs to communicate how each component is attached, and that would be within its Evidence.

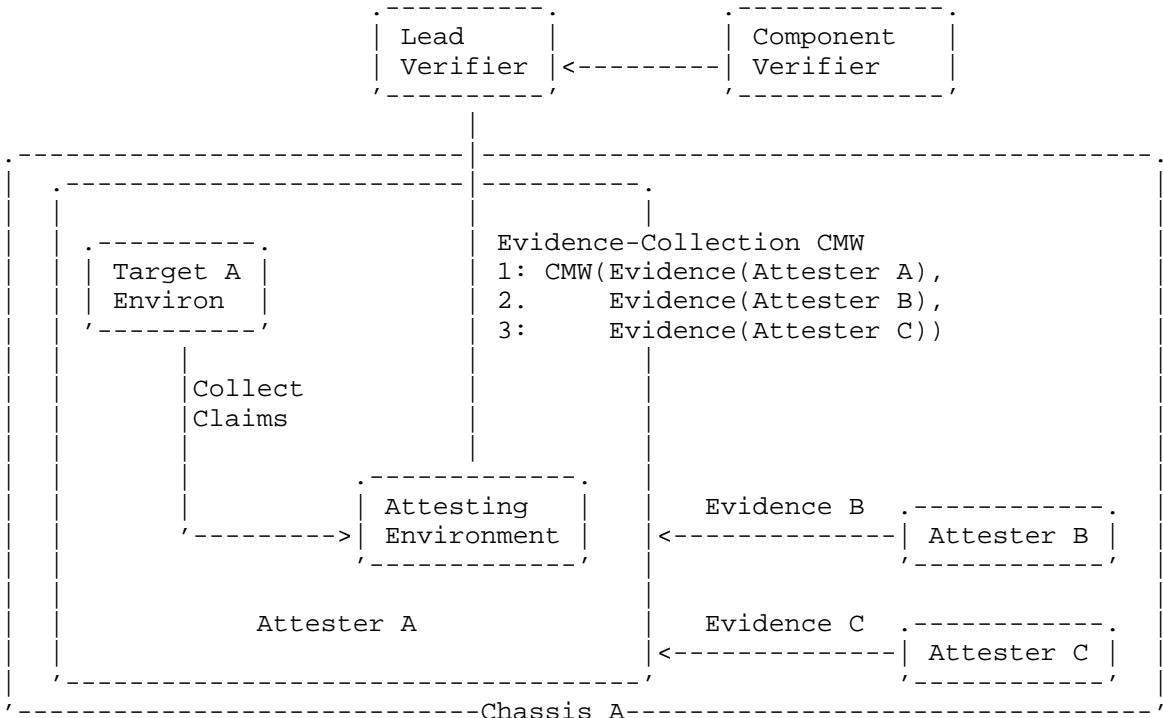


Figure 4: Class 3B Composite Background-check Attester

The Lead Verifier, acting a Relying Party, connects to Component Verifiers capable of evaluating the Component Evidence, retrieving Attestation Results from those Verifiers as part of evaluating the Lead Attester.



This case is similar to Class 1, however the integration of the component attestation results in Class 1 is not included in the Evidence, while in this case, it is.

#### 1.7. Class 3P Composite Passport-Model Attester

In this scenario, the Components relay their Evidence to the Lead Attester. The Lead Attester does not operate a Verifier itself. Instead, the Lead Attester, acting as a Presenter (term To-Be-Defined), connects to an appropriate Verifier, in passport mode. It retrieves an Attestation Result from the Verifier, which it then includes within the Evidence that the Lead Attester produces.

The Lead Attester's Verifier considers the Components during its assessment. It needs to consider if the component has been assessed by a Verifier it trusts, if the component is appropriately connected to the Lead Attester, and if there are an appropriate number of such components.

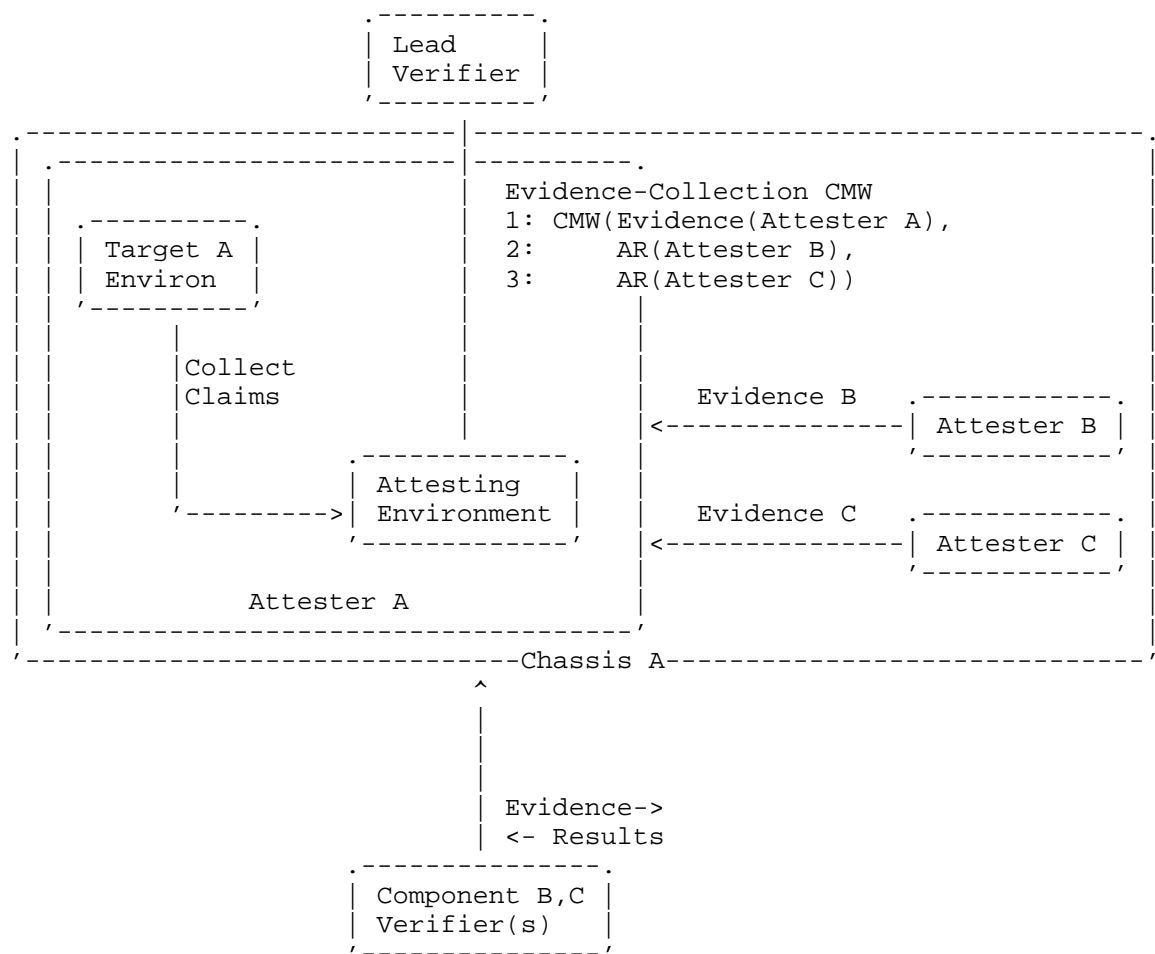


Figure 5: Class 3P Composite Password Attester

For instance, when accessing a vehicle such as a car, where each tire is it's own component, then a car with three wheels is not trustworthy. Most cars should have four wheels. A car with five wheels might be acceptable, if at least one wheel is installed into the "spare" holder. (And, it may be of concern if the spare is flat, but the car can still be operated)

A more typical digital use case would involve a main CPU with a number of attached specialized intelligent components that contain their own firmware, such as Graphical Processors (GPU), Network Processors (NPU).

### 1.8. Class 4 Composite Attester

In certain systems, it is possible to have two independent Attesting Environments in an Attester to collect claims about a single Target Environment. In such cases, one of the Attesting Environment, acts as a Primary, while the other acts as a Secondary Attesting Environment.

The two Attesting Environments will have a fixed and collaborative structure where each can be responsible for a subset of Evidence. Because of the collaborative structure it may be arranged that either of the Attesting Environment can present Evidence collected by the other (but this is deployment specific).

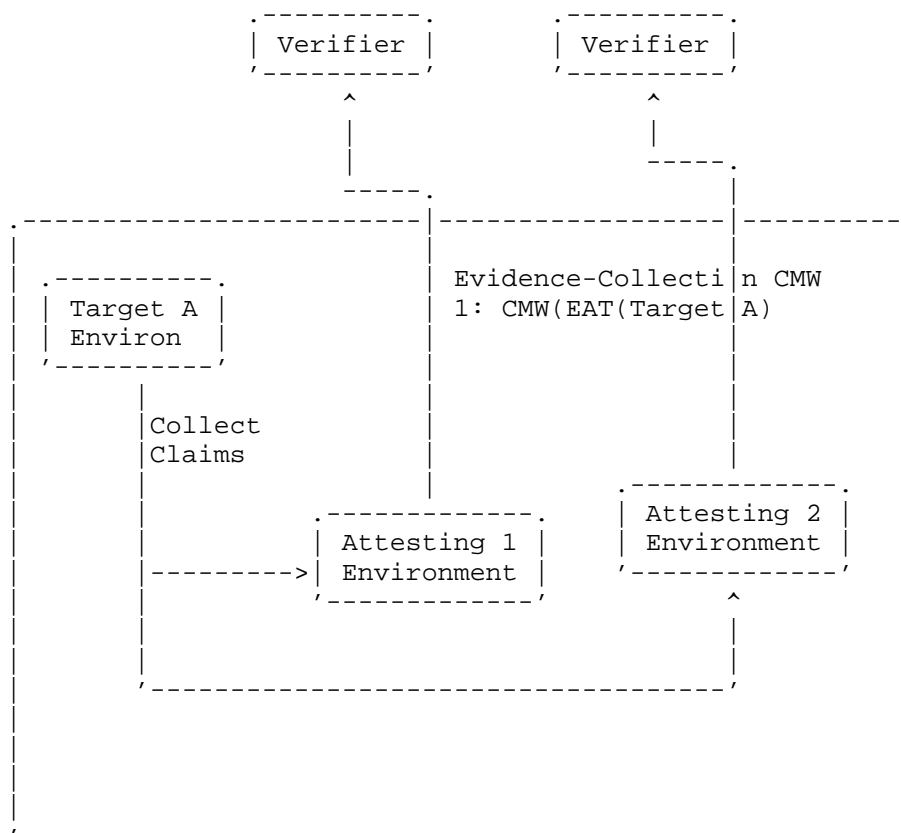


Figure 6: Class 4 Composite (Dual) Attester

Example of one such system is a CPU system of a desktop from a Vendor X, which has its built in Attesting Environment, integrated into a product Y which requires a mandatory TPM support. In such situations one can anchor the Roots of Trust of Vendor X's CPU Attestation using a secondary Attesting Environment with the TPM Attestation. Alternatively, generate a TPM Quote and anchor it to Root of Trust of CPU Attestation based of Vendor X's Attesting Environment.

A Verifier/RP may decide to direct the Attestation Request to an AE of choice to reflect the relevant subset of Evidence required for trust assessment.

#### 1.9. Class 5 Mixed Composite Attester

As soon as there is more than one Component, it is reasonable that the different Components interact with the Lead Attester in different ways. A Mixed Composite Attester would have a components that come from different classes. This is not a class itself, but a class of classes.

Degenerately, all previous classes can be considered mixes of one, but such a trivial category does not help discussionn. Except that adding/moving/replacing Components in the field can change things, so some system architectures will need to always consider themselves to be Mixed Composite Attesters, even if when shipped, they might be degenerate instances.

#### 2. Attestation Results as Evidence

In cases 2, 3B and 3P Attestation Results are included as Evidence. This results in a Verifier that must evaluate these results. It must be able to validate the signatures on the Evidence.

This creates \_stacked\_ Remote Attestation. This is very much different and \_distinct\_ from [RFC9334], Section 3.2 Layered Attestation.

Layered Attestation produces a \_single\_ set of Evidence, with claims about different layers.

#### 3. Privacy Considerations

YYY

#### 4. Security Considerations

ZZZ

## 5. Nonce Architecture

In all classes other than the class 0 and class 1, there are cases that multiple (local or external) Verifiers exist in the system. To address the conflict between different nonces generated by different Verifiers, there are possible candidate solutions as follows

-- Using one unique nonce from one external Verifier: This Verifier initiates the attestation progress and other Verifiers use the same nonce to challenge their corresponding Attesters. To ensure the integrity of the nonce, this nonce SHOULD be signed by this initial Verifier.

-- Each Verifier uses their own nonce: The Evidence in such a case is the mixing of certain Evidences and Attestation Result-as-Evidences. The receiver of the Attestation Results (the Attester) can apply the technique in [RFC9334], Appendix A.2 to ensure the freshness of the Attestation Result-as-Evidences.

## 6. IANA Considerations

## 7. Acknowledgements

Jun Zhang contributed the terms "Le Petit" and "La Grand" to qualify Verifier.

## 8. Changelog

## 9. References

### 9.1. Normative References

- [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>>.
- [RFC9334] Birkholz, H., Thaler, D., Richardson, M., Smith, N., and W. Pan, "Remote ATtestation procedureS (RATS) Architecture", RFC 9334, DOI 10.17487/RFC9334, January 2023, <<https://www.rfc-editor.org/info/rfc9334>>.

### 9.2. Informative References

- [RFC9781] Birkholz, H., O'Donoghue, J., Cam-Winget, N., and C. Bormann, "A Concise Binary Object Representation (CBOR) Tag for Unprotected CBOR Web Token Claims Sets (UCCS)", RFC 9781, DOI 10.17487/RFC9781, May 2025, <<https://www.rfc-editor.org/info/rfc9781>>.

Authors' Addresses

Michael Richardson  
Sandelman Software Works  
Email: mcr+ietf@sandelman.ca

Henk Birkholz  
Fraunhofer SIT  
Email: henk.birkholz@ietf.contact

Yogesh Deshpande  
Arm  
Email: yogesh.deshpande@arm.com