RATS Working Group
Internet-Draft
Intended status: Informational
Expires: 23 April 2026

M. Novak
J.P. Morgan Chase
Y. Deshpande
Arm
H. Birkholz
Franhaufer Inst.
20 October 2025

Remote Attestation for Trustworthy Workload Identity
draft-novak-twi-attestation-00

Abstract

Trustworthy Workloads are workloads that operate in environments that provide isolation of data in use. This document describes how Trustworthy workloads can acquire credentials containing stable identifiers, upon proving the trust in the environments in which they operate via Remote Attestation.

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-novak-twi-attestation/.

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/.

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 23 April 2026.

Table of Contents

1. Introduction

As organisations move more workloads into untrusted or shared environments, protection of data in use becomes increasingly important. One way of isolating data in use is Confidential Computing: executing a workload (for example an AI model, database process or financial service) inside a hardware-based, remotely attested Trusted Execution Environment (TEE). Workloads operating in such environments need stable and trustworthy identifiers to communicate over the network to the external world. Often such identifiers are provided to them via Credential Authorities upon ascertaining trust in the environments in which these workloads operate. The standard practice to establish trust in the operating environment is through Remote Attestation.

This draft specifies how a Workload operating in Confidential Computing Environment can obtain trustworthy, stable, and workload-bound credentials using Remote Attestation.

2. Conventions and Definitions

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.

This document uses terms and concepts defined by the WIMSE and RATS architectures, as well as the terms defined by the Trustworthy Workload Identity Special Interest Group at the Confidential Computing Consortium. For a complete glossary, see Section 4 of [RFC9334] , [I-D.draft-ietf-wimse-arch] & [TWISIGDef].

The definitions of terms like Trustworthy Workload Identity and Workload Credential match those specified by the TWI SIG Definitions [TWISIGDef].

Workload: [I-D.draft-ietf-wimse-arch] defines ’Workload’ as "an instance of software executing for a specific purpose". Here we restrict that definition to the portions of the deployed software and its configuration that are subject to Remote Attestation.

Workload Identifier: a stable construct around which Relying Parties can form long-lived Workload authorization policies.

Workload Identity: the definition of Workload Identity is identical to the definition of the same term by [I-D.draft-ietf-wimse-arch]: "a combination of three basic building blocks: trust domain, Workload Identifier and identity credentials.

Workload Credential: an ephemeral identity document containing the Workload Identifier and a number of additional claims, that can be short-lived or long-lived, and that is used to represent and prove Workload Identity to a Relying Party.

Stable Workload Identity, Stable Authorization Policy: a Workload Identity or Authorization Policy is considered Stable if it remains constant in the face of software and hardware changes (updates and rollbacks), so long as those updates and rollbacks are authorized, i.e., comply with the policy of what consitutes the allowed version(s) of the software and hardware in question.

Credential Authority: an entity trusted to issue Workload

Credentials

Bound Workload Credential: a Workload Credential is considered Bound if it can only be used in conjunction with a secret Credential Key that only a Workload authorized for the use of that Key can obtain, either by generating and certifying it, or by retrieving it from a secure Key Store.

Workload Owner: an entity tasked with specifying policies concerning what Workload composition is considered valid for the purposes of issuing Workload Credentials

Verifier: an entity performing the role of Attestation Verification, as documented in Section 4 of [RFC9334]

3. Available Options

When dealing with a client Workload that is running inside a remotely attested Trusted Execution Environment, the goal of having a Relying Party having a stable authorization policy and utilizing industry-standard mechanisms for authorization can be achieved by issuing Credentials in a relying party-friendly format, such as those specified by [I-D.draft-ietf-wimse-arch]. Such Credentials may take the form of x.509 certificates or Workload Identity Tokens (WITs) defined in Section 3.1 of [WIMSES2S]. A Workload can start using the Credential for authentication and authorization once it has two items in its possession: the public portion the Workload Credential itself, and the secret Credential Key necessary to utilize this Credential.

A Stable authorization policy can only be achieved if Workloads can have Stable identities. The decision about what constitutes a trustworthy Workload and a trustworthy configuration is a composition verification, with multiple entities providing Reference Values for the components they vouch for. For the issued Workload Identity to be Stable in addition to Trustworthy, a mapping must be performed between these Reference Values and the issued Identities. In a typical enterprise, Stable authorization policies are expressed in terms of business- rather than technology-oriented concepts, e.g., "Payroll Application", "Located in Germany", "Cleared for handling Personally Identifiable Information", etc. This contrasts with what RATS has historically thought of as Attestation Results, which may relate to the hardware manufacturer, firmware and software versions, etc.

In some implementations, a Credential is precomputed, and the Credential Key is obtained from a Key Store following successful Remote Attestation. In other implementations, the Workload generates its own Credential Key and uses Remote Attestation to certify it.

Within the RATS Architecture, either of these options can be accomplished in one of two ways:

1. The Attestation Results convey to the attesting Workload both the public Credential and the secret Credential Key.

2. The Attestation Results are encoded in an Entity Attestation Token or EAT [RFC9711], or a bespoke Verifier-specific format, and can be used by the attesting Workload to obtain a Bound Credential and an associated Credential Key, e.g., by contacting a Credential Authority and/or a Key Store, but without further involving the Verifier.

In either case, the detailed information about the Workload's composition conveyed to the Verifier using RATS "Evidence" is mapped to Stable, technology-agnostic, business-oriented claims about the Workload.

These two options can be visualised at a high level as:
// Tracked at: https://github.com/confidential-computing/twi-rats/issues/5

From the Workload's perspective, Variant 2 carries with it an extra network roundtrip (the first roundtrip being the workload exchanging "Evidence" for "Attestation Results"). It is the only option available to the Workload for using existing Verifier implementations that make no changes associated with this proposal. This option does however introduce additional latency and reliability costs inherent in an extra roundtrip.

Variant 1 does not carry with it the extra roundtrip, and thus does not carry the additional performance costs or reliability risks.

Several distinct options are possible. In all cases, the Credential is generated and signed by a Credential Authority. The difference is in how the Workload obtains these Credentials. The main pivots are:

1. Where the Credential Key is generated (Key Source):

    1. Inside the Workload Instance

    2. Inside a secure Key Store such as a Hardware Security Module (HSM), by the Workload Owner

2. Where the Workload gets its Credential from (Credential Source):

    1. The Verifier

    2. The Credential Authority (e.g., a Certificate Authority, a Security Token Service, or similar)

    3. The Workload Owner (via the Control Plane)

Note that it is safe to receive the Credential from an untrusted source such as the Control Plane, because it is public. The only requirement is that the obtained Credential matches the Credential Key, which MUST always be obtained securely and only by an authorized Workload instance.

Further, under pivot 2.i, the order of interactions involved in Credential generation might differ:

1. A Workload invokes the Verifier which collaborates with the Credential Authority to compute and return Credentials, returning these Credentials inside the Attestation Results, or

2. A Workload invokes the Verifier, obtains from it the Attestation Results, and forwards these Attestation Results to the Credential Authority inside a Credential Request to get the Credential.

This set of variants results in several distinct Credential Acquisition Mechanisms (CAMs), some of which are listed in the table below:

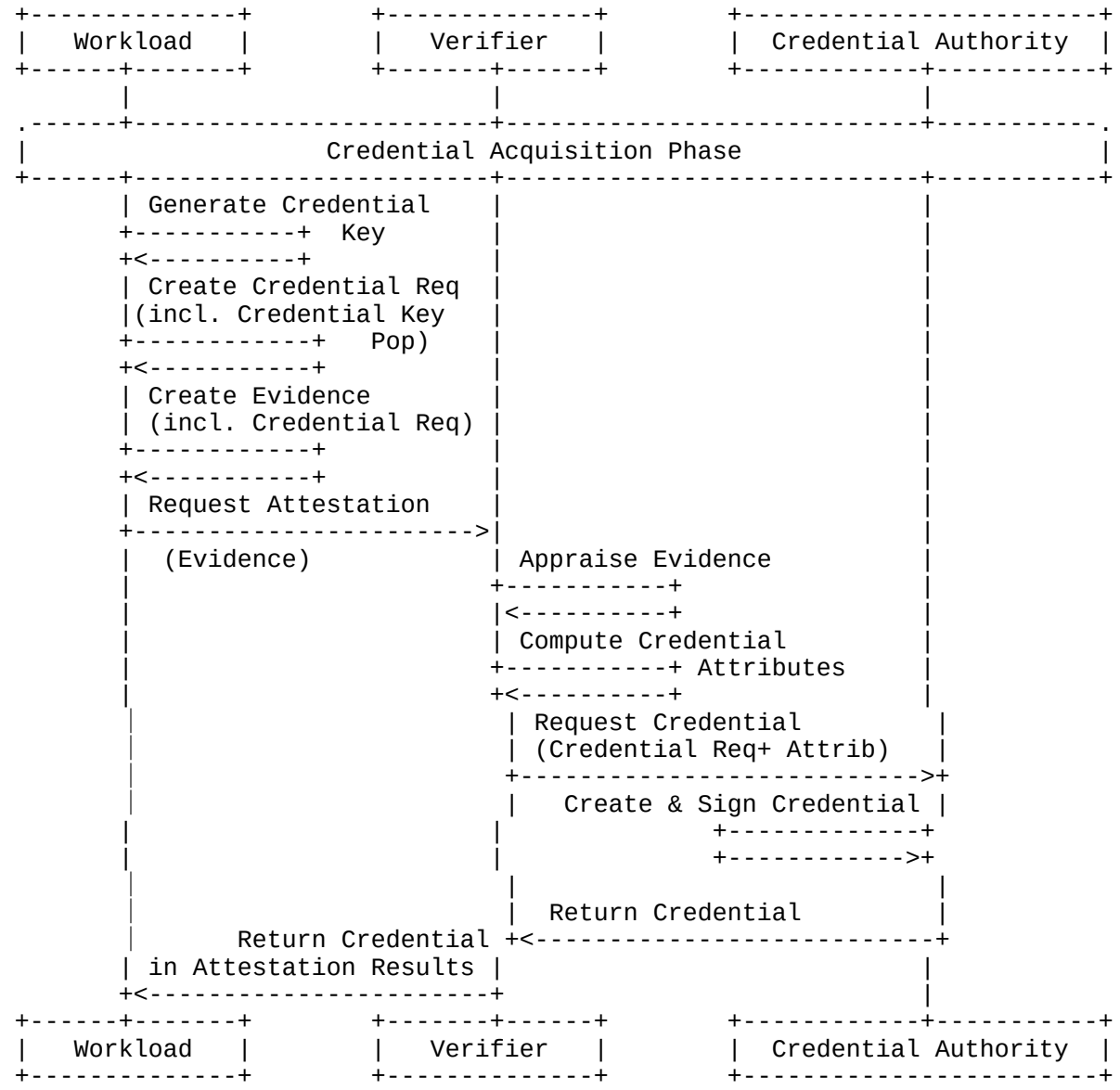
CAM	Key Source	Credential Source	Description
A	Workload	Verifier	A Proof-of-Possession of the Credential Key is included in the Evidence submitted by the Workload Instance to the Verifier. The Verifier checks the Evidence, then contacts the Credential Authority to compute a Credential based on this Credential Key and returns it to the Workload Instance as part of Attestation Results.
B	Workload	Credential Authority	A Proof-of-Possession of the Credential Key is included in

			the Evidence submitted by the Workload Instance to the Verifier, and also in the Attestation Results returned by the Verifier. The Workload Instance sends the Attestation Results obtained from the Verifier to the Credential Authority, which computes and returns to the Workload Instance a Credential based on these Attestation Results.
C	Workload	Credential Authority	A Proof-of-Possession of the Credential Key is included in a Credential Request submitted by the Workload to the Credential Authority alongside Evidence destined for the Verifier. Credential Authority handles the Credential Request by contacting the Verifier on the Workload's behalf, supplying the Evidence from the Credential Request. The Verifier responds with Attestation Results which the Credential Authority uses to compute a Credential, which it then returns to the Workload.
N/A	Workload	Workload Owner	This is not a viable option since a Workload that generates its own Credential Key MUST contact either the Verifier or the Credential Authority to build a Credential for this Key.
D	Key Store	Workload Owner	The Credential is generated and handed to the Workload by the Workload Owner. The Workload Owner stores the Credential Key in the Key Store. The Workload obtains the Credential Key from the Key Store after completing Remote Attestation.

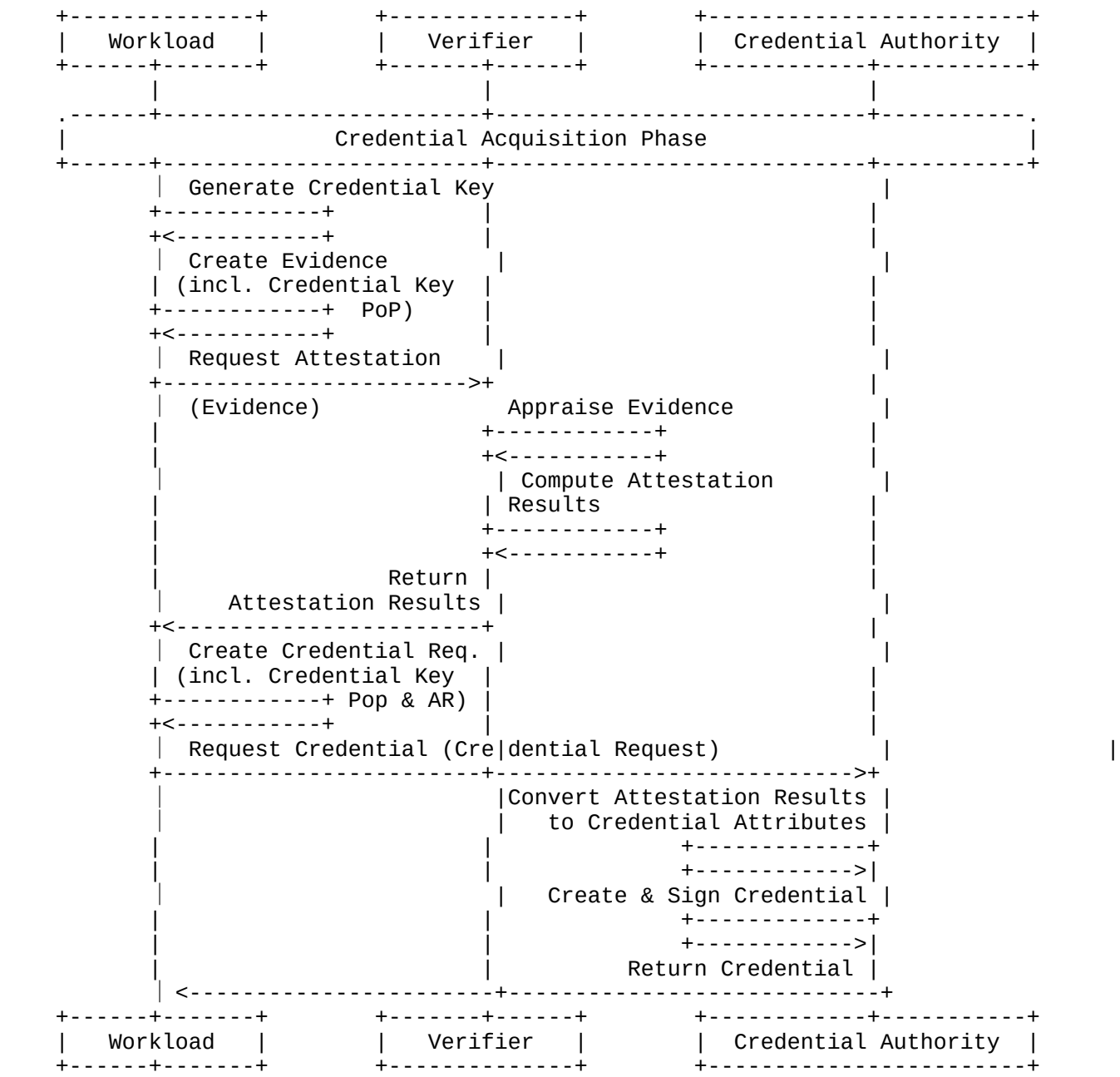
Table 1

These options are illustrated below with sequence diagrams.

3.1. Mechanism A



3.2. Mechanism B



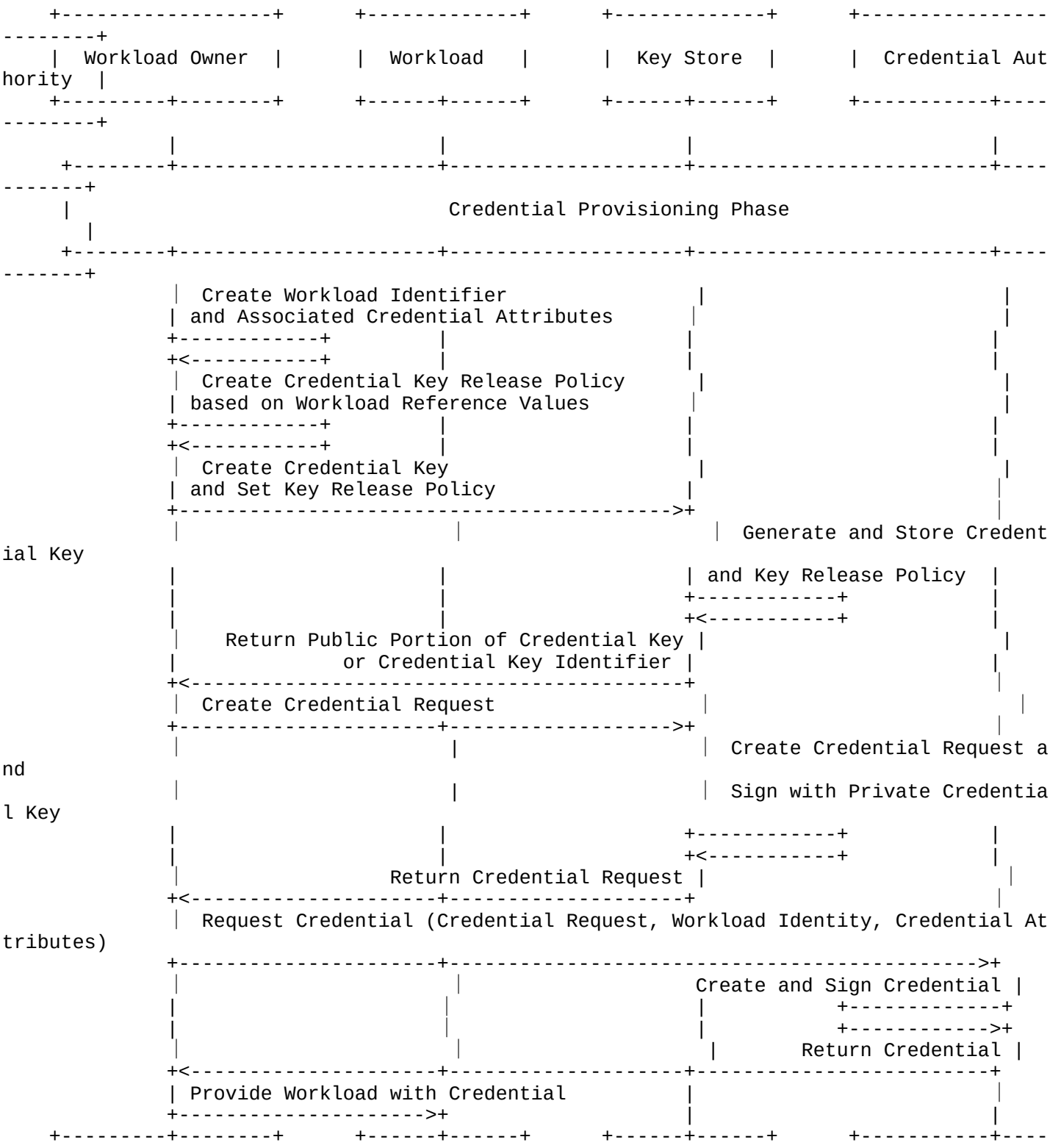
3.3. Mechanism C


Workload
Credential Authority
Verifier
Credential Acquisition Phase
Generate Credential Key
Create Evidence
(incl. Credential Key
PoP)
Create Credential Request
(Evidence,
Credential
Key PoP)
Request Credential
(Credential Request)
Request Attestation
(Evidence from
Credential Request)
Appraise Evidence
Compute Attestation Results
Return Attestation Results
Compute Credential Attrib
(from AR)
Create & Sign Credential
Return Credential
Workload
Credential Authority
Verifier

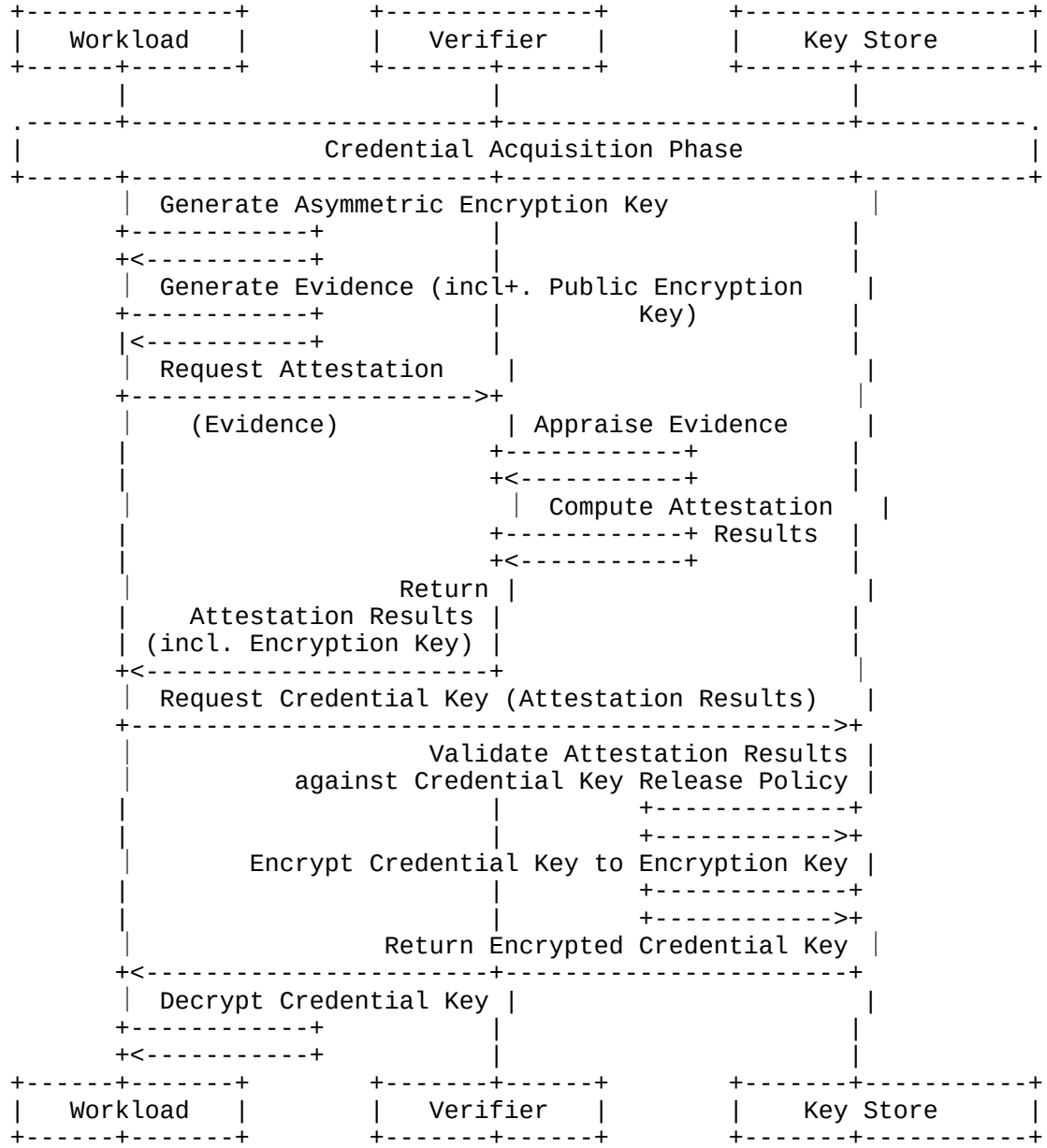
3.4. Mechanisms D

Mechanism D consists of a "Credential Provisioning" phase followed by the "Credential Acquisition" phase.

3.4.1. Credential Provisioning Phase



3.4.2. Credential Acquisition Phase


Workload
Verifier
Key Store
Credential Acquisition Phase
Generate Asymmetric Encryption Key
Generate Evidence (incl. Public Encryption Key)
Request Attestation
(Evidence)
Appraise Evidence
Compute Attestation Results
Return Attestation Results (incl. Encryption Key)
Request Credential Key (Attestation Results)
Validate Attestation Results against Credential Key Release Policy
Encrypt Credential Key to Encryption Key
Return Encrypted Credential Key
Decrypt Credential Key
Workload
Verifier
Key Store

4. Security Considerations

All communications between entities (Workload to Credential Authority, Workload to Verifier etc) MUST be secured using mutually authenticated, confidential, and integrity-protected channels (e.g., TLS).

In addition to the considerations herein, Verifier, which is a central point of anchor for Trustworthy Workload Identifer MUST follow the security guidance detailed in the "Security and Privacy considerations" as detailed in the RATS Architecture Section 11 and Section 12 of [RFC9334].

The credential key MUST always be stored securely at all time, for example in a secure element within the workload.

5. Pivacy Considerations

Remote Attestation of a Workload requires exchange of attestation related messages, for example, Evidence and Attestation Results. This can potentially leak sensitive information about the Workload.

Confidentiality: Encryption could be used to prevent unauthorised parties from accessing sensitive information from Evidence or Attestation Results. This is crucial in multi-tenant environments. The Credential Key to be released to a Workload MUST always be encrypted to avoid potential leakage to unauthorised actors.

6. IANA Considerations

This document has no IANA actions.

7. References

7.1. 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/rfc/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/rfc/rfc8174>.

7.2. Informative References

[I-D.draft-ietf-wimse-arch]
Salowey, J. A., Rosomakho, Y., and H. Tschofenig, "Workload Identity in a Multi System Environment (WIMSE) Architecture", Work in Progress, Internet-Draft, draft-ietf-wimse-arch-06, 30 September 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-wimse-arch-06>.

[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/rfc/rfc9334>.

[RFC9711] Lundblade, L., Mandyam, G., O'Donoghue, J., and C. Wallace, "The Entity Attestation Token (EAT)", RFC 9711, DOI 10.17487/RFC9711, April 2025, <https://www.rfc-editor.org/rfc/rfc9711>.

[TWISIGCharter]
Confidential Computing Consortium Trustworthy Workload Identity SIG, "Trustworthy Workload Identity (TWI) Special Interest Group — Charter", n.d., <https://github.com/confidential-computing/governance/blob/main/SIGs/TWI/TWI_Charter.md>.

[TWISIGDef]
Confidential Computing Consortium Trustworthy Workload Identity SIG, "Trustworthy Workload Identity (TWI) Special Interest Group — Definitions", n.d., <https://github.com/confidential-computing/twi/blob/main/TWI_Definitions.md>.

[TWISIGReq]
Confidential Computing Consortium Trustworthy Workload Identity SIG, "Trustworthy Workload Identity (TWI) Special Interest Group — Requirements", n.d., <https://github.com/confidential-computing/twi/blob/main/TWI_Requirements.md>.

[WIMSES2S] IETF, "WIMSE Service-to-Service Protocol", n.d., <https://datatracker.ietf.org/doc/draft-ietf-wimse-s2s-protocol/>.

Acknowledgments

The following persons, in no specific order, contributed to the work directly, participated in design team meetings, or provided valuable comments during the review of this document.

Pieter Kasselman (SPIRL), Arieal Feldman (Google), Mateusz Bronk (Intel), Manu Fontaine (Hushmesh Inc.), Benedict Lau (EQTY Lab), Zvonko Kaiser (NVIDIA), David Quigley (Intel), Sal Kimmich (GadflyAI), Alex Dalton (Shielded Technologies), Eric Wolfe (Mainsail Industries), Nicolae Paladi(Canary Bit), Mark Gentry (JPMorgan Chase), Jag Raman (Oracle), Brian Hugenbruch (IBM), Jens Alberts (Fr0ntierX), Mira Spina (MITRE) and John Suykerbuyk.

Authors' Addresses

Mark Novak
J.P. Morgan Chase
Email: mark.f.novak@jpmchase.com

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

Henk Birkholz
Franhaufer Inst.
Email: Henk.Birkholz@ietf.contact