RATS
Internet-Draft
Intended status: Standards Track
Expires: 16 September 2026

T. Reddy
Nokia
H. Tschofenig
UniBw M.

15 March 2026

Key Attestation for Entity Attestation Tokens (EAT)
draft-reddy-rats-key-binding-00

Abstract

This document defines a CWT-based Entity Attestation Token (EAT) profile and a new EAT claim that cryptographically bind a private key used to sign a certificate signing request (CSR), or the private key corresponding to an end-entity certificate used for TLS authentication, to an attested execution environment.

The subject public key is conveyed using the EAT cnf claim defined in [RFC8747], and freshness uses the EAT nonce claim defined in [RFC9711]. The proof of possession of the subject key is obtained from the surrounding protocol, such as TLS certificate-based authentication or CSR signature verification. Because the EAT is signed by a hardware-backed Attestation Key (AK), successful verification of the EAT signature together with protocol-level proof of possession establishes a cryptographic binding between the private key and the attested platform state. This mechanism addresses key substitution attacks that arise when attestation evidence and the certificate private keys are validated independently.

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-reddy-rats-key-binding/.

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.

This Internet-Draft will expire on 16 September 2026.

Copyright Notice

Table of Contents

1. Introduction 3
2. Conventions and Definitions 5
3. Key Confirmation and Binding Profile 6
  3.1. Overview 6
  3.2. High-Level Construction 7
4. Proof-of-Possession 8
  4.1. Freshness Requirements 8
  4.2. Verification Procedure 9
5. Claim Definition 10
  5.1. Claim Structure 10
  5.2. Subject Key in cnf 10
  5.3. Protocol-Based PoP 11
  5.4. Key Protection Attributes 11
  5.5. Purpose 12
6. Security Considerations 12
  6.1. Threat Model 12
  6.2. Proof-of-Possession Rationale 13
  6.3. Key Protection Properties 13
  6.4. Key Substitution Attack Illustration 13
  6.5. Mitigation of Key Substitution Attacks 14

6.6. Claim Omission and Downgrade 14
6.7. Scope of Guarantees 14
7. IANA Considerations 15
Acknowledgments 15
Normative References 15
Authors' Addresses 17

1. Introduction

Remote attestation enables an entity to produce attestation evidence that a verifier can use to assess whether the entity's platform state satisfies a required policy. In certificate enrollment and authentication protocols (e.g., TLS), a common security policy requirement is that a private key used by an endpoint must be generated, stored, and protected within a trusted execution environment (TEE) or comparable hardware root of trust.

In certificate enrollment workflows, a Certification Authority (CA) may require attestation evidence demonstrating that the private key corresponding to the public key in a certificate signing request (CSR) is protected by a hardware-backed environment. The LAMPS CSR Attestation specification [I-D.ietf-lamps-csr-attestation] defines mechanisms for including attestation evidence alongside a CSR. In this model, the CA verifies the CSR signature using the public key contained in the request and independently verifies the attestation evidence according to the RATS architecture, using applicable endorsements and trust anchors. However, attestation evidence does not inherently provide a cryptographic proof that the private key used to sign the CSR is the same key that is generated, stored, or protected within the attested environment. The CSR signature demonstrates possession of a private key, and the attestation demonstrates properties of a platform state, but there is no standardized mechanism that cryptographically binds these two validations together. An endpoint could present valid attestation evidence from a protected environment while submitting a certificate signing request (CSR) that is signed with a private key not generated or stored within that environment. In this case, the Certification Authority has no intrinsic cryptographic assurance that the private key corresponding to the CSR public key benefits from the protections described in the attestation evidence.

A similar problem exists in TLS-based scenarios. The TLS Exported Attestation specification [I-D.fossati-tls-exported-attestation] and the TLS Early Attestation specification [I-D.fossati-seat-early-attestation] define mechanisms for conveying attestation evidence within a TLS connection. While the attestation evidence is bound to the TLS connection in these approaches, it does not intrinsically bind the attested environment to the private key

corresponding to the end-entity certificate used for TLS authentication. An endpoint could therefore obtain valid attestation evidence from a protected environment while performing certificate-based TLS authentication using a private key that is not confined to that environment. For example, the TLS private key may reside outside the trusted execution environment and lack the protections claimed by the attestation evidence.

This separation between validation of attestation evidence and validation of certificate private key creates a class of key substitution attacks. In such an attack:

* A valid attestation produced by a genuine hardware-protected environment is presented to the verifier; and

* A private key that is not generated or protected within that environment is used in the CSR or TLS authentication flow.

Because the attestation evidence and the certificate private key are validated independently, the verifier has no intrinsic cryptographic assurance that the operational private key benefits from the protections described in the attestation evidence. This undermines security policy objectives that require the certificate private key to be generated and constrained within the attested environment.

Addressing this problem requires a mechanism that provides both proof of possession of the private key and a cryptographic binding between that key and the attested platform state.

A relying party will also require additional claims describing key protection properties, such as non-exportability or hardware-level protection. For example, [I-D.ietf-rats-pkix-key-attestation] defines an evidence format for reporting properties of cryptographic modules and managed keys in PKIX environments. The PKIX Key Attestation specification [I-D.ietf-rats-pkix-key-attestation] defines attributes including extractable, never-extractable, sensitive, and local that describe protection properties of keys managed by cryptographic modules. These attributes can be conveyed using the key-attributes claim defined in this document while key confirmation itself is conveyed using cnf ([RFC8747]) and protocol-level proof of possession.

Appendix A.1.4 of [RFC9711] illustrates how a key and key store may be represented in evidence. However, the example uses private-use claim labels and does not define standardized key-protection claims. This specification uses the standardized cnf claim from [RFC8747] and defines a new claim for key-protection attributes and usage constraints, while relying on protocol-level proof of possession.

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.

The reader is assumed to be familiar with the vocabulary and concepts defined in the RATS Architecture ([RFC9334]) such as Attester, Relying Party, Verifier.

The reader is assumed to be familiar with common vocabulary and concepts defined in [RFC5280] such as certificate, signature, attribute, verification and validation.

The following terms are used in this document:

Attestation Key (AK): A cryptographic key, typically hardware-backed, used to sign attestation evidence that conveys claims about the state of a platform or trusted execution environment.

Attestation Evidence: Claims about a platform’s state, signed by an Attestation Key, and conveyed in an Entity Attestation Token (EAT).

Entity Attestation Token (EAT): A token format that conveys attestation claims, as defined in [RFC9711]

Subject Key: An asymmetric key pair for which the protection of the private component within an attested execution environment is being asserted. The corresponding public component is conveyed in the EAT cnf claim and compared with the key used in a CSR or TLS end-entity certificate.

Proof of Possession (PoP): Evidence that demonstrates control of the private component of the Subject Key at a given point in time. In this profile, PoP is obtained from the surrounding protocol (for example, TLS certificate-based authentication or CSR signature verification).

Verifier: The entity that validates attestation evidence and evaluates whether the platform state and associated keys satisfy its policy.

Certification Authority (CA): An entity that issues certificates and may require attestation evidence during certificate enrollment.

Trusted Execution Environment (TEE): An isolated execution context that provides confidentiality and integrity protections for code and data, including cryptographic key material.

Key Substitution Attack: An attack in which valid attestation evidence from a protected execution environment is presented to a verifier, while a private key used in a CSR or authentication protocol is not generated or protected within that environment. Because attestation evidence and certificate private key are validated independently, the verifier may lack cryptographic assurance that the certificate private key benefits from the protections claimed in the attestation.

3. Key Confirmation and Binding Profile

3.1. Overview

This document defines a CWT-based EAT profile and a new EAT claim that establish a cryptographic binding between a Subject Key and an attested execution environment. The profile is defined for EAT conveyed as a CWT Claims Set in a COSE_Sign1 message.

The profile provides two properties:

1. Proof of Possession : Demonstration that the entity presenting the attestation controls the private component of the Subject Key.

2. Key-to-Platform Binding : Cryptographic association between the private component of the Subject Key and the attested platform state.

The mechanism combines:

* an EAT signature generated by the Attestation Key (AK); and

* protocol-level proof of possession for the Subject Key.

The subject public key used for protocol-level PoP verification is carried in the EAT cnf claim as a COSE_Key, following [RFC8747]. Because the attestation evidence is authenticated by the AK, and PoP is verified in the surrounding protocol, successful verification establishes that:

* The attested platform state is authentic; and

* The entity participating in the surrounding protocol demonstrates control of the private component of the Subject Key during protocol execution.

This construction creates a cryptographic linkage between the Subject Key and the attested platform state, mitigating Key Substitution Attacks.

The key-attributes claim conveys attributes describing key protection properties and permitted usage of the Subject Key. The key-attributes claim MUST be present and MUST contain at least one member. When present, attributes such as extractable, never-extractable, sensitive, and local MUST follow the semantics defined in [I-D.ietf-rats-pkix-key-attestation]. These attributes enable relying parties to enforce policies such as requiring keys to be generated within the attested environment and prohibiting extraction of private key material.

3.2. High-Level Construction

At a high level, the mechanism binds a certificate private key to an attested execution environment by combining AK-signed attestation evidence and protocol-level proof of possession.

First, the attester constructs an EAT Claims Set including:

* the verifier-provided nonce claim from [RFC9711];

* the cnf claim from [RFC8747] containing the Subject Public Key as COSE_Key;

* the key-attributes claim defined in this document.

Second, the EAT is signed using the Attestation Key.

During verification, three relationships are checked:

* The digitally signed and protected EAT establishes that the platform state is authentic.

* The protocol-level PoP check establishes control of the private component of the Subject Key.

* The public key in cnf is compared with the public key in the CSR or TLS end-entity certificate to ensure they refer to the same operational key.

When these checks succeed, the verifier gains assurance that the certificate private key used in the protocol is the same key whose protection within the attested execution environment is being asserted.

4. Proof-of-Possession

The Proof of Possession (PoP) demonstrates control of the private component of the Subject Key.

In this profile, PoP MUST be verified by the surrounding protocol:

* In certificate enrollment workflows, by validating the CSR signature.

* In TLS workflows, by validating certificate-based TLS authentication.

The public key used for protocol-level PoP verification MUST correspond to the Subject Public Key in EAT cnf.

For this profile, the EAT nonce claim defined in [RFC9711] is the mandatory freshness mechanism. The EAT nonce claim MUST be present and MUST contain a single nonce value supplied by the verifier.

The nonce MUST be supplied by the verifier and MUST be unpredictable and unique within the verifier’s replay window.

The validity period of the key attestation evidence is determined by the lifetime of the enclosing EAT. Verifiers MUST enforce the iat, nbf, and exp claims defined in [RFC9711] to ensure that attestation evidence is not used outside its intended validity window.

4.1. Freshness Requirements

The verifier MUST provide a nonce with sufficient entropy to prevent replay. The nonce is conveyed to the Attester by the Relying Party through the surrounding protocol. The nonce MUST be unpredictable and unique within the verifier’s replay window. The verifier MUST validate that the nonce claim in the EAT matches the nonce it supplied. Failure to include verifier-provided freshness renders the mechanism vulnerable to replay of previously valid attestation evidence. Mechanisms for obtaining and conveying such nonces in certificate enrollment protocols are described in [I-D.ietf-lamps-attestation-freshness].

The verifier-provided nonce is the primary mechanism for ensuring freshness of the attestation evidence. The EAT time-based claims (iat, nbf, and exp) provide an additional validity window for the attestation evidence but do not replace the requirement for a verifier-provided nonce. Verifiers MUST validate both the nonce and the applicable time-based claims when evaluating this profile.

4.2. Verification Procedure

Upon receipt of attestation evidence for this profile, the Verifier MUST perform the following checks:

1. Validate the signature on the EAT using the applicable trust anchors and endorsements for the Attestation Key. If this validation fails, the attestation evidence MUST be rejected.

2. Validate the key-attributes claim. The key-attributes claim MUST be present and MUST contain at least one member.

3. Validate the EAT nonce claim. The EAT nonce claim MUST be present, MUST contain a single nonce value, and MUST match the verifier-supplied nonce.

4. Extract the Subject Public Key from the EAT cnf claim. This profile requires the cnf claim defined in [RFC8747] and requires cnf to contain COSE_Key. Use of Encrypted_COSE_Key or a kid-only representation is outside the scope of this profile.

5. Compare the Subject Public Key contained in cnf with the public key used for protocol-level PoP verification. This public key is either obtained directly from the protocol or supplied to the Verifier by the Relying Party.

    * In certificate enrollment, the public key is obtained from the CSR.

    * In TLS, the public key is obtained from the end-entity certificate used for TLS authentication.

The public key provided to the Verifier MUST correspond to the same key for which protocol-level PoP verification was performed. If the public key parameters do not match, the binding verification MUST fail.

Successful completion of all checks establishes a cryptographic binding between the private component corresponding to the public key used in the CSR or TLS end-entity certificate and the attested execution environment at the time the evidence was generated.

The Verifier conveys the result of the binding verification to the Relying Party as part of the attestation result.

5. Claim Definition

This document defines a new EAT claim named key-attributes that conveys key protection attributes and key-usage constraints for the Subject Key.

5.1. Claim Structure

The claim is defined using CDDL as follows:

```
key-attributes = {
; Optional key protection attributes
? extractable: bool,
? never-extractable: bool,
? sensitive: bool,
? local: bool,

; Optional cryptographic usage constraints
? purpose: [* oid]

}

oid = tstr  ; dotted-decimal OID string
```

The key-attributes claim MUST contain at least one member.

5.2. Subject Key in cnf

This profile uses the EAT cnf claim defined in [RFC8747] to carry the Subject Public Key. The cnf claim MUST be present and MUST contain a COSE_Key member.

When comparing the Subject Public Key contained in cnf with the public key used in a CSR or TLS end-entity certificate, the comparison MUST be performed over the public key parameters rather than over their serialized encodings. This ensures that differences in encoding formats (e.g., ASN.1 DER versus CBOR) do not cause two equivalent public keys to be incorrectly treated as unequal.

For example:

* For RSA keys: The modulus (n) and public exponent (e) MUST match.

* For elliptic curve keys: The curve identifier and public key coordinates (e.g., x and y values) MUST match. If an implementation supports point compression, keys MUST be decompressed to a common format before the comparison is performed.

* For ML-DSA, SLH-DSA, and FN-DSA keys: The comparison MUST be performed over the raw public key byte string defined by the relevant algorithm specification (e.g., FIPS-204 for ML-DSA). In cnf, the Verifier MUST extract the raw public key bytes from the pub parameter of that structure. In an X.509 certificate [RFC5280], the public key is carried in the SubjectPublicKeyInfo structure. The Verifier MUST extract the contents of the subjectPublicKey BIT STRING and obtain the contained public key byte string. The raw public key byte string extracted from cnf and the byte string extracted from the certificate MUST match exactly.

* For other key types: The public key parameters defined by the relevant cryptographic specification MUST match exactly. Comparison based solely on serialized encodings (e.g., raw CBOR, JSON, or DER byte sequences) is NOT RECOMMENDED, as differences in encoding rules may cause equivalent keys to appear unequal.

If the comparison fails, the key-to-platform binding is not established.

5.3. Protocol-Based PoP

This profile relies on protocol-level proof of possession for the Subject Key. This document does not define a new in-token PoP container or signature format.

5.4. Key Protection Attributes

The key-attributes claim includes attributes describing key protection properties of the private component of the Subject Key.

When present, the attributes extractable, never-extractable, sensitive, and local MUST follow the definitions and semantics specified in [I-D.ietf-rats-pkix-key-attestation].

This document does not redefine these attributes. Their interpretation and security semantics are defined in [I-D.ietf-rats-pkix-key-attestation].

A Verifier will evaluate these attributes as part of its security policy when determining whether the Subject Key satisfies requirements for key generation, storage, or exportability.

5.5. Purpose

The purpose parameter identifies the key capabilities associated with the Subject Key.

The value of this parameter is a list of object identifiers (OIDs) identifying the key capabilities defined in [I-D.ietf-rats-pkix-key-attestation].

These OIDs correspond to the key capability identifiers defined in Section 5.2.5 of [I-D.ietf-rats-pkix-key-attestation].

When the key-attributes claim is used in certificate enrollment workflows, the reported key capabilities MUST be compatible with the KeyUsage extensions requested in the CSR and included in the issued certificate.

6. Security Considerations

6.1. Threat Model

This document addresses Key Substitution Attacks. In such attacks, valid attestation evidence from a protected execution environment is presented to a verifier while a private key used in certificate enrollment or TLS authentication is not generated or protected within that environment.

The mechanism defined in this document assumes:

* The AK is securely provisioned and protected.
* The AK correctly signs attestation evidence reflecting the platform state.
* The surrounding protocol correctly verifies proof of possession of the Subject Key private component.
* The verifier provides an unpredictable nonce to ensure freshness.

If these assumptions do not hold, the security guarantees of this mechanism do not apply.

6.2. Proof-of-Possession Rationale

The AK signature over the EAT provides evidence about the Subject Key and its asserted protection properties. Protocol-level PoP verification provides direct evidence of control of the Subject Key.

By requiring both checks, the profile prevents reliance solely on self-reported claims about the presence of the Subject Key in the attested environment.

6.3. Key Protection Properties

The cnf claim establishes a cryptographic binding between the Subject Key and the attestation evidence. However, the cnf claim alone does not convey information about the protection properties of the private component of that key.

Some deployments require assurances regarding how the private key is generated, stored, and protected (for example, whether the key is non-exportable or generated within a hardware-protected environment). The key-attributes claim enables the Verifier to evaluate such properties and determine whether the Subject Key satisfies the security requirements of the deployment.

6.4. Key Substitution Attack Illustration

The following example illustrates how the mechanism detects a Key Substitution Attack.

1. An attacker generates a private key outside the trusted execution environment, denoted as K_bad.

2. The attacker also possesses or obtains attestation evidence for a different key protected within the trusted execution environment, denoted as K_good.

3. The attacker submits a certificate signing request (CSR) signed using K_bad while attaching an EAT containing cnf for K_good.

4. During verification, the Subject Public Key contained in cnf (corresponding to K_good) is compared with the public key contained in the CSR (corresponding to K_bad).

Because the public keys do not match, the binding verification fails. Although the attestation evidence and the CSR signature may each be valid independently, the mismatch prevents the establishment of a cryptographic binding between the certificate private key and the attested execution environment.

6.5. Mitigation of Key Substitution Attacks

This mechanism mitigates Key Substitution Attacks by requiring cryptographic proof that the private key used in a CSR or TLS authentication flow corresponds to the key whose protection within the attested execution environment is being asserted.

Because protocol-level proof of possession is validated together with the EAT signature, and because the claimed Subject Public Key in cnf must match the public key used in the protocol, substitution of a private key that is not generated or protected within the attested execution environment is detected.

If any of these validations fail, the binding is not established.

6.6. Claim Omission and Downgrade

If a security policy requires that the private key corresponding to a certificate be generated or protected within an attested execution environment, the Relying Party MUST ensure that the key-attributes claim defined in this document is present, that the EAT nonce claim is present, that the cnf claim is present with COSE_Key, that protocol-level PoP verification succeeds, and that binding verification succeeds.

In deployments using a separate Verifier, the Relying Party MUST require the Verifier to enforce the presence and successful validation of the key-attributes claim, EAT nonce, cnf with COSE_Key, and protocol-level PoP verification as part of attestation appraisal.

6.7. Scope of Guarantees

This mechanism provides cryptographic evidence that the entity participating in the surrounding protocol demonstrated control of the private component of the Subject Key during protocol execution.

It does not guarantee:

* That the key remains protected after attestation;

* That the key cannot later be exported or migrated;

* That the platform remains in the same state after evidence generation.

Such guarantees depend on platform-specific properties and lifecycle management outside the scope of this document.

7. IANA Considerations

This document requests registration of a new claim in the "CBOR Web Token (CWT) Claims" registry (established by [RFC8392]).

The following value is to be added to this registry:

* Claim Name: key-attributes

* CWT Claim Key: TBD

* Claim Description: Key protection attributes and key-usage constraints associated
with the Subject Key identified by the EAT cnf claim.

* Claim Value Type: CBOR map

* Change Controller: IETF

* Reference: RFCXXXX

Acknowledgments

The authors thank Paul Walters for raising the relay attack threat considered in this document.

Normative References

[I-D.fossati-seat-early-attestation]
Sheffer, Y., Mihalcea, I., Deshpande, Y., Fossati, T., and T. Reddy.K, "Using Attestation in Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", Work in Progress, Internet-Draft, draft-fossati-seat-early-attestation-03, 1 March 2026, <https://datatracker.ietf.org/doc/html/draft-fossati-seat-early-attestation-03>.

[I-D.fossati-tls-exported-attestation]
Fossati, T., Sardar, M. U., Reddy.K, T., Sheffer, Y., Tschofenig, H., and I. Mihalcea, "Remote Attestation with Exported Authenticators", Work in Progress, Internet-Draft, draft-fossati-tls-exported-attestation-02, 3 July 2025, <https://datatracker.ietf.org/doc/html/draft-fossati-tls-exported-attestation-02>.

[I-D.ietf-lamps-attestation-freshness]
Tschofenig, H., Brockhaus, H., Mandel, J., and S. Turner, "Nonce-based Freshness for Remote Attestation in

Certificate Signing Requests (CSRs) for the Certification Management Protocol (CMP), for Enrollment over Secure Transport (EST), and for Certificate Management over CMS (CMC)", Work in Progress, Internet-Draft, draft-ietf-lamps-attestation-freshness-05, 19 October 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-lamps-attestation-freshness-05>.

[I-D.ietf-lamps-csr-attestation]
Ounsworth, M., Tschofenig, H., Birkholz, H., Wiseman, M., and N. Smith, "Use of Remote Attestation with Certification Signing Requests", Work in Progress, Internet-Draft, draft-ietf-lamps-csr-attestation-23, 1 March 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-lamps-csr-attestation-23>.

[I-D.ietf-rats-pkix-key-attestation]
Ounsworth, M., Fiset, J., Tschofenig, H., Birkholz, H., Wiseman, M., and N. Smith, "Evidence Encoding for Hardware Security Modules", Work in Progress, Internet-Draft, draft-ietf-rats-pkix-key-attestation-03, 1 March 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-rats-pkix-key-attestation-03>.

[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>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/rfc/rfc5280>.

[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>.

[RFC8392] Jones, M., Wahlstroem, E., Erdtman, S., and H. Tschofenig, "CBOR Web Token (CWT)", RFC 8392, DOI 10.17487/RFC8392, May 2018, <https://www.rfc-editor.org/rfc/rfc8392>.

[RFC8747] Jones, M., Seitz, L., Selander, G., Erdtman, S., and H. Tschofenig, "Proof-of-Possession Key Semantics for CBOR Web Tokens (CWTs)", RFC 8747, DOI 10.17487/RFC8747, March 2020, <https://www.rfc-editor.org/rfc/rfc8747>.

[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>.

Authors' Addresses

Tirumaleswar Reddy
Nokia
Bangalore
Karnataka
India
Email: k.tirumaleswar_reddy@nokia.com

Hannes Tschofenig
University of the Bundeswehr Munich
Neubiberg
Germany
Email: hannes.tschofenig@gmx.net