Digital Emblems
Internet-Draft
Intended status: Informational
Expires: 19 July 2026

F. Linker
15 January 2026

ADEM Core Specification
draft-linker-diem-adem-core-00

Abstract

In times of armed conflict, the protective emblems of the red cross, red crescent, and red crystal are used to mark physical assets. This enables military units to identify assets as respected and protected under international humanitarian law. This draft specifies the format and trust architecture of a protective, digital emblem to network-connected infrastructure. Such emblems mark assets as protected under IHL analogously to the physical emblems.

About This Document

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

The latest revision of this draft can be found at https://adem-wg.github.io/adem-core-spec/draft-linker-diem-adem-core.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-linker-diem-adem-core/.

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

Source for this draft and an issue tracker can be found at https://github.com/adem-wg/adem-core-spec.

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/.

Copyright Notice

Table of Contents

1. Introduction

International Humanitarian Law (IHL) mandates that military units must not attack medical facilities, such as hospitals. The emblems of the red cross, red crescent, and red crystal are used to mark physical infrastructure (e.g., by a red cross painted on a hospital's rooftop), thereby enabling military units to identify those assets as protected under IHL. This document specifies the structure and trust model of digital emblems for IHL that can be used to mark digital infrastructure as protected under IHL analogously to the physical emblems. We call this system _ADEM_, which stands for an Authentic Digital EMblem.

In ADEM, emblems are signed statements that mark a _assets_ as proteced under IHL. Emblems are issued by _emblem issuers_. Emblem issuer can be authorized by _authorities_. Authorities do so by signing _endorsements_ for emblem issuers. We call both emblems and endorsements _tokens_. Emblems are consumed and validated by _validators_.

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.

Token A token is either an emblem or an endorsement and encoded as a JWS.

Emblem An emblem is a sign of protection under IHL.

Endorsement An endorsement associates a public key with an identity, and hence, resembles the idea of a certificate. When signed by an authority, it attests that the authorized issuer can generally issue claims of protection.

Root Key Organizations control root keys, which identify them cryptographically. Any key of an organization that is endorsed by other parties is a root key.

Asset An asset is a network-connected service that enjoys the specific protections under IHL. Assets must be unambiguously identifiable and unambiguously protected, for example, if they are identified by a domain name that domain name must not be used for services that do not enjoy specific protections under IHL.

Emblem issuer An emblem issuer is an organization entitled to issue claims of protection for their digital infrastructure.

Authority An authority is an organization that is trusted by some to attest a party’s status as protected. This trust may stem from law. For example, nation states or NGOs can take the role of authorities.

Organization An emblem issuer or autority.

Validator A validator is an agent interested in observing and verifying digital emblems.

Beyond these terms, we use the terms "claim" and "header parameter" as references to the JWT specification [RFC7519].

3. Tokens

3.1. Identifiers and their Semantics

Emblems are issued for assets by emblem issuers, which in turn are authorized by authorities. Both emblem issuers and authorities are _organizations_. This section specifies how assets and organizations are identified.

3.1.1. Asset Identifiers

Assets are identified by _asset identifiers_ (AIs). Asset identifiers closely resemble Uniform Resource Identifiers (URIs) as specified in [RFC3986]. However, to limit their scope, we do not follow the specification of URIs and instead define our own syntax.

3.1.1.1. Syntax

Asset identifiers follow the syntax (domain-name, IPv6 defined below):

```
asset-identifier = domain-name | "[" IPv6 "]"
```

Domain names (domain-name) MUST be formatted as usual and specified in [RFC1035] with the exception that the leftmost label MAY be the single-character wildcard "*". In particular, "*" itself is a valid domain name in context of this specification.

IPv6 addresses (IPv6) MUST be formatted following [RFC4291]. IPv6 addresses MUST be global unicast or link-local unicast addresses. Note that the syntax of IPv6 addresses also support IPv4 addresses through "IPv4-Mapped IPv6 Addresses" (cf. [RFC4291], Section 2.5.5.2 (https://www.rfc-editor.org/rfc/rfc4291.html#section-2.5.5.2)).

These are examples of AIs:

* *.example.com

* [2001:0db8::248:1893:25c8:1946]

* [::FFFF:93.184.216.34]

3.1.1.2. Semantics

Several kinds of assets can be identified by asset identifiers:

* Network facing processes, e.g., web servers

* Computational devices both in the virtual sense, e.g., a virtual machine, and in the physical sense, e.g., a laptop

* Networks

An AI identifies a set of IPv4 or IPv6 addresses:

* If the AI is an IPv6 address, it identifies this address only.

* If the AI an IPv6 address prefix, it identifies all IPv6 addresses matching that prefix.

* If the AI is a domain name, it identifies any address for which there is an A or AAAA record for that domain name.

* If the AI is a domain name starting with the wildcard "*", it identifies any address for which there is an A or AAAA record for that domain name or any of its subdomains.

Any process reachable under any of the addresses pointed towards by address and on the port specified (or any port, if unspecified) is pointed by the respective AI.

3.1.1.3. Order

AIs may not only be used for identification but also for constraint purposes. For example, an endorsement may constrain emblems to only signal protection for a specific IP address range. In this section, we define an order on AIs so that one can verify if an identifying AI complies with a constraining AI.

We define an AI A to be _more general_ than an AI B, if all of the following conditions apply:

* If A encodes a domain name and does not contain the wildcard "*", B encodes a domain name, too, and A is equal to B.

* If A encodes a domain name and contains the wildcard "*", B encodes a domain name, too, and B is a subdomain of A excluding the wildcard "*". In this regard, any domain is considered a subdomain of itself.

* If A encodes an IP address, B encodes an IP address, too, and A is a prefix of B.

Note that AIs encoding a domain name are incomparable to AIs encoding IP addresses, i.e., neither can be more general than the other.

3.1.2. Organization Identifiers

Emblems can be associated to an organization. Organizations are identified by URIs, bearing the scheme "https" and a domain name. We call URIs identifying organizations _organization identifiers_ (OIs).

More precisely, an OI has the syntax:

```
organization-identifier = "https://" domain-name
```

Domain names must be formatted as usual, specified in [RFC1035], but always represented in all lower-case. For example, https://example.com is a valid OI, but https://EXAMPLE.COM is not.

3.2. Token Encoding

Tokens MUST be encoded as a JWS [RFC7515] or as an unsecured JWT as defined in [RFC7519], Section 6 (https://datatracker.ietf.org/doc/html/rfc7519#section-6), encoded either in compact serialization or as signed CBOR Web Token (CWT) [RFC8392]. Tokens encoded as JWS MUST only use JWS protected headers and MUST include the jwk header parameter. Any token MUST include the cty (content type) header parameter.

3.2.1. Key Identifiers and Key Formats

Keys are encoded as JSON Web Keys (JWKs) [RFC7517]. In context of ADEM, keys MUST include the alg parameter. We identify keys using their key identifier kid. Whenever a JWK in context of ADEM contains the kid parameter, it MUST be computed as per Section 6.1. See Section 6.1 for an example.

3.2.2. Emblems

An emblem is encoded either as JWS or as an unsecured JWT which signals protection of assets. It is distinguished by the cty header parameter value which MUST be "adem-emb". Its payload includes the JWT claims defined in the table below, following [RFC7519], Section 4.1 (https://datatracker.ietf.org/doc/html/rfc7519#section-4.1). All other registered JWT claims MUST NOT be included.

Claim	Status	Semantics	Encoding
ver	REQUIRED	Version string	"v1"
iat	REQUIRED	As per [RFC7519]	
nbf	REQUIRED	As per [RFC7519]	
exp	REQUIRED	As per [RFC7519]	
iss	RECOMMENDED	Organization signaling protection	OI
assets	REQUIRED	AIs marked a protected	Array of AIs
emb	REQUIRED	Emblem details	JSON object (as follows)

Table 1

Multiple AIs within assets may be desirable, e.g., to include both a asset’s IPv4 and IPv6 address. The claim value of emb MUST be a JSON [RFC8259] object with the following key-value mappings.

Claim	Status	Semantics	Encoding
prp	OPTIONAL	Emblem purposes	Array of purpose (as follows)
dst	OPTIONAL	Permitted distribution channels	Array of distribution-method (as follows)

Table 2

```
purpose = "protective" | "indicative"

distribution-method = "dns" | "icmp" | "udp"
```

3.2.2.1. Example

For example, an emblem might comprise the following header and payload.

Header:

```
{
  "alg": "ES512",
  "jwk": { ... },
  "cty": "adem-emb"
}
```

Payload:

```
{
  "emb": {
    "dst": ["icmp"],
    "prp": ["protective"]
  },
  "iat": 1672916137,
  "nbf": 1672916137,
  "exp": 1675590932,
  "iss": "https://example.com",
  "assets": ["[2001:0db8:582:ae33::29]"]
}
```

3.2.3. Endorsements

Endorsements are encoded as JWSs. Endorsements attest two statements: that a public key is affiliated with an organization, pointed to by OIs, and that this organization is eligible to issue emblems for their assets. They are distinguished by the cty header parameter value which MUST be "adem-end". An endorsement's payload includes the JWT claims defined in the table below. All otger registered JWT claims MUST NOT be included.

Claim	Status	Semantics	Encoding
ver	REQUIRED	Version string	"v1"
iat	REQUIRED	As per [RFC7519]	
nbf	REQUIRED	As per [RFC7519]	
exp	REQUIRED	As per [RFC7519]	
iss	RECOMMENDED	Endorsing organization	OI
sub	RECOMMENDED	Endorsed organization	OI
key	REQUIRED	Endorsed organization's public key	Endorsed key by reference to its kid.
log	OPTIONAL	Root key CT logs	Array (as follows)
end	REQUIRED	Endorsed key can endorse further	Boolean
emb	REQUIRED	Emblem constraints	JSON object (as follows)

Table 3

If an endorsement was signed by a root key, it MUST include log. log maps to an array of JSON objects with the following claims. The semantics of these fields are defined in [RFC6962] for v1 and [RFC9162] for v2.

Claim	Status	Semantics	Encoding
ver	REQUIRED	CT log version	"v1" or "v2"
id	REQUIRED	The CT log's ID	Base64-encoded string
hash	REQUIRED	The binding certificate's leaf hash in the log	Base64-encoded string

Table 4

emb resembles the emblem's emb claim and includes the following claims.

Claim	Status	Semantics	Encoding
prp	OPTIONAL	Purpose constraint	Array of purpose
dst	OPTIONAL	Distribution method constraint	Array of distribution-method
assets	OPTIONAL	Asset constraint	Array of AIs
wnd	OPTIONAL	Maximum emblem lifetime	Integer

Table 5

We say that an endorsement _endorses_ a token if its key claim equals the token's verification key, and its sub claim equals the token's iss claim. We note that the latter includes the possibility of both sub and iss being undefined.

We say that an emblem is _valid_ with respect to an endorsement if all the following conditions apply:

* The endorsement's emb.prp claim is undefined or a superset of the emblem's emb.prp claim.

* The endorsement's emb.dst claim is undefined or a superset of the emblem's emb.dst claim.

* The endorsement’s emb.assets claim is undefined or for each AI within the emblem’s emb.assets claim, there exists an AI within the endorsement’s emb.assets claim which is more general than the emblem’s emb.assets claim.

* The endorsement’s emb.wnd claim is undefined or the sum of emblem’s nbf and the endorsement’s emb.wnd claims is greater than or equal to the emblem’s exp claim.

4. Public Key Commitment

Parties must undeniably link their root public keys to their OI. In this section, we specify the configuration of a emblem issuer’s OI. Root public keys are all public keys which are only endorsed by third parties and never endorsed by the organization itself. A party MAY have multiple root public keys. For a root public key to be configured correctly, there MUST be an X.509 certificate that:

* MUST NOT be revoked

* MUST be logged in the Certificate Transparency logs [RFC6962], [RFC9162]

  - Note that log inclusion requires a valid certificate chain that leads to one of the logs accepted root certificates. Clients are RECOMMENDED to verify that this chain is valid and that none of the certificates along it have been revoked.

* MUST be valid for at least all the following domains (<OI> is understood to be a placeholder for the party’s OI):

  - adem-configuration.<OI>

  - For root public key’s kid <KID> (to be understood as a placeholder): <KID>.adem-configuration.<OI>

We intentionally do not specify how clients should check a certificate’s revocation status. It is RECOMMENDED that clients use offline revocation checks that are provided by major browser vendors, for example, OneCRL or CRLite by Mozilla (https://wiki.mozilla.org/CA/Revocation_Checking_in_Firefox), or CRLSet by Chrome (https://chromium.googlesource.com/playground/chromium-org-site/+/refs/heads/main/Home/chromium-security/crlsets.md).

5. Signs of Protection

A sign of protection is an emblem, accompanied by one or more endorsements. Whenever a token includes OIs (in iss or sub claims), these OIs must be configured accordingly. An OI serves to identify an emblem issuer or authority in the real world. Hence, parties MUST configure the website hosted under their OI to provide sufficient identifying information.

5.1. Verification

Whenever a validator receives an emblem, they MAY check if it is valid. The validity of an emblem is defined with respect to a public key. A validity checking algorithm MUST returns the following values. The order of these values encodes the _strength_ of the verification result.

1. UNSIGNED
2. INVALID
3. SIGNED-UNTRUSTED
4. SIGNED-TRUSTED
5. ORGANIZATIONAL-UNTRUSTED
6. ORGANIZATIONAL-TRUSTED
7. ENDORSED-UNTRUSTED
8. ENDORSED-TRUSTED

Given an input public key and an emblem with a set of endorsements, a verification algorithm takes the following steps:

1. If the emblem does not bear a signature, return UNSIGNED.
2. Run the _signed emblem verification procedure_ (Section 6.2; results in one of SIGNED-TRUSTED, SIGNED-UNTRUSTED, or INVALID).
3. If previous procedure resulted in INVALID or the emblem does not include the iss claim, return the last verification procedure’s result and the emtpy set of OIs.
4. Run the _organizational emblem verification procedure_ (Section 6.3; results in one of ORGANIZATIONAL-TRUSTED, ORGANIZATIONAL-UNTRUSTED, INVALID).

5. If the previous procedure resulted in INVALID return INVALID and the empty set of OIs.

6. If all tokens include the same iss claim, return the strongest return value matching *-TRUSTED, the strongest return value matching *-UNTRUSTED provided that it is strictly stronger than the strongest return value matching *-TRUSTED, and the empty set of OIs.

7. Run the _endorsed emblem verification procedure_ (Section 6.4; results in a set of OIs and one of ENDORSED-TRUSTED, ENDORSED-UNTRUSTED, INVALID).

8. If the previous procedure resulted in INVALID return INVALID and the empty set of OIs.

9. Return the strongest return value matching *-TRUSTED, the strongest return value matching *-UNTRUSTED provided that it is strictly stronger than the strongest return value matching *-TRUSTED, and the set of OIs returned by the endorsed emblem verification procedure.

Note that the endorsed emblem verification procedure resulting in INVALID is handled implicitly in step 8. As the procedure did not terminate in step 5, organizational verification must have been successful. Hence, INVALID cannot be the strongest return value, and an emblem not being accompanied by valid endorsements are downgraded to organizational emblems.

The set of OIs returned by the verification procedure encodes the OIs of endorsing parties where verification passed.

5.1.1. Comments on Trust Policies

We strongly RECOMMEND against accepting emblems resulting in SIGNED-UNTRUSTED. In such cases, validators should aim to authenticate the respective public keys via other, out-of-band methods. This effectively lifts the result to SIGNED-TRUSTED. Signed emblems are supported for cases of emergency where an emblem issuer is able to communicate one or more public key, but might not be able to set up a signing infrastructure linking their assets to a root key.

There is no definite guideline on how to choose which keys to trust, i.e., which keys to pass as trusted public key to the verification procedure. Some validators may have pre-existing trust relationships with some authorities, e.g., military units of a nation state could use the public keys of their nation state or allies. Other validators might be fine with fetching public keys authenticated only by the web PKI.

5.2. Protection

An emblem for which the verification procedure produces a result other than INVALID marks any asset whose address is identified by at least one of the emblem’s AIs. Such an emblem signals that the respective asset is enjoys the specific protections of IHL.

Emblem issuers MUST only issue emblems for assets that are used only for protected purposes.

6. Algorithms

6.1. JWK Hashing

Context:

* Input: A JWK public key as per [RFC7517] in arbitrary encoding.

* Output: A hash of the JWK

Algorithm:

1. Parse the JWK as JSON object.

2. Drop the kid parameter from the JWK.

3. Compute the key’s thumbprint using SHA-256 as per [RFC7638].

4. Return the digest in base32 encoding as per [RFC4648] in all lower-case and with trailing = removed.

6.2. Signed Emblem Verification Procedure

Context:

* Input: An emblem, a set of endorsements, and a trusted public key.

* Output: SIGNED-TRUSTED, SIGNED-UNTRUSTED, or INVALID.

Algorithm:

1. Ignore all endorsements including an iss claim different to the emblem’s iss claim. A defined iss claim is understood to be different to an undefined iss claim.

2. Verify every signature.

3. Verify that all endorsements form a consecutive chain where there is a unique root endorsement and the public key which verifies the emblem is transitively endorsed by that root endorsement.

4. Verify that no endorsement expired.

5. Verify that all endorsements bear the claim end=true except for the emblem signing key’s endorsement.

6. Verify that the emblem is valid with regard to every endorsement.

7. If any of the aforementioned verification steps fail, return INVALID. If there is a token signed by the trusted input public key, return SIGNED-TRUSTED. Otherwise, return SIGNED-UNTRUSTED.

Distribution methods MAY indicate an order of tokens to guide clients assembling the chain of endorsements in step 3. Whenever such an order is specified, clients MAY immediately reject a set of tokens as invalid if the indicated order does not yield a valid chain of endorsements.

6.3. Organizational Emblem Verification Procedure

Context:

* Assumptions: Signed emblem verification has been performed and did not return INVALID. Every token as part of the input includes the iss claim.

* Input: An emblem, a set of endorsements, and a trusted public key.

* Output: ORGANIZATIONAL-TRUSTED, ORGANIZATIONAL-UNTRUSTED, or INVALID.

Algorithm:

1. Ignore all endorsements including an iss claim different to the emblem’s iss claim.

2. Verify that the top-most endorsement’s iss claim value (its OI) is configured correctly as specified in Section 4.

3. If the aforementioned verification step fails, return INVALID. If the top-most endorsing key is equal to the trusted input public key, return ORGANIZATIONAL-TRUSTED. Otherwise, return ORGANIZATIONAL-UNTRUSTED.

6.4. Endorsed Emblem Verification Procedure

Context:

* Assumptions: Organizational emblem verification has been performed and did not return INVALID. There are emblems as part of the input including an iss claim different to the emblem’s iss claim.

* Input: An emblem, a set of endorsements, and a trusted public key.

* Output: ENDORSED-TRUSTED, ENDORSED-UNTRUSTED, or INVALID, and a set of OIs.

Algorithm:

1. Ignore all endorsements including an iss claim equal to the emblem’s iss claim.

2. For every endorsement:

    1. Verify its signature.

    2. Verify that it endorses the top-most endorsing key with the same iss claim as the emblem.

    3. Verify that it did not expire.

    4. Verify that it bears the claim end=true.

    5. Verify that the emblem is valid with regard to this endorsement.

    6. Implementations SHOULD verify that the endorsement’s iss claim value (its OI) is configured correctly as specified in Section 4.

    7. Should any of the aforementioned verification steps fail, ignore this endorsement.

3. If there are no endorsements remaining after the last step, return INVALID and the empty set of OIs. If in the set of remaining endorsements, there is an endorsement with a verification key equal to the trusted input public key, return

ENDORSED-TRUSTED. Otherwise, return ENDORSED-UNTRUSTED. In both the latter cases, also return the set of all iss claims of the remaining endorsements.

7. Security Considerations

7.1. No Endorsements without iss

The procedures to verify organizational or endorsed emblems as specified in Section 6.3 and Section 6.4 assume that the emblem's iss claim is defined. Practically speaking, this implies that parties can only go beyond pure public key authentication (where public keys need to be authenticated out-of-band) by stating an OI.

The constraints on well-configured OIs offers two beneficial security properties:

* Parties cannot equivocate their keys, i.e., they need to commit to a consistent set of keys.

* Parties cannot deny having used certain root public keys.

These properties stem from parties needing to include a hash of their key in a TLS certificate, and consequently, in certificate transparency logs.

7.2. Token Order

As specified in Section 6.2, clients MAY reject sets of tokens as invalid if the order of tokens as indicated by the sending client does not yield a valid chain of endorsements. This allows an adversary to force rejection of a set of tokens by altering, e.g., sequence numbers on non-integrity protected channels such as UDP.

However, this does not constitute a new attack. Such adversaries could flip a bit in the emblem's signature, rendering the set of tokens invalid, too.

7.3. Key Identifiers

Key identifiers were designed such that they commit to the identified key, i.e., key identifiers must provide strong collision-resistance. This is ensured by computing it using SHA-256.

8. IANA Considerations

This document has no IANA actions.

9. Normative References

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/rfc/rfc1035>.

[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>.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <https://www.rfc-editor.org/rfc/rfc3986>.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <https://www.rfc-editor.org/rfc/rfc4291>.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <https://www.rfc-editor.org/rfc/rfc4648>.

[RFC6962] Laurie, B., Langley, A., and E. Kasper, "Certificate Transparency", RFC 6962, DOI 10.17487/RFC6962, June 2013, <https://www.rfc-editor.org/rfc/rfc6962>.

[RFC7515] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Signature (JWS)", RFC 7515, DOI 10.17487/RFC7515, May 2015, <https://www.rfc-editor.org/rfc/rfc7515>.

[RFC7517] Jones, M., "JSON Web Key (JWK)", RFC 7517, DOI 10.17487/RFC7517, May 2015, <https://www.rfc-editor.org/rfc/rfc7517>.

[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/rfc/rfc7519>.

[RFC7638] Jones, M. and N. Sakimura, "JSON Web Key (JWK) Thumbprint", RFC 7638, DOI 10.17487/RFC7638, September 2015, <https://www.rfc-editor.org/rfc/rfc7638>.

[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>.

[RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <https://www.rfc-editor.org/rfc/rfc8259>.

[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>.

[RFC9162] Laurie, B., Messeri, E., and R. Stradling, "Certificate Transparency Version 2.0", RFC 9162, DOI 10.17487/RFC9162, December 2021, <https://www.rfc-editor.org/rfc/rfc9162>.

Author’s Address

Felix Linker
Email: linkerfelix@gmail.com