

Remote Attestation Procedures  
Internet-Draft  
Intended status: Standards Track  
Expires: 13 April 2023

C. Wallace  
Red Hound  
R. Housley  
Vigil Security  
T. Fossati  
Y. Deshpande  
arm  
10 October 2022

Concise TA Stores (CoTS)  
draft-wallace-rats-concise-ta-stores-01

## Abstract

Trust anchor (TA) stores may be used for several purposes in the Remote Attestation Procedures (RATS) architecture including verifying endorsements, reference values, digital letters of approval, attestations, or public key certificates. This document describes a Concise Reference Integrity Manifest (CoRIM) extension that may be used to convey optionally constrained trust anchor stores containing optionally constrained trust anchors in support of these purposes.

## 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-wallace-rats-concise-ta-stores/>.

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 13 April 2023.

#### Copyright Notice

Copyright (c) 2022 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 . . . . .	3
1.1. Constraints . . . . .	4
2. Conventions and Definitions . . . . .	4
3. Trust anchor management for RATS . . . . .	5
3.1. TA and CA conveyance . . . . .	5
3.1.1. The concise-ta-stores Container . . . . .	6
3.1.2. The concise-ta-store-map Container . . . . .	6
3.1.3. The cas-and-tas-map Container . . . . .	8
3.2. Environment definition . . . . .	9
3.2.1. The environment-group-list Array . . . . .	9
3.2.2. The abbreviated-swid-tag-map Container . . . . .	10
3.2.3. The named-ta-store Type . . . . .	11
3.3. Constraints definition . . . . .	11
3.3.1. The \$tas-list-purpose Type . . . . .	11
3.3.2. Claims . . . . .	12
3.4. Processing a concise-ta-stores RIM . . . . .	12
3.5. Verifying a concise-ta-stores RIM . . . . .	12
4. CDDL definitions . . . . .	12
5. Examples . . . . .	14
6. Security Considerations . . . . .	21
7. IANA Considerations . . . . .	21
7.1. CoRIM CBOR Tag Registration . . . . .	21
8. References . . . . .	21
8.1. Normative References . . . . .	21
8.2. Informative References . . . . .	22

Acknowledgments . . . . .	23
Appendix A . . . . .	23
Authors' Addresses . . . . .	25

## 1. Introduction

The RATS architecture [I-D.draft-ietf-rats-architecture] uses the definition of a trust anchor from [RFC6024]: "A trust anchor represents an authoritative entity via a public key and associated data. The public key is used to verify digital signatures, and the associated data is used to constrain the types of information for which the trust anchor is authoritative." In the context of RATS, a trust anchor may be a public key or a symmetric key. This document focuses on trust anchors that are represented as public keys.

The Concise Reference Integrity Manifest (CoRIM) [I-D.draft-birkholz-rats-corim] specification defines a binary encoding for reference values using the Concise Binary Object Representation (CBOR) [RFC8949]. Amongst other information, a CoRIM may include key material for use in verifying evidence from an attesting environment (see section 3.11 in [I-D.draft-birkholz-rats-corim]). The extension in this document aims to enable public key material to be decoupled from reference data for several reasons, described below.

Trust anchor (TA) and certification authority (CA) public keys may be less dynamic than the reference data that comprises much of a reference integrity manifest (RIM). For example, TA and CA lifetimes are typically fairly long while software versions change frequently. Conveying keys less frequently and independent from reference data enables a reduction in size of RIMs used to convey dynamic information and may result in a reduction in the size of aggregated data transferred to a verifier. CoRIMs themselves are signed and some means of conveying CoRIM verification keys is required, though ultimately some out-of-band mechanism is required at least for bootstrapping purposes. Relying parties may verify attestations from both hardware and software sources and some trust anchors may be used to verify attestations from both hardware and software sources, as well. The verification information included in a CoRIM optionally includes a trust anchor, leaving trust anchor management to other mechanisms. Additionally, the CoRIM verification-map structure is tied to CoMIDs, leaving no simple means to convey verification information for CoSWIDs [I-D.draft-ietf-sacm-coswid].

This document defines means to decouple TAs and CAs from reference data and adds support for constraining the use of trust anchors, chiefly by limiting the environments to which a set of trust anchors is applicable. This constraints mechanism is similar to that in

[fido-metadata] and [fido-service] and should align with existing attestation verification practices that tend to use per-vendor trust anchors. TA store instances may be further constrained using coarse-grained purpose values or a set of finer-grained permitted or excluded claims. The trust anchor formats supported by this draft allow for per-trust anchor constraints, if desired. Conveyance of trust anchors is the primary goal, CA certificates may optionally be included for convenience.

### 1.1. Constraints

This document aims to support different PKI architectures including scenarios with various combinations of the following characteristics:

- \* TA stores that contain a TA or set of TAs from a single organization
- \* TA stores that contain a set of TAs from multiple organizations
- \* TAs that issue certificates to CAs within the same organization as the TA
- \* TAs that issue certificates to CAs from multiple organizations
- \* CAs that issue certificates that may be used to verify attestations or certificates from the same organization as the TA and CA
- \* CAs that issue certificates that may be used to verify attestations or certificates from multiple organizations

Subsequent specifications may define extensions to express constraints as well as processing rules for evaluating constraints expressed in TA stores, TAs, CA certificates and end entity (EE) certificates. Support for constraints is intended to enable misissued certificates to be rejected at verification time. Any public key that can be used to verify a certificate is assumed to also support verification of revocation information, subject to applicable constraints defined by the revocation mechanism.

## 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.

### 3. Trust anchor management for RATS

Within RATS, trust anchors may be used to verify digital signatures for a variety of objects, including entity attestation tokens (EATs), CoRIMs, X.509 CA certificates (possibly containing endorsement information), X.509 EE certificates (possibly containing endorsement or attestation information), other attestation data, digital letters of approval [dloa], revocation information, etc. Depending on context, a raw public key may suffice or additional information may be required, such as subject name or subject public key identifier information found in an X.509 certificate. Trust anchors are usually aggregated into sets that are referred to as "trust anchor stores". Different trust anchor stores may serve different functional purposes.

Historically, trust anchors and trust anchor stores are not constrained other than by the context(s) in which a trust anchor store is used. The path validation algorithm in [RFC5280] only lists name, public key, public key algorithm and public key parameters as the elements of "trust anchor information". However, there are environments that do constrain trust anchor usage. The RPKI uses extensions from trust anchor certificates as defined in [RFC3779]. FIDO provides a type of constraint by grouping attestation verification root certificates by authenticator model in [fido-metadata].

This document aims to support each of these types of models by allowing constrained or unconstrained trust anchors to be grouped by abstract purpose, i.e., similar to traditional trust anchor stores, or grouped by a set of constraints, such as vendor name.

#### 3.1. TA and CA conveyance

An unsigned concise TA stores object is a list of one or more TA stores, each represented below as a concise-ta-store-map element.

```
concise-ta-stores
  concise-ta-store-map #1
  ...
  concise-ta-store-map #n
```

Each TA store instance identifies a target environment and features one or more public keys. Optional constraints on usage may be defined as well.

```
concise-ta-store-map
  language
  store-identity
  target environment
  abstract coarse-grained constraints on TA store usage
  concrete fine-grained constraints on TA store usage
  public keys (possibly included per-instance constraints)
```

The following sections define the structures to support the concepts shown above.

#### 3.1.1. The concise-ta-stores Container

The concise-ta-stores type is the root element for distributing sets of trust anchor stores. It contains one or more concise-ta-store-map elements where each element in the list identifies the environments for which a given set of trust anchors is applicable, along with any constraints.

```
concise-ta-stores = [+ concise-ta-store]
```

The \$concise-tag-type-choice [I-D.draft-birkholz-rats-corim] is extended to include the concise-ta-stores structure. As shown in Section 4 of [I-D.draft-birkholz-rats-corim], the \$concise-tag-type-choice type is used within the unsigned-corim-map structure, which is used within COSE-Sign1-corim structure. The COSE-Sign1-corim provides for integrity of the CoTS data. CoTS structures are not intended for use as stand-alone, unsigned structures. The signature on a CoTS instance SHOULD be verified using a TA associated with the cots purpose (Section 3.3.1).

```
$concise-tag-type-choice /= #6.TBD(bytes .cbor concise-ta-stores)
```

#### 3.1.2. The concise-ta-store-map Container

A concise-ta-store-map is a trust anchor store where the applicability of the store is established by the tastore.environment field with optional constraints on use of trust anchors found in the tastore.keys field defined by the tastore.purpose, tastore.perm\_claims and tastore.excl\_claims fields.

```
concise-ta-store-map = {  
  ? tastore.language => language-type  
  ? tastore.store-identity => tag-identity-map  
  tastore.environments => environment-group-list  
  ? tastore.purposes => [+ $$tas-list-purpose]  
  ? tastore.perm_claims => [+ $$claims-set-claims]  
  ? tastore.excl_claims => [+ $$claims-set-claims]  
  tastore.keys => cas-and-tas-map  
}  
  
; concise-ta-store-map indices  
tastore.language = 0  
tastore.store-identity = 1  
tastore.environment = 2  
tastore.purpose = 3  
tastore.perm_claims = 4  
tastore.excl_claims = 5  
tastore.keys = 6
```

The following describes each member of the concise-ta-store-map.

**tastore.language:** A textual language tag that conforms with the IANA Language Subtag Registry [IANA.language-subtag-registry].

**tastore.store-identity:** A composite identifier containing identifying attributes that enable global unique identification of a TA store instance across versions and facilitate linking from other artifacts. The tag-identity-map type is defined in [I-D.draft-birkholz-rats-corim].

**tastore.environment:** A list of environment definitions that limit the contexts for which the tastore.keys list is applicable. If the tastore.environment is empty, TAs in the tastore.keys list may be used for any environment.

**tastore.purpose:** Contains a list of purposes (Section 3.3.1) for which the tastore.keys list may be used. When absent, TAs in the tastore.keys list may be used for any purpose. This field is similar to the extendedKeyUsage extension defined in [RFC5280]. The initial list of purposes are: cots, corim, comid, coswid, eat, key-attestation, certificate

**tastore.perm\_claims:** Contains a list of claim values (Section 3.3.2) [I-D.draft-ietf-rats-eat] for which tastore.keys list MAY be used to verify. When this field is absent, TAs in the tastore.keys list MAY be used to verify any claim subject to other restrictions.

`tastore.excl_claims`: Contains a list of claim values (Section 3.3.2) [I-D.draft-ietf-rats-eat] for which `tastore.keys` list MUST NOT be used to verify. When this field is absent, TAs in the `tastore.keys` list may be used to verify any claim subject to other restrictions.

`tastore.keys`: Contains a list of one or more TAs and an optional list of one or more CA certificates.

The `perm_claims` and `excl_claims` constraints MAY alternatively be expressed as extensions in a TA or CA. Inclusion of support here is intended as an aid for environments that find CBOR encoding support more readily available than DER encoding support.

### 3.1.3. The `cas-and-tas-map` Container

The `cas-and-tas-map` container provides the means of representing trust anchors and, optionally, CA certificates.

```
trust-anchor = [
  format => $pkix-ta-type
  data => bstr
]

cas-and-tas-map = {
  tastore.tas => [ + trust-anchor ]
  ? tastore.cas => [ + pkix-cert-data ]
}

; cas-and-tas-map indices
tastore.tas = 0
tastore.cas = 1

; format values
$pkix-ta-type /= tastore.pkix-cert-type
$pkix-ta-type /= tastore.pkix-tainfo-type
$pkix-ta-type /= tastore.pkix-spki-type

tastore.pkix-cert-type = 0
tastore.pkix-tainfo-type = 1
tastore.pkix-spki-type = 2

; certificate type
pkix-cert-data = bstr
```

The `tastore.tas` element is used to convey one or more trust anchors and an optional set of one or more CA certificates. TAs are implicitly trusted, i.e., no verification is required prior to use.



However, limitations on the use of the TA may be asserted in the corresponding concise-ta-store-map or within the TA itself. The `tastore.cas` field provides certificates that may be useful in the context where the corresponding concise-ta-store-map is used. These certificates are not implicitly trusted and MUST be validated to a trust anchor before use. End entity certificates SHOULD NOT appear in the `tastore.cas` list.

The structure of the data contained in the `data` field of a trust-anchor is indicated by the `format` field. The `pkix-cert-type` is used to represent a binary, DER-encoded X.509 Certificate as defined in section 4.1 of [RFC5280]. The `pkix-key-type` is used to represent a binary, DER-encoded SubjectPublicKeyInfo as defined in section 4.1 of [RFC5280]. The `pkix-tainfo-type` is used to represent a binary, DER-encoded TrustAnchorInfo as defined in section 2 of [RFC5914].

The `$pkix-ta-type` provides an extensible means for representing trust anchor information. It is defined here as supporting the `pkix-cert-type`, `pkix-spki-type` or `pkix-tainfo-type`. The `pkix-spki-type` may be used where only a raw public key is necessary. The `pkix-cert-type` may be used for most purposes, including scenarios where a raw public key is sufficient and those where additional information from a certificate is required. The `pkix-tainfo-type` is included to support scenarios where constraints information is directly associated with a public key or certificate (vs. constraints for a TA set as provided by `tastore.purpose`, `tastore.perm_claims` and `tastore.excl_claims`).

The `pkix-cert-data` type is used to represent a binary, DER-encoded X.509 Certificate.

### 3.2. Environment definition

#### 3.2.1. The environment-group-list Array

In CoRIM, "composite devices or systems are represented by a collection of Concise Module Identifiers (CoMID) and Concise Software Identifiers (CoSWID)". For trust anchor management purposes, targeting specific devices or systems may be too granular. For example, a trust anchor or set of trust anchors may apply to multiple device models or versions. The environment-map definition as used in a CoRIM is tightly bound to a CoMID. To allow for distribution of key material applicable to a specific or range of devices or software, the environment-group-list and environment-group-map are defined as below. These aim to enable use of coarse-grained naturally occurring values, like vendor, make, model, etc. to determine if a set of trust anchors is applicable to an environment.

```
environment-group-list = [* environment-group-list-map]

environment-group-list-map = {
  ? tastore.environment_map => environment-map,
  ? tastore.concise_swid_tag => abbreviated-swid-tag,
  ? tastore.named_ta_store => named-ta-store,
}

; environment-group-list-map indices
tastore.environment_map = 0
tastore.abbreviated_swid_tag = 1
tastore.named_ta_store = 2
```

An environment-group-list is a list of one or more environment-group-list-map elements that are used to determine if a given context is applicable. An empty list signifies all contexts SHOULD be considered as applicable.

An environment-group-list-map is one of environment-map[I-D.draft-birkholz-rats-corim], abbreviated-swid-tag-map (Section 3.2.2) or named-ta-store (Section 3.2.3).

As defined in [I-D.draft-birkholz-rats-corim], an environment-map may contain class-map, \$instance-id-type-choice, \$group-id-type-choice.

QUESTION: Should the above dispense with environment\_map and concise\_swid\_tag and use or define some identity-focused structure with information common to both (possibly class-map from [I-D.draft-birkholz-rats-corim])? If not, should a more complete CoMID representation be used (instead of environment\_map)?

### 3.2.2. The abbreviated-swid-tag-map Container

The abbreviated-swid-tag-map allows for expression of fields from a concise-swid-tag [I-D.draft-ietf-sacm-coswid] with all fields except entity designated as optional, compared to the concise-swid-tag definition that requires tag-id, tag-version and software-name to be present.

```

abbreviated-swid-tag-map = {
  ? tag-id => text / bstr .size 16,
  ? tag-version => integer,
  ? corpus => bool,
  ? patch => bool,
  ? supplemental => bool,
  ? software-name => text,
  ? software-version => text,
  ? version-scheme => $version-scheme,
  ? media => text,
  ? software-meta => one-or-more<software-meta-entry>,
  entity => one-or-more<entity-entry>,
  ? link => one-or-more<link-entry>,
  ? payload-or-evidence,
  * $$coswid-extension,
  global-attributes,
}

```

### 3.2.3. The named-ta-store Type

This specification allows for defining sets of trust anchors that are associated with an arbitrary name instead of relative to information typically expressed in a CoMID or CoSWID. Relying parties MUST be configured using the named-ta-store value to select a corresponding concise-ta-store-map for use.

```
named-ta-store = tstr
```

## 3.3. Constraints definition

### 3.3.1. The \$\$tas-list-purpose Type

The \$\$tas-list-purpose type provides an extensible means of expressions actions for which the corresponding keys are applicable. For example, trust anchors in a concise-ta-store-map with purpose field set to eat may not be used to verify certification paths. Extended key usage values corresponding to each purpose listed below (except for certificate) are defined in a companion specification.

```

$$tas-list-purpose /= "cots"
$$tas-list-purpose /= "corim"
$$tas-list-purpose /= "coswid"
$$tas-list-purpose /= "eat"
$$tas-list-purpose /= "key-attestation"
$$tas-list-purpose /= "certificate"
$$tas-list-purpose /= "dloa"

```

TODO - define verification targets for each purpose. QUESTION - should this have a registry?

### 3.3.2. Claims

A concise-ta-store-map may include lists of permitted and/or excluded claims [I-D.draft-ietf-rats-eat] that limit the applicability of trust anchors present in a cas-and-tas-map. A subsequent specification will define processing rules for evaluating constraints expressed in TA stores, TAs, CA certificates and end entity certificates.

### 3.4. Processing a concise-ta-stores RIM

When verifying a signature using a public key that chains back to a concise-ta-stores instance, elements in the concise-ta-stores array are processed beginning with the first element and proceeding until either a matching set is found that serves the desired purpose or no more elements are available. Each element is evaluated relative to the context, i.e., environment, purpose, artifact contents, etc.

For example, when verifying a CoRIM, each element in a triples-group MUST have an environment value that matches an environment-group-list-map element associated with the concise-ta-store-map containing the trust anchor used to verify the CoMID. Similarly, when verifying a CoSWID, the values in a abbreviated-swid-tag element from the concise-ta-store-map MUST match the CoSWID tag being verified. When verifying a certificate with DICE attestation extension, the information in each DiceTcbInfo element MUST be consistent with an environment-group-list-map associated with the concise-ta-store-map.

### 3.5. Verifying a concise-ta-stores RIM

[I-D.draft-birkholz-rats-corim] defers verification rules to [RFC8152] and this document follows suit with the additional recommendation that the public key used to verify the RIM SHOULD be present in or chain to a public key present in a concise-ta-store-map with purpose set to cots.

## 4. CDDL definitions

The CDDL definitions present in this document are provided below. Definitions from [I-D.draft-birkholz-rats-corim] are not repeated here.

```
concise-ta-stores = [+ concise-ta-store-map]
$concise-tag-type-choice /= #6.TBD(bytes .cbor concise-ta-stores)

concise-ta-store-map = {
  ? tastore.language => language-type
  ? tastore.store-identity => tag-identity-map
  tastore.environments => environment-group-list
  ? tastore.purposes => [+ $$tas-list-purpose]
  ? tastore.perm_claims => [+ $$claims-set-claims]
  ? tastore.excl_claims => [+ $$claims-set-claims]
  tastore.keys => cas-and-tas-map
}

; concise-ta-store-map indices
tastore.language = 0
tastore.store-identity = 1
tastore.environment = 2
tastore.purpose = 3
tastore.perm_claims = 4
tastore.excl_claims = 5
tastore.keys = 6

trust-anchor = [
  format => $pkix-ta-type
  data => bstr
]

cas-and-tas-map = {
  tastore.tas => [ + trust-anchor ]
  ? tastore.cas => [ + pkix-cert-type ]
}

; cas-and-tas-map indices
tastore.tas = 0
tastore.cas = 1

; format values
$pkix-ta-type /= tastore.pkix-cert-type
$pkix-ta-type /= tastore.pkix-tainfo-type
$pkix-ta-type /= tastore.pkix-spki-type

tastore.pkix-cert-type = 0
tastore.pkix-tainfo-type = 1
tastore.pkix-spki-type = 2

; certificate type
pkix-cert-data = bstr
```

```

environment-group-list = [* environment-group-list-map]

environment-group-list-map = {
  ? environment-map => environment-map,
  ? concise-swid-tag => abbreviated-swid-tag,
  ? named-ta-store => named-ta-store,
}

abbreviated-swid-tag = {
  ? tag-version => integer,
  ? corpus => bool,
  ? patch => bool,
  ? supplemental => bool,
  ? software-name => text,
  ? software-version => text,
  ? version-scheme => $version-scheme,
  ? media => text,
  ? software-meta => one-or-more<software-meta-entry>,
  ? entity => one-or-more<entity-entry>,
  ? link => one-or-more<link-entry>,
  ? payload-or-evidence,
  * $$coswid-extension,
  global-attributes,
}

named-ta-store = tstr

$tas-list-purpose /= "cots"
$tas-list-purpose /= "corim"
$tas-list-purpose /= "comid"
$tas-list-purpose /= "coswid"
$tas-list-purpose /= "eat"
$tas-list-purpose /= "key-attestation"
$tas-list-purpose /= "certificate"
$tas-list-purpose /= "dloa"

```

## 5. Examples

The following examples are isolated concise-ta-store-map instances shown as JSON for ease of reading. The final example is an ASCII hex representation of a CBOR-encoded concise-ta-stores instance containing each example below (and using a placeholder value for the concise-ta-stores tag).

The TA store below contains a TA from a single organization ("Zesty Hands, Inc,") that is used to verify CoRIMs for that organization. Because this TA does not verify certificates, a bare public key is appropriate. It features a tag identity field containing a UUID for the tag identity and a version indication.

```
{
  "tag-identity": {
    "id": "ab0f44b1-bfdc-4604-ab4a-30f80407ebcc",
    "version": 5
  },
  "environments": [
    {
      "environment": {
        "class": {
          "vendor": "Worthless Sea, Inc."
        }
      }
    }
  ],
  "keys": {
    "tas": [
      {
        "format": 2,
        "data":
"MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAErYoMAdqe2gJT3CvCcifZxyE9+
N8T6Jy5zbeo5LYtnOipmilwXA9/gNtlwAbRCRQitH/GEcvUaG1zPZxIOITV/g=="
      }
    ]
  }
}
```

The TA store below features three TAs from different organizations grouped as a TA store with the name "Miscellaneous TA Store". The first TA is an X.509 certificate. The second and third TAs are TrustAnchorInfo objects containing X.509 certificates. Though not shown in this example, constraints could be added to the TrustAnchorInfo elements, i.e., to restrict verification to attestations asserting a specific vendor name. It features a tag identity field containing a string as the tag identity with no version field present.

```
{
  "tag-identity": {
    "id": "some_tag_identity"
  },
  "environments": [
    {
```

```

    "namedtastore": "Miscellaneous TA Store"
  },
  "keys": {
    "tas": [
      {
        "format": 0,
        "data":
"
MIIBvTCCAWSgAwIBAgIVANCdkL89U1zHc9Ui7XfVniK7pFuIMAOGCCqGSM49BAMCMD4
xCzAJBgNVBAYMALVTMRAwDgYDVQQKDAdFeGFtcGx1MR0wGwYDVQQDDBRFeGFtcGx1IF
RydXN0IEFuY2hvcjAeFw0yMjA1MTkxNTEzMdDaFw0zMjA1MTYxNTEzMdDaMD4xCzAJB
gNVBAYMALVTMRAwDgYDVQQKDAdFeGFtcGx1MR0wGwYDVQQDDBRFeGFtcGx1IFRydXN0
IEFuY2hvcjBZMBMGBYqGSM49AgEGCCqGSM49AwEHA0IABONRqhA5JAekvQN8oLwRVND
nAfBnTznLLE+SEGks677sHSeXfcVhZXUeDiN7/fsVNumaiEWRQpZh3zXPwL8rUMyJPz
A9MB0GA1UdDgQWBQBQBXEXJrLBGKnFd1xCgeMAVSfEBPzALBgNVHQ8EBAMCAoQwDwYDV
R0TAQH/BAUwAwEB/zAKBggqhkjOPQQDAgNHADBEAiALBidABsfpzG01TL9Eh9b6AUbq
nzF+koEZbgvpppvvt9QIGVoE+bhEN0j6wSPzePJLrEdD+PEgyjHJ5rbA11SPq/1M="
        },
        {
          "format": 1,
          "data":
"
ooICtjCCArIwWTATBgcqhkJOPQIBBggqhkJOPQMBBwNCAASXz21w12owQAx58euratY
WiHEkhxDU9MEgetrvAtGYZxNnkfLCsp9vLcw8ISXC8tL97k9ZCUTnr0MzLw37XKRABB
T22tH1Eou/DenPU0OzcCb3/+fibjCCAj0wUjELMAkGA1UEBgcVVMxGjAYBgNVBAoME
Vplc3R5IEhhbmRzLCBjbmuMScwJQYDVQQDDb5aZXN0eSBIYW5kcywgSW5jLiBUCnVz
dCBBbmNob3KgggHlMIIBi6ADAgECAhQL3EqgUX1QP1jyddVSRnNHvK1MzAKBggqhkJOP
QQDAjBSMQswCQYDVQQGDAJVUzEaMBGGA1UECgwRWmVzdHkgSGFuZHM5IEluYy4xJzA
lBgNVBAMMH1plc3R5IEhhbmRzLCBjbmuIFRydXN0IEFuY2hvcjAeFw0yMjA1MTkxNT
EzMdDaFw0zMjA1MTYxNTEzMdDaMFIXCzAJBgNVBAYMALVTMRowGAYDVQQKDBFzZXN0e
SBIYW5kcywgSW5jLjEnMCUGA1UEAwwWmVzdHkgSGFuZHM5IEluYy4gVHJ1c3QgQW5j
aG9yMFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAE189tcNdqMEAMefHrq2rWfohxJIC
Q1PTBIHra7JwLRmGctZ5HywrKfby3MPCElwvLS/e5PWQ1LZ69DMY8N+1ykQKM/MD0wHQ
YDVR00BBYEFpba0eUSi78N6elTQ7Nxxvf/5+JuMasGA1UdDwQEAwICHdAPBgNVHRMBA
f8EBTADAQH/MAOGCCqGSM49BAMCA0gAMEUCIB2li+f6RCxs2EnvNWciSpIDwiUViWay
Gv1A8xks80eYAiEAmCez4KGrolFKOZT6bvqf1sYQuJBfvtk/y1JQdUvoqlg="
        },
        {
          "format": 1,
          "data":
"
ooIC1TCCAtEwWTATBgcqhkJOPQIBBggqhkJOPQMBBwNCAATN0f5kzywEzZOYbaV2303
N8cku39JoLNj1HPwECbXDDWp0LpAO1z248/hoy6UW/TZMTPPR/93XwHsG16mSFy8XBB
SKhm/5gJWjvDbW7quY1peNm9cfYDCCA1wwXDELMakGA1UEBgcVVMxHzAdBgNVBAoMF
lNub2JiaXNoIEFwcGFyZW5IEluYy4xLDAqBgNVBAMMI1Nub2JiaXNoIEFwcGFyZW5
IEluYy4gVHJ1c3QgQW5jaG9yOUIB+jCCAZ+gAwIBAgIUEBuTRGXAEVEEHhu4xafAnqm
+qYqwCqYIKoZIzj0EAwIwXDELMakGA1UEBgcVVMxHzAdBgNVBAoMF1Nub2JiaXNoIE

```



```
FwcGFyZWwsIEluYy4xLDAqBgNVBAMMI1Nub2JiaXNoIEFwcGFyZWwsIEluYy4gVHJlc
3QgQW5jaG9yMB4XDTIyMDUxOTE1MTMwOFoXDTMyMDUxNjE1MTMwOFowXDELMakGA1UE
BgcwCVVMxHzAdBgNVBAoMF1Nub2JiaXNoIEFwcGFyZWwsIEluYy4xLDAqBgNVBAMMI1N
ub2JiaXNoIEFwcGFyZWwsIEluYy4gVHJlc3QgQW5jaG9yMFkwEwYHKoZIzj0CAQYIKo
ZIzj0DAQcDQgAEzdH+ZM8sBM2TmG2ldtztzfHJLt/SaCzY5Rz8BAmlww1qdC6QDtc9u
PP4aMulFv02TEzz0f/dl8B7BtephkcvF6M/MD0wHQYDVR0OBBYEFiqEz/mAlaO8Ntbu
pRjWl42b1x9gMAsGA1UdDwQEAwIChDAPBgNVHRMBAf8EBTADAQH/MAoGCCqGSM49BAM
CA0kAMEYCIQC2cf43f3PP1CO6/dxv40ftIgxToKHF72UzENv7+y4ygIhAIGtC/r6SG
aFMaP7zD2EloBuIXTtyWu8Hw1+YGdXRY93"
```

```
    }
  ]
}
}
```

The TA Store below features one TA with an environment targeting CoSWIDs with entity named "Zesty Hands, Inc," and one permitted EAT claim for software named "Bitter Paper".

```

{
  "environments": [
    {
      "swidtag": {
        "entity": [
          {
            "entity-name": "Zesty Hands, Inc.",
            "role": "softwareCreator"
          }
        ]
      }
    }
  ],
  "permclaims": [
    {
      "swname": "Bitter Paper"
    }
  ],
  "keys": {
    "tas": [
      {
        "format": 0,
        "data":
MIIB5TCCAYugAwIBAgIUC9xKoFF5UD5Y8nXVUkZzR7yvtTMwCgYIKoZIzj0EAwI
wUjELMAkGA1UEBgwCVVMxGjAYBgNVBAoMEVplc3R5IEhhbmRzLCBJbmMuMScwJQ
YDVQDDDB5aZXN0eSBIYW5kcywgSW5jLiBUcnVzdCBBbmNob3IwHhcNMjIwNTE5M
TUxMzA3WhcNMzIwNTE2MTUxMzA3WjBSMQswCQYDVQQGDAJVUzEaMBGGA1UECgwR
WmVzdHkgSGFuZHMzIEluYy4xJzA1BgNVBAMMH1plc3R5IEhhbmRzLCBJbmMuIFR
ydXN0IEFuY2hvcjBZMBMGByqGSM49AgEGCCqGSM49AwEHA0IABJfPbXDXa jBADH
nx66tqlhaIcSSHENT0wSB62u8C0ZhnE2eR8sKyn28tzDwhJcLy0v3uT1kJS2evQ
zMvDftcpECjPzA9MB0GA1UdDgQWBBT22tH1Eou/DenpU00zcCb3/+fibjALBgNV
HQ8EBAMCAoQwDwYDVR0TAQH/BAUwAwEB/zAKBggqhkjOPQQDAgNIADBFAiAdpYv
n+kQsbNhJ7zVnIkqSA8IlFYlmsHr9QPMZLPNHmAiHAJgns+Chq6JRSjmU+m76n9
bGELiQX77ZP8tSUHVL6KpY"
      }
    ]
  }
}

```

The dump below shows the COSE-Sign1-corim contents from the ASCII hex above. A full base64-encoded version of this example is given in Appendix A.

18 ([h'  
A3012603746170706C69636174696F6E2F72696D2B63626F72085841A200A20  
07441434D45204C7464207369676E696E67206B657901D8207468747470733A  
2F2F61636D652E6578616D706C6501A200C11A61CE480001C11A69546780',  
{},  
h'  
A30050EBA916FB1E3E42679214E07E1A9BF9130181590A56D901FB83A301A20  
050FB51FAC913C546C39390DC306B167F5A01050281A101A100A10173576F72  
74686C657373205365612C20496E632E06A100818202585B3059301306072A8  
648CE3D020106082A8648CE3D03010703420004AD8A0C01DA9EDA0253DC2BC2  
7227D9C7213DF8DF13E89CB9CDB7A8E4B62D9CE8A99A2D705C0F7F80DB65C00  
6D1091422B47FC611CBD46869733D9C483884D5FEA301A10071736F6D655F74  
61675F6964656E746974790281A103764D697363656C6C616E656F757320544  
12053746F726506A1008382005901C1308201BD30820164A003020102021500  
D09D90BF3D525CC773D522ED77D59E22BBA45B88300A06082A8648CE3D04030  
2303E310B300906035504060C0255533110300E060355040A0C074578616D70  
6C65311D301B06035504030C144578616D706C6520547275737420416E63686  
F72301E170D3232303531393135313330375A170D3332303531363135313330  
375A303E310B300906035504060C0255533110300E060355040A0C074578616  
D706C65311D301B06035504030C144578616D706C6520547275737420416E63  
686F723059301306072A8648CE3D020106082A8648CE3D03010703420004E35  
1AA10392407A4BD037CA0BC1154D0E701F0674F39CB2C4F9210692CEBBEEC1D  
27977DC56165751E0E237BFDFB1536E99A884591429661DF35CFC0BF2B50CCA  
33F303D301D0603551D0E04160414015C45C9ACB0462A715DD710A078C01549  
F1013F300B0603551D0F040403020284300F0603551D130101FF04053003010  
1FF300A06082A8648CE3D040302034700304402200B06274006C7E9CC6D254C  
BF4487D6FA0146EA9F317E9281196E0BE9A6FBEDF5022056813E6E110DD23EB  
048FCDE3E32EB11D0FE3C48328C7279ADB035D523EAF5382015902BAA28202  
B6308202B23059301306072A8648CE3D020106082A8648CE3D0301070342000  
497CF6D70D76A30400C79F1EBAB6AD6168871248710D4F4C1207ADAEF02D198  
67136791F2C2B29F6F2DCC3C2125C2F2D2FDEE4F59094B67AF43332F0DFB5CA  
4400414F6DAD1E5128BBF0DE9E95343B371C6F7FFE7E26E3082023D3052310B  
300906035504060C025553311A3018060355040A0C115A657374792048616E6  
4732C20496E632E3127302506035504030C1E5A657374792048616E64732C20  
496E632E20547275737420416E63686F72A08201E53082018BA003020102021  
40BDC4AA05179503E58F275D552467347BCAFB533300A06082A8648CE3D0403  
023052310B300906035504060C025553311A3018060355040A0C115A6573747  
92048616E64732C20496E632E3127302506035504030C1E5A65737479204861  
6E64732C20496E632E20547275737420416E63686F72301E170D32323035313  
93135313330375A170D3332303531363135313330375A3052310B3009060355  
04060C025553311A3018060355040A0C115A657374792048616E64732C20496  
E632E3127302506035504030C1E5A657374792048616E64732C20496E632E20  
547275737420416E63686F723059301306072A8648CE3D020106082A8648CE3  
D0301070342000497CF6D70D76A30400C79F1EBAB6AD6168871248710D4F4C1  
207ADAEF02D19867136791F2C2B29F6F2DCC3C2125C2F2D2FDEE4F59094B67A  
F43332F0DFB5CA440A33F303D301D0603551D0E04160414F6DAD1E5128BBF0D  
E9E95343B371C6F7FFE7E26E300B0603551D0F040403020284300F0603551D1  
30101FF040530030101FF300A06082A8648CE3D040302034800304502201DA5

8BE7FA442C6CD849EF3567224A9203C225158966B21AFD40F3192CF34798022  
1009827B3E0A1ABA2514A3994FA6EFA9FD6C610B8905FBED93FCB5250754BE8  
AA5882015902D9A28202D5308202D13059301306072A8648CE3D020106082A8  
648CE3D03010703420004CDD1FE64CF2C04CD93986DA576DCEDCDF1C92EDFD2  
682CD8E51CFC0409B5C30D6A742E900ED73DB8F3F868CBA516FD364C4CF3D1F  
FDDD7C07B06D7A992172F1704148A84CFF98095A3BC36D6EEA518D6978D9BD7  
1F603082025C305C310B300906035504060C025553311F301D060355040A0C1  
6536E6F6262697368204170706172656C2C20496E632E312C302A0603550403  
0C23536E6F6262697368204170706172656C2C20496E632E205472757374204  
16E63686F72A08201FA3082019FA0030201020214101B934465C01045441E1B  
B8C5A7C09EA9BEA988300A06082A8648CE3D040302305C310B3009060355040  
60C025553311F301D060355040A0C16536E6F6262697368204170706172656C  
2C20496E632E312C302A06035504030C23536E6F62626973682041707061726  
56C2C20496E632E20547275737420416E63686F72301E170D32323035313931  
35313330385A170D3332303531363135313330385A305C310B3009060355040  
60C025553311F301D060355040A0C16536E6F6262697368204170706172656C  
2C20496E632E312C302A06035504030C23536E6F62626973682041707061726  
56C2C20496E632E20547275737420416E63686F723059301306072A8648CE3D  
020106082A8648CE3D03010703420004CDD1FE64CF2C04CD93986DA576DCEDC  
DF1C92EDFD2682CD8E51CFC0409B5C30D6A742E900ED73DB8F3F868CBA516FD  
364C4CF3D1FFDDD7C07B06D7A992172F17A33F303D301D0603551D0E0416041  
48A84CFF98095A3BC36D6EEA518D6978D9BD71F60300B0603551D0F04040302  
0284300F0603551D130101FF040530030101FF300A06082A8648CE3D0403020  
349003046022100B671FE377F73CF9423BAFDDC6FE347ED220C714E828717BD  
94CC436FEFECB8CA02210081AD0BFAFA48668531A3FBCC3D8496806E2174EDC  
96BBC1F097E606757458F77A30281A102A102A2181F715A657374792048616E  
64732C20496E632E1821020481A11903E66C42697474657220506170657206A  
1008182005901E9308201E53082018BA00302010202140BDC4AA05179503E58  
F275D552467347BCAFB533300A06082A8648CE3D0403023052310B300906035  
504060C025553311A3018060355040A0C115A657374792048616E64732C2049  
6E632E3127302506035504030C1E5A657374792048616E64732C20496E632E2  
0547275737420416E63686F72301E170D3232303531393135313330375A170D  
3332303531363135313330375A3052310B300906035504060C025553311A301  
8060355040A0C115A657374792048616E64732C20496E632E31273025060355  
04030C1E5A657374792048616E64732C20496E632E20547275737420416E636  
86F723059301306072A8648CE3D020106082A8648CE3D0301070342000497CF  
6D70D76A30400C79F1EBAB6AD6168871248710D4F4C1207ADAEF02D19867136  
791F2C2B29F6F2DCC3C2125C2F2D2FDEE4F59094B67AF43332F0DFB5CA440A3  
3F303D301D0603551D0E04160414F6DAD1E5128BBF0DE9E95343B371C6F7FFE  
7E26E300B0603551D0F040403020284300F0603551D130101FF040530030101  
FF300A06082A8648CE3D040302034800304502201DA58BE7FA442C6CD849EF3  
567224A9203C225158966B21AFD40F3192CF347980221009827B3E0A1ABA251  
4A3994FA6EFA9FD6C610B8905FBED93FCB5250754BE8AA5804A200C11A61CE4  
80001C11A69546780',  
h'  
19E82D7A5C7A73B44F06305AECF0EF8CF8764286323F6D2BA27D7291F92FF5B  
0CF789F6FF88B7E2EE8EF262B4FA1DFD7D7AFB0AE2C0062C98DB332243B3E99  
94' ] )

## 6. Security Considerations

As a profile of CoRIM, the security considerations from [I-D.draft-birkholz-rats-corim] apply.

As a means of managing trust anchors, the security considerations from [RFC6024] and [RFC5934] apply. a CoTS signer is roughly analogous to a "management trust anchor" as described in [RFC5934].

## 7. IANA Considerations

### 7.1. CoRIM CBOR Tag Registration

IANA is requested to allocate tags in the "CBOR Tags" registry [IANA.cbor-tags], preferably with the specific value requested:

Tag	Data Item	Semantics
507	tagged array	Concise Trust Anchor Stores (CoTS)

Table 1

## 8. References

### 8.1. Normative References

- [I-D.draft-birkholz-rats-corim]  
Birkholz, H., Fossati, T., Deshpande, Y., Smith, N., and W. Pan, "Concise Reference Integrity Manifest", Work in Progress, Internet-Draft, draft-birkholz-rats-corim-03, 11 July 2022, <<https://datatracker.ietf.org/doc/html/draft-birkholz-rats-corim-03>>.
- [I-D.draft-ietf-rats-eat]  
Lundblade, L., Mandyam, G., O'Donoghue, J., and C. Wallace, "The Entity Attestation Token (EAT)", Work in Progress, Internet-Draft, draft-ietf-rats-eat-16, 9 October 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-rats-eat-16>>.
- [I-D.draft-ietf-sacm-coswid]  
Birkholz, H., Fitzgerald-McKay, J., Schmidt, C., and D. Waltermire, "Concise Software Identification Tags", Work in Progress, Internet-Draft, draft-ietf-sacm-coswid-22, 20 July 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-sacm-coswid-22>>.

- [IANA.cbor-tags]  
IANA, "Concise Binary Object Representation (CBOR) Tags",  
19 September 2013,  
<<https://www.iana.org/assignments/cbor-tags>>.
- [IANA.language-subtag-registry]  
IANA, "Language Subtag Registry",  
<<https://www.iana.org/assignments/language-subtag-registry>>.
- [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>>.
- [RFC5914] Housley, R., Ashmore, S., and C. Wallace, "Trust Anchor Format", RFC 5914, DOI 10.17487/RFC5914, June 2010, <<https://www.rfc-editor.org/rfc/rfc5914>>.
- [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>>.
- [RFC8949] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", STD 94, RFC 8949, DOI 10.17487/RFC8949, December 2020, <<https://www.rfc-editor.org/rfc/rfc8949>>.

## 8.2. Informative References

- [dloa] GlobalPlatform, "GlobalPlatform Card - Digital Letter of Approval Version 1.0", November 2015, <[https://globalplatform.org/wp-content/uploads/2015/12/GPC\\_DigitalLetterOfApproval\\_v1.0.pdf](https://globalplatform.org/wp-content/uploads/2015/12/GPC_DigitalLetterOfApproval_v1.0.pdf)>.
- [fido-metadata] FIDO Alliance, "FIDO Metadata Statement", May 2021, <<https://fidoalliance.org/specs/mds/fido-metadata-statement-v3.0-ps-20210518.html>>.

- [fido-service]  
FIDO Alliance, "FIDO Metadata Service", May 2021,  
<<https://fidoalliance.org/specs/mds/fido-metadata-service-v3.0-ps-20210518.html>>.
- [I-D.draft-ietf-rats-architecture]  
Birkholz, H., Thaler, D., Richardson, M., Smith, N., and  
W. Pan, "Remote Attestation Procedures Architecture", Work  
in Progress, Internet-Draft, draft-ietf-rats-architecture-  
22, 28 September 2022,  
<<https://datatracker.ietf.org/doc/html/draft-ietf-rats-architecture-22>>.
- [RFC3779] Lynn, C., Kent, S., and K. Seo, "X.509 Extensions for IP  
Addresses and AS Identifiers", RFC 3779,  
DOI 10.17487/RFC3779, June 2004,  
<<https://www.rfc-editor.org/rfc/rfc3779>>.
- [RFC5934] Housley, R., Ashmore, S., and C. Wallace, "Trust Anchor  
Management Protocol (TAMP)", RFC 5934,  
DOI 10.17487/RFC5934, August 2010,  
<<https://www.rfc-editor.org/rfc/rfc5934>>.
- [RFC6024] Reddy, R. and C. Wallace, "Trust Anchor Management  
Requirements", RFC 6024, DOI 10.17487/RFC6024, October  
2010, <<https://www.rfc-editor.org/rfc/rfc6024>>.
- [RFC8152] Schaad, J., "CBOR Object Signing and Encryption (COSE)",  
RFC 8152, DOI 10.17487/RFC8152, July 2017,  
<<https://www.rfc-editor.org/rfc/rfc8152>>.

## Acknowledgments

TODO acknowledge.

## Appendix A

The base64 encoded data below represents a signed CoRIM that features  
a concise-ta-stores containing the three examples shown above.

```
0oRYXaMBJgN0YXBwbGljYXRpb24vcmltK2Nib3IIWEGiAKIAdEFDTUUgTHRkIHN  
pZ25pbmcga2V5AdggdGh0dHBzOi8vYWNtZS5leGFtcGxlAaIAwRphzkGAAcEaaV  
RngKBZCn6jAFDrqRb7Hj5CZ5IU4H4am/kTAYFZC1bZAfuDowGiAFD7UfrJE8VGw  
5OQ3DBrFn9aAQUcGaEBoQChAXNXb3J0aGxl3MgU2VhLCBJbmMuBqEAgYICWFsw  
WTATBgqcqhkjOPQIBBgqhkjOPQMBBwNCAASTigwB2p7aAlPcK8JyJ9nHIT343xP  
onLnNt6jkti2c6KmaLXBcd3+A22XABtEJFCK0f8YRy9RoaxM9nEg4hNX+owGhAH  
Fzb21lX3RhZl9pZGVudG10eQKBoQN2TW1zY2VsbGFuZW91cyBUQSBTdG9yZQahA  
IOCAFkBWTCcAB0wggFkoAMCAQICFQDQnZC/PVJcx3PVIu13lZ4iu6RbiDAKBggq
```

hk jOPQQDAjA+MQswCQYDVQQGDAJVUzEQMA4GA1UECgwHRXhhbXBsZTEdMBsGA1UEAwURXhhbXBsZSBUCnVzdCBBbmNob3IwHhcNMjIwNTE5MTUxMzA3WhcNMzIwNTE2MTUxMzA3WjA+MQswCQYDVQQGDAJVUzEQMA4GA1UECgwHRXhhbXBsZTEdMBsGA1UEAwURXhhbXBsZSBUCnVzdCBBbmNob3IwWTATBgcqhk jOPQIBBggqhk jOPQMBBwNCAATjUaoQOSQHPLODfKC8EVTQ5wHwZ085yyxPkhBpLOu+7B0n133FYWV1Hg4je/37FTbpmohFkUKWYd81z8C/K1DMoz8wPTAdBgNVHQ4EFgQUAVxYaywRipxXd cQoHjAFUnxAT8wCwYDVR0PBAQDAgKEMA8GA1UdEwEB/wQFMAMBAf8wCgYIKoZIZ j0EAwIDRWAwRAIGCwYnQAbH6cxtJUy/RIfW+gFG6p8xfpKBGW4L6ab77fUCIFaB Pm4RDdI+sEj83j4y6xHQ/jxIMoxyea2wNdUj6v9TggFZArqiggK2MIICs jBZMBM GBYqGSM49AgEGCCqGSM49AwEHA0IABJfPbXDXa jBADHnx66tqlhaIcSSHENT0wS B62u8C0ZhnE2eR8sKyn28tzDwhJcLy0v3uT1kJS2evQzMvDftcpEAEFFba0eUSi 78N6e1TQ7Nxxvf/5+JuMIICPTBSMQswCQYDVQQGDAJVUzEaMBGGA1UECgwRWmVz dHkgSGFuZHM sIEluYy4xJzA1BgNVBAMMH1plc3R5IEhhbmRzLCBjbmuIFRydXN0IEFuY2hvcqCCAeUwggGLOAMCAQICFAvcSqBREVA+WPJ11VJGc0e8r7UzMAoGCC qGSM49BAMCMFIxCzAJBgNVBAYMA1VTMR0wGAYDVQQKDBFazXN0eSBIYW5kcywgS W5jLjEnMCUGA1UEAwweWmVzdHkgSGFuZHM sIEluYy4gVHJ1c3QgQW5jaG9yMB4X DTIyMDUxOTE1MTMwN1oXDTMyMDUxNjE1MTMwN1owUjELMAKGA1UEBgcVVMxGjA YBgNVBAoMEVplc3R5IEhhbmRzLCBjbmuMScwJQYDVQQDDDB5aXN0eSBIYW5kcy wgSW5jLiBUcnVzdCBBbmNob3IwWTATBgcqhk jOPQIBBggqhk jOPQMBBwNCAASXz 21w12owQAx58euratYWiHEkhxDU9MEgetrvAtGYZxNnkfLCsp9vLcw8ISXC8tL9 7k9ZCUtnr0MzLw37XKRAoz8wPTAdBgNVHQ4EFgQU9trR5RKLv3p6VNDs3HG9// n4m4wCwYDVR0PBAQDAgKEMA8GA1UdEwEB/wQFMAMBAf8wCgYIKoZIZ j0EAwIDSA AwRQIGHaWL5/pELGzYSe81ZyJKkgPCJRWJZrIa/UDzGSzZr5gCIQCYJ7PgoaiiU Uo5lPpu+p/WxhC4kF++2T/LU1B1s+iqWIIIBWQLZooIC1TCCAtEwWTATBgcqhk jO PQIBBggqhk jOPQMBBwNCAATN0f5kzywEzZOYbaV2303N8cku39JoLNj1HPwECbX DDWpOLpAO1z248/hoy6UW/TZMTPPR/93XwHsG16mSFy8XBBSKhM/5gJWjvDbW7q UY1peNm9cfYDCCA1wwXDELMAKGA1UEBgcVVMxHzaDgNVBAoMF1Nub2JiaXNoIE FwGfYzWwsIEluYy4xLDAqBgNVBAMMI1Nub2JiaXNoIEFwGfYzWwsIEluYy4g VHJ1c3QgQW5jaG9yOIIIB+jCCAZ+gAwIBAgIUEBuTRGXAEVEHhu4xafAnqm+qYg wCgYIKoZIZ j0EAwIwXDELMAKGA1UEBgcVVMxHzaDgNVBAoMF1Nub2JiaXNoIE FwGfYzWwsIEluYy4xLDAqBgNVBAMMI1Nub2JiaXNoIEFwGfYzWwsIEluYy4gV HJ1c3QgQW5jaG9yMB4XD TIyMDUxOTE1MTMwOFoXDTMyMDUxNjE1MTMwOFowXDEL MAKGA1UEBgcVVMxHzaDgNVBAoMF1Nub2JiaXNoIEFwGfYzWwsIEluYy4xLDA qBgNVBAMMI1Nub2JiaXNoIEFwGfYzWwsIEluYy4gVHJ1c3QgQW5jaG9yMFkwEw YHkoZIZ j0CAQYIKoZIZ j0DAQcDQgAEzdH+ZM8sBM2TmG2ldtztzfHJLt/SaCzY5 Rz8BAm1ww1qdC6QDtc9uPP4aMulFv02Tezz0f/dl8B7Btepkhcvf6M/MD0wHQYD VR0OBBYEFiqEz/mAlaO8NtbupRjW142b1x9gMAsGA1UdDwQEAWICHDAPBgNVHRM BAf8EBTADAQH/MAoGCCqGSM49BAMCA0kAMEYCIQC2cf43f3PP1CO6/dxv40ftIg xxToKHF72UzENv7+y4ygIhAIGtC/r6SGaFMaP7zD2EloBuIXTtyWu8Hw1+YGdXR Y93owKBoQKhAqIYH3FaZNX0eSBIYW5kcywgSW5jLhghAgSBoRkD5mxCaXR0ZXIq UGFwZXIGoQCBggBZAekwggH1MIIBi6ADAgECAhQL3EggUX1QP1jyddVSRnNHvK+ 1MzAKBggqhk jOPQQDAjBSMQswCQYDVQQGDAJVUzEaMBGGA1UECgwRWmVzdHkgSG FuZHM sIEluYy4xJzA1BgNVBAMMH1plc3R5IEhhbmRzLCBjbmuIFRydXN0IEFuY2hvc jAeFw0yMjA1MTkxNTEzMdDaFw0zMjA1MTYxNTEzMdDaMF1xCzAJBgNVBAYM A1VTMR0wGAYDVQQKDBFazXN0eSBIYW5kcywgSW5jLjEnMCUGA1UEAwweWmVzdHk gSGFuZHM sIEluYy4gVHJ1c3QgQW5jaG9yMFkwEwYHkoZIZ j0CAQYIKoZIZ j0DAQ cDQgAE189tcNdqMEAMEfHrq2rWFohxJICQ1PTBIHra7wLRmGcTZ5HywrKfby3MP CELwvLS/e5PWQ1LZ69DMY8N+1ykQKM/MD0wHQYDVR0OBBYEFpba0eUSi78N6e1T



Q7Nxxvf/5+JuMAsGA1UdDwQEAwIChDAPBgNVHRMBAf8EBTADAQH/MAoGCCqGSM4  
9BAMCA0gAMEUCIB2li+f6RCxs2EnvNWciSpIDwiUViWayGv1A8xks80eYAiEAmC  
ez4KGrolFKOZT6bvqflsYQuJBfvtk/y1JQdUvoqlgEogDBGmHOSAABwRppVGeAW  
EAZ6C16XHpzte8GMFrs8O+M+HZChjI/bSui fXKR+S/1sM94n2/4i34u6O8mK0+h  
39fXr7CuLABiyY2zMtQ7PpmU

## Authors' Addresses

Carl Wallace  
Red Hound Software  
United States of America  
Email: carl@redhoundsoftware.com

Russ Housley  
Vigil Security, LLC  
516 Dranesville Road  
Herndon, VA 20170  
United States of America  
Email: housley@vigilsec.com

Thomas Fossati  
arm  
Email: Thomas.Fossati@arm.com

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