SIDROPS
Internet-Draft
Intended status: Standards Track
Expires: 19 March 2026

J. Snijders
B. Bakker
T. Bruijnzeels
RIPE NCC

15 September 2025

A Profile for Resource Public Key Infrastructure (RPKI) Canonical Cache Representation (CCR)
draft-spaghetti-sidrops-rpki-ccr-03

Abstract

This document specifies a Canonical Cache Representation (CCR) content type for use with the Resource Public Key Infrastructure (RPKI). CCR is a DER-encoded data interchange format which can be used to represent various aspects of the state of a validated cache at a particular point in time. The CCR profile is a compact and versatile format well-suited for a diverse set of applications such as audit trail keeping, validated payload dissemination, and analytics pipelines.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 19 March 2026.

Copyright Notice

Table of Contents

1. Introduction

This document specifies a Canonical Cache Representation (CCR) content type for use with the Resource Public Key Infrastructure (RPKI). A validated cache contains all RPKI objects that the Relying Party (RP) has verified to be valid according to the rules for validation (see [RFC6487], [RFC6488], [RFC9286]). CCR is a data interchange format using Distinguished Encoding Rules (DER, [X.690])

which can be used to represent various aspects of the state of a validated cache at a particular point in time. The CCR profile is a compact and versatile format well-suited for a diverse set of applications such as audit record keeping, validated payload dissemination, and analytics pipelines.

The format was primarily designed to support comparative analysis of uniformities and differences among multiple RP instances using different RPKI transport protocols (such as [RFC5781], [RFC8182], and [I-D.spaghetti-sidrops-rpki-erik-protocol]).

1.1. Requirements Language

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.

2. The Canonical Cache Representation content type

The contentType for a CCR currently is defined as id-ct-rpkiCanonicalCacheRepresentation, with temporary Object Identifier (OID) 1.3.6.1.4.1.41948.825.

Note: as part of the standardization process, at a future point in time, the aforementioned contentType value will change from the current Private Enterprise Number ([RFC9371]) to an OID assigned by IANA (Section 6).

3. The Canonical Cache Representation content

The content of a Canonical Cache Representation is formally defined as follows:

```
RpkiCanonicalCacheRepresentation-2025
  { iso(1) member-body(2) us(840) rsadsi(113549)
    pkcs(1) pkcs9(9) smime(16) mod(0) id-mod-rpkiCCR-2025(TBD) }

DEFINITIONS EXPLICIT TAGS ::=
BEGIN

IMPORTS
  CONTENT-TYPE, Digest, DigestAlgorithmIdentifier, SubjectKeyIdentifier
  FROM CryptographicMessageSyntax-2010 -- in [RFC6268]
    { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1)
      pkcs-9(9) smime(16) modules(0) id-mod-cms-2009(58) }
```

```
    -- in [draft-spaghetti-sidrops-rpki-erik-protocol-02]
    ManifestRef
    FROM RpkiErikPartition-2025
      { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1)
        pkcs9(9) smime(16) mod(0) id-mod-rpkiErikPartition-2025(TBD) }

    ASID, ROAIPAddressFamily
    FROM RPKI-ROA-2023 -- in [RFC9582]
      { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1)
        pkcs9(9) smime(16) mod(0) id-mod-rpkiROA-2023(75) }

    SubjectPublicKeyInfo
    FROM PKIX1Explicit-2009
      { iso(1) identified-organization(3) dod(6) internet(1) security(5)
        mechanisms(5) pkix(7) id-mod(0) id-mod-pkix1-explicit-02(51) }
    ;

  ct-rpkiCanonicalCacheRepresentation CONTENT-TYPE ::=
    { TYPE RpkiCanonicalCacheRepresentation
      IDENTIFIED BY id-ct-rpkiCanonicalCacheRepresentation }

  id-ct-rpkiCanonicalCacheRepresentation OBJECT IDENTIFIER ::=
    { iso(1) identified-organization(3) dod(6) internet(1) private(4)
      enterprise(1) snijders(41948) ccr(825) }

  RpkiCanonicalCacheRepresentation ::= SEQUENCE {
    version     [0] INTEGER DEFAULT 0,
    hashAlg         DigestAlgorithmIdentifier,
    producedAt      GeneralizedTime,
    mfts        [1] ManifestState OPTIONAL,
    vrps        [2] ROAPayloadState OPTIONAL,
    vaps        [3] ASPAPayloadState OPTIONAL,
    tas         [4] TrustAnchorState OPTIONAL,
    rks         [5] RouterKeyState OPTIONAL,
    ... }
    -- at least one of mfts, vrps, vaps, or tas MUST be present
    ( WITH COMPONENTS { ..., mfts PRESENT } |
      WITH COMPONENTS { ..., vrps PRESENT } |
      WITH COMPONENTS { ..., vaps PRESENT } |
      WITH COMPONENTS { ..., tas PRESENT } |
      WITH COMPONENTS { ..., rks PRESENT } )

  ManifestState ::= SEQUENCE {
    mftrefs           SEQUENCE OF ManifestRef,
    mostRecentUpdate  GeneralizedTime,
    hash              Digest }

  ROAPayloadState ::= SEQUENCE {
```

```
    rps               SEQUENCE OF ROAPayloadSet,
    hash              Digest }

  ROAPayloadSet ::= SEQUENCE {
    asID              ASID,
    ipAddrBlocks      SEQUENCE (SIZE(1..2)) OF ROAIPAddressFamily }

  ASPAPayloadState ::= SEQUENCE {
    aps               SEQUENCE OF ASPAPayloadSet,
    hash              Digest }

  ASPAPayloadSet ::= SEQUENCE {
    customerASID      ASID,
    providers         SEQUENCE (SIZE(1..MAX)) OF ASID }

  TrustAnchorState ::= SEQUENCE {
    skis              SEQUENCE (SIZE(1..MAX)) OF SubjectKeyIdentifier,
    hash              Digest }

  RouterKeyState ::= SEQUENCE {
    rksets            SEQUENCE OF RouterKeySet,
    hash              Digest }

  RouterKeySet ::= SEQUENCE {
    asID              ASID,
    routerKeys        SEQUENCE (SIZE(1..MAX)) OF RouterKey }

  RouterKey ::= SEQUENCE {
    ski               SubjectKeyIdentifier,
    spki              SubjectPublicKeyInfo }

  END
```

3.1. version

The version field contains the format version for the RpkiCanonicalCacheRepresentation structure, in this version of the specification it MUST be 0.

3.2. hashAlg

The hashAlg field specifies the algorithm used to construct the message digests. This profile uses SHA-256 [SHS], therefore the OID MUST be 2.16.840.1.101.3.4.2.1.

3.3. producedAt

The producedAt field contains a GeneralizedTime and indicates the moment in time the CCR was generated.

3.4. State aspect fields

Each CCR contains one or more fields representing particular aspects of the cache's state. Implementers should note the ellipsis extension marker in the RpkiCanonicalCacheRepresentation ASN.1 notation and anticipate future changes as new signed object types are standardized.

Each state aspect generally consists of a sequence of details extracted from RPKI Objects of a specific type, along with a digest computed by hashing the aforementioned DER-encoded sequence, optionally including some metadata.

3.4.1. ManifestState

An instance of ManifestState represents the set of valid, current Manifests ([RFC9286]) in the cache. It contains three fields:

The mftrefs field contains a SEQUENCE of ManifestRef structures (see Section 3 of [I-D.spaghetti-sidrops-rpki-erik-protocol]) sorted in ascending order by hash value. The hash value in each instance of ManifestRef MUST be unique with respect to the other instances of ManifestRef.

The mostRecentUpdate is a metadata field which contains the most recent thisUpdate amongst all Manifests. If the mftrefs field contains an empty sequence, the mostRecentUpdate MUST be set to the POSIX Epoch ("19700101000000Z").

The hash field contains a message digest computed using the mftrefs value (encoded in DER format) as input message.

3.4.2. ROAPayloadState

An instance of ROAPayloadState contains a field named rps which represents the current set of Validated ROA Payloads (Section 2 of [RFC6811]) encoded as a SEQUENCE of ROAPayloadSet instances.

The ROAPayloadSet structure is modeled after the RouteOriginAttestation (Section 4 of [RFC9582]). The asID value in each instance of ROAPayloadSet MUST be unique with respect to other instances of ROAPayloadSet. The contents of the ipAddrBlocks field MUST appear in canonical form and ordered as defined in Section 4.3.3 of [RFC9582].

The hash field contains a message digest computed using the rps value (encoded in DER format) as input message.

3.4.3. ASPAPayloadState

An instance of ASPAPayloadState contains an aps field which represents the current set of deduplicated and merged ASPA payloads ([I-D.ietf-sidrops-aspa-profile]) ordered by ascending customerASID value encoded as a SEQUENCE of ASPAPayloadSet instances. The customerASID value in each instance of ASPAPayloadSet MUST be unique with respect to other instances of ASPAPayloadSet.

The ASPAPayloadSet structure is modeled after the ProviderASSet (Section 3.3 of [I-D.ietf-sidrops-aspa-profile]).

The hash field contains a message digest computed using the aps value (encoded in DER format) as input message.

3.4.4. TrustAnchorState

An instance of TrustAnchorState represents the set of valid Trust Anchor (TA) Certification Authority (CA) resource certificates used by the relying party when producing the CCR.

Each SubjectKeyIdentifier is the 160-bit SHA-1 hash of the value of the DER-encoded ASN.1 bit string of the TA’s Subject Public Key, as described in Section 4.8.2 of [RFC6487]. The skis field contains a sequence of Subject Key Identifiers (SKI) sorted in ascending order by interpreting the SKI value as an unsigned 160-bit integer.

The hash field contains a message digest computed using the skis value (encoded in DER format) as input message.

3.4.5. RouterKeyState

An instance of RouterKeyState contains an rksets field which represents the current set of valid BGPsec Router Keys [RFC8205] encoded as a SEQUENCE of RouterKeySet instances. The asID value in each instance of RouterKeySet MUST be unique with respect to other instances of RouterKeySet. Instances of RouterKeySet are sorted by ascending value of asID. Instances of RouterKey are sorted by

ascending value of ski by interpreting the SKI value as an unsigned 160-bit integer.

The hash field contains a message digest computed using the rks value (encoded in DER format) as input message.

4. Operational Considerations

Comparing the ManifestState mostRecentUpdate timestamp value with the producedAt timestamp might help offer insight into the timing and propagation delays of the RPKI supply chain.

Given the absence of public keys and fairly repetitive content in RPKI AccessDescription instances, it should be noted CCR content compresses well.

4.1. Verifying CCR file integrity

The integrity of a CCR object can be checked by confirming whether the hash values embedded inside state aspects match the computed hash value of the respective state aspect payload structure.

5. Security Considerations

CCR objects are not signed objects.

6. IANA Considerations

6.1. SMI Security for S/MIME CMS Content Type (1.2.840.113549.1.9.16.1)

IANA is requested to allocate the following in the "SMI Security for S/MIME CMS Content Type (1.2.840.113549.1.9.16.1)" registry:

Decimal	Description	References
TBD	id-ct-rpkiCanonicalCacheRepresentation	draft-spaghetti-sidrops-rpki-ccr

Table 1

6.2. RPKI Repository Name Schemes

IANA is requested to add the Canonical Cache Representation file extension to the "RPKI Repository Name Schemes" registry [RFC6481] as follows:

Filename Extension	RPKI Object	Reference
.ccr	Canonical Cache Representation	draft-spaghetti-sidrops-rpki-ccr

Table 2

6.3. SMI Security for S/MIME Module Identifier (1.2.840.113549.1.9.16.0)

IANA is requested to allocate the following in the "SMI Security for S/MIME Module Identifier (1.2.840.113549.1.9.16.0)" registry:

Decimal	Description	References
TBD	id-mod-rpkiCCR-2025	draft-spaghetti-sidrops-rpki-ccr

Table 3

6.4. Media Types

IANA is requested to register the media type "application/rpki-ccr" in the "Media Types" registry as follows:

6.4.1. Canonical Cache Representation Media Type

Type name: application
Subtype name: rpki-ccr
Required parameters: N/A
Optional parameters: N/A
Encoding considerations: binary
Security considerations: This media type contains no active content.
Interoperability considerations: N/A
Published specification: draft-spaghetti-sidrops-rpki-ccr
Applications that use this media type: RPKI operators
Fragment identifier considerations: N/A
Additional information:
Content: This media type is a RPKI Canonical Cache Representation object, as defined in draft-spaghetti-sidrops-rpki-ccr.
Magic number(s): N/A
File extension(s): .ccr
Macintosh file type code(s): N/A

Person & email address to contact for further information: Job Snijders (job@sobornost.net)
Intended usage: COMMON
Restrictions on usage: N/A
Author: Job Snijders (job@sobornost.net)
Change controller: IETF

7. References

7.1. Normative References

[I-D.ietf-sidrops-aspa-profile]
Azimov, A., Uskov, E., Bush, R., Snijders, J., Housley, R., and B. Maddison, "A Profile for Autonomous System Provider Authorization", Work in Progress, Internet-Draft, draft-ietf-sidrops-aspa-profile-20, 18 August 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-sidrops-aspa-profile-20>.

[I-D.spaghetti-sidrops-rpki-erik-protocol]
Snijders, J., Bruijnzeels, T., Harrison, T., and W. Ohgai, "The Erik Synchronization Protocol for use with the Resource Public Key Infrastructure (RPKI)", Work in Progress, Internet-Draft, draft-spaghetti-sidrops-rpki-erik-protocol-01, 11 September 2025, <https://datatracker.ietf.org/doc/html/draft-spaghetti-sidrops-rpki-erik-protocol-01>.

[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/info/rfc2119>.

[RFC6481] Huston, G., Loomans, R., and G. Michaelson, "A Profile for Resource Certificate Repository Structure", RFC 6481, DOI 10.17487/RFC6481, February 2012, <https://www.rfc-editor.org/info/rfc6481>.

[RFC6487] Huston, G., Michaelson, G., and R. Loomans, "A Profile for X.509 PKIX Resource Certificates", RFC 6487, DOI 10.17487/RFC6487, February 2012, <https://www.rfc-editor.org/info/rfc6487>.

[RFC6488] Lepinski, M., Chi, A., and S. Kent, "Signed Object Template for the Resource Public Key Infrastructure (RPKI)", RFC 6488, DOI 10.17487/RFC6488, February 2012, <https://www.rfc-editor.org/info/rfc6488>.

[RFC6811] Mohapatra, P., Scudder, J., Ward, D., Bush, R., and R. Austein, "BGP Prefix Origin Validation", RFC 6811, DOI 10.17487/RFC6811, January 2013, <https://www.rfc-editor.org/info/rfc6811>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC9286] Austein, R., Huston, G., Kent, S., and M. Lepinski, "Manifests for the Resource Public Key Infrastructure (RPKI)", RFC 9286, DOI 10.17487/RFC9286, June 2022, <https://www.rfc-editor.org/info/rfc9286>.

[RFC9582] Snijders, J., Maddison, B., Lepinski, M., Kong, D., and S. Kent, "A Profile for Route Origin Authorizations (ROAs)", RFC 9582, DOI 10.17487/RFC9582, May 2024, <https://www.rfc-editor.org/info/rfc9582>.

[SHS] National Institute of Standards and Technology, "Secure Hash Standard", March 2012, <https://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf>.

[X.690] ITU-T, "Information technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (B ER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)", ITU-T Recommendation X.690, ISO/IEC 8825-1:2021, February 2021, <https://www.itu.int/rec/T-REC-X.690-202102-I/en>.

7.2. Informative References

[RFC5781] Weiler, S., Ward, D., and R. Housley, "The rsync URI Scheme", RFC 5781, DOI 10.17487/RFC5781, February 2010, <https://www.rfc-editor.org/info/rfc5781>.

[RFC8182] Bruijnzeels, T., Muravskiy, O., Weber, B., and R. Austein, "The RPKI Repository Delta Protocol (RRDP)", RFC 8182, DOI 10.17487/RFC8182, July 2017, <https://www.rfc-editor.org/info/rfc8182>.

[RFC8205] Lepinski, M., Ed. and K. Sriram, Ed., "BGPsec Protocol Specification", RFC 8205, DOI 10.17487/RFC8205, September 2017, <https://www.rfc-editor.org/info/rfc8205>.

[RFC9371] Baber, A. and P. Hoffman, "Registration Procedures for Private Enterprise Numbers (PENs)", RFC 9371, DOI 10.17487/RFC9371, March 2023, <https://www.rfc-editor.org/info/rfc9371>.

[rpki-client] Jeker, C., Dzonsons, K., Buehler, T., and J. Snijders, "rpki-client", September 2025, <https://www.rpki-client.org/>.

Appendix A. Acknowledgements

The authors wish to thank Theo Buehler and Russ Housley for their generous feedback on this specification,

Appendix B. Example CCR

The below is a Base64-encoded example CCR object. For a more elaborate example based on the global RPKI, see the URL in Appendix C.

MIII/wYKKwYBBAGCx1yGOaCCCO8EggjrMIII5wYJYIZIAWUDBAIBGA8yMDI1MDkxNTEyMjg0
MVqhggVHMIIFQzCCBQwwgdEEIAAAiSH0phBIZFlYrQgsHFJqSJ+ax/asiFndLzYVUuLBAgIH
zgQUqgvXjKLKpeuP87noI3dNwbdN6+sCAgVQGA8yMDI1MDkxNTA5MDE1MlowfjB8BggrBgEF
BQcwC4ZwcnN5bmM6Ly9ycGtpLnJpcGUubmV0L3JlcG9zaXRvcnkvREVGQVVMVC9jMy85Nzg4
MzctMTAyMy00YjM5LTliMzctYTEwOGRiZTBmZDFmLzEvcWd2WGpLTEtwZXVQODdub0kzZE53
YmRONi1zLm1mdDCB0QQgAAIINi/FNk2gG24RmitkMFRwPv2mr4OuSOeTnChnM2ECAgeEBBRM
cenyPurKaMchy+QoBhALdeFXcAICBUQYDzIwMjUwOTE1MDQwMTM3WjB+MHwGCCsGAQUFBzAL
hnByc3luYzovL3Jwa2kucmlwZS5uZXQvcmVwb3NpdG9yeS9ERUZBVUxULzE2Lzk0ZjY2Mi01
MDZiLTRlYzktYTdhYS1jOWI3ZWIyY2VhNTcvMS9USEhwOGo3cXltakhJY3ZrS0FZUUMzWGhW
M0EubWZ0MIHUBCAAAmznsh/USQWDhOQsuF0UsFIu8uvGjQVEwz8XHXyuSAICCNwEFI6N1bLB
AbmDp5nJOKoOQKZR/L4HAgIBLxgPMjAyNTA5MTUwNTAxMjlaMIGAMH4GCCsGAQUFBzALhnJy
c3luYzovL3Jwa2kuYXBuaWMubmV0L21lbWJlcl9yZXBvc2l0b3J5L0E5MTU2MzM0L0U3Q0RE
RkNFQzhEODExRUU5RjRBQjk2REM0RjlBRTAyL2pvM1Zzc0VCdVlPbm1jazRxZzVBcGxIOHZn
Yy5tZnQwgeIEIAACk3htPo58kvOxNTrL+VTUvxdYjfPp0tNhnWsqwfBuAgIIwAQUV419yXeb
J4Feu1w/X88HzFg3CLMCAgEhGA8yMDI1MDkxNTAwMjA1NVowgY4wgYsGCCsGAQUFBzALhn9y
c3luYzovL3JzeW5jLnBhYXMucnBraS5yaXBlLm5ldC9yZXBvc2l0b3J5L2VjYTI1MzUwLTll
M2YtNDZlMS1hZTJiLTFjNjFiNmVhNWFlMi8wLzU3OEQ3REM5Nzc5QjI3ODE1RUJCNUMzRjVG
Q0YwN0NDNTgzNzA4QjMubWZ0MIHRBCAABdMuNuWiXRU47KUgaQAp1Qeej1PCy2yUN+SDzXws
VAICDZEEFBufIboE5eMHezkePUj/WwpQszafAgIKvxgPMjAyNTA5MTUxMTAxMDNaMH4wfAYI
KwYBBQUHMAuGcHJzeW5jOi8vcnBraS5yaXBlLm5ldC9yZXBvc2l0b3J5L0RFRkFVTFQvMjIv
ZjY3YTc4LTE4M2QtNDk0YS05OTVlLWY1MWM1ZmIzZGY5YS8xL0c1OGh1Z1RsNHdkN09SNDlT
UDliQ2xDek5wOC5tZnQwgdEEIAAHsnERrXUhPz3IFrxam/4W+HF+d3GAlfFatYjlLPRfAgIH
hAQUOe9KpZd/HzcTPnMQgJ//iTXAyLwCAgPeGA8yMDI1MDkxNTEyMDE0NFowfjB8BggrBgEF
BQcwC4ZwcnN5bmM6Ly9ycGtpLnJpcGUubmV0L3JlcG9zaXRvcnkvREVGQVVMVC9mZS8xNDhm
NzEtYWIyYy00NDA5LWFlMWYtZTdkNWZjZTZiZmFjLzEvT2U5S3BaZF9IemNUUG5NUWdKX19p
VFhBeUx3Lm1mdBgPMjAyNTA5MTUxMjAxNDRaBCARm9fasElYJ3tNOED50oo4u6Y8pJYv3yWy
289+U6Pph6KCAXIwggFuMIIBSDBkAgEHMF8wSAQCAAEwQjAJAwQAwCNeAgEgMAkDBADAQysC
ASAwCQMEAMIgRQIBIDAJAwQBwiDaAgEgMAkDBADCIooCASAwCQMEAcI9XAIBIDATBAIAAjAN
MAsDBQMqCztAAgIAgDCBowICIFswgZwwfgQCAAEweDAGAwQAW9AiMAYDBABejvAwBgMEA16O
8DAGAwQAXo7xMAYDBABejvIwBgMEAF6O9DAGAwQAXo71MAYDBABejvYwBgMEAF6O9zAGAwQA
uTTgMAYDBAK5NOAwBgMEALk04TAGAwQAuTTiMAYDBAC5NOMwBgMEAMs4LDAaBAIAAjAUMAkD
BwAgAQZ4BogwBwMFACoCCJgwOgICPMowNDAyBAIAAjAsMAkDBwAgAQZ8IIwwCQMHACABBygY
CDAJAwcAKg6yQAAAMAkDBwAqDrJAARgEIAJfihdmpfxPBloevnOquIqGrimX4wGRErhD32bn
VO0Mo4GWMIGTMG8wDgICA7EwCAICBY0CAjOdMBcCAh4nMBECAgOxAgIFjQICM50CAwDu0jAP
AgJIrzAJAgJP+QIDAyp/MAoCAwCYWjADAgEAMCcCAwCi+DAgAgIArgICBRMCAiMqAgMAxr0C
AwDqpAIDAxq2AgMDPhwEIBeP7Zc4aZi+GCoXMcr/DKHazjIGckNbl4z+LxHLcE7OpFIwUDAs
BBQLnMqQ3Q16ijdmaxkhf+DYQDe3ogQU6FUrH9bRpPfkBMbY5WgNHrwWP8MEIP+wmtCcK45g
tkciIlpky13WtimBp00mgS+Yvpr0vi+HpYIBGTCCARUwgfAwge0CAjzKMIHmMHEEFF1CUOLY
HURI2KKe/OkdKf8HXsniMFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEgFcjQ/g//LAQerAH
2Mpp+GucoDAGBbhIqD33wNPsXxnAGb+mtZ7XQrVO9DQ6UlAShtig5+QfEKpTtFgiqfiAFTBx
BBS+iJtV0Lc3OX11xJ9IW4WPqYrRHzBZMBMGByqGSM49AgEGCCqGSM49AwEHA0IABOBcSa9J
9m7sdbl9RL5fkFsGWLyGnT4y7hV9psairgBlISp6+1Syw4KxPvpfaeXh9pFkzVQDdthVFN3W
/0SqRNsEILpftEnO+2ugDzYSeWKi7qboZ/6FErvdrenG5Li8FsHS

It decodes as follows:

```
   File:                      /var/db/rpki-client/rpki.ccr
   Hash identifier:           k5quWMYc3o8nIzijN+dNEWNBNbHmjjtGfFvzdhT8MsM=
   CCR produced at:           Mon 15 Sep 2025 12:28:41 +0000
   Manifest state hash:       MTE5QkQ3REFCMDQ5NTgyNzdCNEQzODQwRjlEMjhBMzg=
   Manifest last update:      Mon 15 Sep 2025 12:01:44 +0000
   Manifest references:
                              hash:AACJIfSmEEhkWVitCCwcUmpIn5rH9qyIWd0vNhVS4sE= size:1998
aki:AA0BD78CA2CAA5EB8FF3B9E823774DC1B74DEBEB seqnum:0550 thisupdate:1757926912 sia:rsync:
//rpki.ripe.net/repository/DEFAULT/c3/978837-1023-4b39-9b37-a108dbe0fd1f/1/qgvXjKLKpeuP87
noI3dNwbdN6-s.mft
                              hash:AAIINi/FNk2gG24RmitkMFRwPv2mr4OuSOeTnChnM2E= size:1924
aki:4C71E9F23EEACA68C721CBE42806100B75E15770 seqnum:0544 thisupdate:1757908897 sia:rsync:
//rpki.ripe.net/repository/DEFAULT/16/94f662-506b-4ec9-a7aa-c9b7eb2cea57/1/THHp8j7qymjHIc
vkKAYQC3XhV3A.mft
                              hash:AAJs57If1EkFg4TkLLhdFLBSLvLrxo0FRMM/Fx18rkg= size:2268
aki:8E8DD5B2C101B983A799C938AA0E40A651FCBE07 seqnum:012F thisupdate:1757912489 sia:rsync:
//rpki.apnic.net/member_repository/A9156334/E7CDDFCEC8D811EE9F4AB96DC4F9AE02/jo3VssEBuYOn
mck4qg5AplH8vgc.mft
                              hash:AAKTeG0+jnyS87E1Osv5VNS/F1iN8+nS02GdayrB8G4= size:2240
aki:578D7DC9779B27815EBB5C3F5FCF07CC583708B3 seqnum:0121 thisupdate:1757895655 sia:rsync:
//rsync.paas.rpki.ripe.net/repository/eca25350-9e3f-46e1-ae2b-1c61b6ea5ae2/0/578D7DC9779B
27815EBB5C3F5FCF07CC583708B3.mft
                              hash:AAXTLjblol0VOOylIGkAKdUHno9TwstslDfkg818LFQ= size:3473
aki:1B9F21BA04E5E3077B391E3D48FF5B0A50B3369F seqnum:0ABF thisupdate:1757934063 sia:rsync:
//rpki.ripe.net/repository/DEFAULT/22/f67a78-183d-494a-995e-f51c5fb3df9a/1/G58hugTl4wd7OR
49SP9bClCzNp8.mft
                              hash:AAeycRGtdSE/PcgWvFqb/hb4cX53cYCV8Vq1iOUs9F8= size:1924
aki:39EF4AA5977F1F37133E7310809FFF8935C0C8BC seqnum:03DE thisupdate:1757937704 sia:rsync:
//rpki.ripe.net/repository/DEFAULT/fe/148f71-ab2c-4409-ae1f-e7d5fce6bfac/1/Oe9KpZd_HzcTPn
MQgJ__iTXAyLw.mft
   ROA payload state hash:    MDI1RjhBMTc2NkE1RkM0RjA2NUExRUJFNzNBQUI4OEE=
   ROA payload entries:
                              192.35.94.0/24-32 AS 7
                              192.67.43.0/24-32 AS 7
                              194.32.69.0/24-32 AS 7
                              194.32.218.0/23-32 AS 7
                              194.34.138.0/24-32 AS 7
                              194.61.92.0/23-32 AS 7
                              2a0b:3b40::/29-128 AS 7
                              91.208.34.0/24 AS 8283
                              94.142.240.0/24 AS 8283
                              94.142.240.0/21 AS 8283
                              94.142.241.0/24 AS 8283
                              94.142.242.0/24 AS 8283
                              94.142.244.0/24 AS 8283
                              94.142.245.0/24 AS 8283
                              94.142.246.0/24 AS 8283
                              94.142.247.0/24 AS 8283
                              185.52.224.0/24 AS 8283
                              185.52.224.0/22 AS 8283
                              185.52.225.0/24 AS 8283
                              185.52.226.0/24 AS 8283
                              185.52.227.0/24 AS 8283
                              203.56.44.0/24 AS 8283
                              2001:678:688::/48 AS 8283
                              2a02:898::/32 AS 8283
                              2001:67c:208c::/48 AS 15562
                              2001:728:1808::/48 AS 15562
                              2a0e:b240::/48 AS 15562
                              2a0e:b240:118::/48 AS 15562
   ASPA payload state hash:   MTc4RkVEOTczODY5OThCRTE4MkExNzMxQ0FGRjBDQTE=
   ASPA payload entries:
```

```
   customer: 945 providers: 1421, 13213
   customer: 7719 providers: 945, 1421, 13213, 61138
   customer: 18607 providers: 20473, 207487
   customer: 39002 providers: 0
```

```
                              customer: 41720 providers: 174, 1299, 9002, 50877, 60068, 20
3446, 212508
   Trust anchor state hash:  RkZCMDlBRDA5QzJCOEU2MEI2NDcyMjIyNUE2NENCNUQ=
   Trust anchor keyids:      0B9CCA90DD0D7A8A37666B19217FE0D84037B7A2, E8552B1FD6D1A4F7E4
04C6D8E5680D1EBC163FC3
   Router key state hash:    QkE1RkI0NDlDRUZCNkJBMDBGMzYxMjc5NjJBMkVFQTY=
   Router keys:
                              asid:15562 ski:5D4250E2D81D4448D8A29EFCE91D29FF075EC9E2 pubk
ey:MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEgFcjQ/g//LAQerAH2Mpp+GucoDAGBbhIqD33wNPsXxnAGb+mtZ
7XQrVO9DQ6UlAShtig5+QfEKpTtFgiqfiAFQ==
                              asid:15562 ski:BE889B55D0B737397D75C49F485B858FA98AD11F pubk
ey:MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAE4FxJr0n2bux1uX1Evl+QWwZYvIadPjLuFX2mxqKuAGUhKnr7VL
LDgrE++l9p5eH2kWTNVAN22FUU3db/RKpE2w==
   Validation:               N/A
```

Appendix C. Implementation status

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

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in RFC 7942. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to RFC 7942, "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

* Example .ccr files were created by Job Snijders. A current example CCR (regenerated every few minutes) is available here: https://console.rpki-client.org/rpki.ccr

* A CCR serializer and deserializer implementation based on [rpki-client] was provided by Job Snijders.

Authors' Addresses

Job Snijders
Amsterdam
Netherlands
Email: job@sobornost.net

Bart Bakker
RIPE NCC
Netherlands
Email: bbakker@ripe.net

Tim Bruijnzeels
RIPE NCC
Netherlands
Email: tbruijnzeels@ripe.net