

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 15, 2021

M. Candela
NTT
R. Bush
IIJ & Arrcus
W. Kumari
Google
R. Housley
Vigil Security
October 12, 2020

Finding and Using Geofeed Data
draft-ymbk-opsawg-finding-geofeeds-04

Abstract

This document describes how to find and authenticate geofeed data.

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 April 15, 2021.

Copyright Notice

Copyright (c) 2020 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 Simplified BSD License text as described in [Section 4.e](#) of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

- [1. Introduction](#) [2](#)
- [1.1. Requirements Language](#) [2](#)
- [2. Geofeed Files](#) [3](#)
- [3. inetnum: Class](#) [3](#)
- [4. Authenticating Geofeed Data](#) [4](#)
- [5. Operational Considerations](#) [5](#)
- [6. Security Considerations](#) [6](#)
- [7. IANA Considerations](#) [6](#)
- [8. Acknowledgements](#) [7](#)
- [9. References](#) [7](#)
- [9.1. Normative References](#) [7](#)
- [9.2. Informative References](#) [8](#)
- [Appendix A. Example](#) [8](#)
- Authors' Addresses [17](#)

1. Introduction

Providers of Internet content and other services may wish to customize those services based on the geographic location of the user of the service. This is often done using the source IP address used to contact the service. Also, infrastructure and other services might wish to publish the locale of their services. [RFC8805] defines geofeed, a syntax to associate geographic locales with IP addresses. But it does not specify how to find the relevant geofeed data given an IP address.

This document specifies how to augment the Routing Policy Specification Language (RPSL) [RFC2622] inetnum: class [INETNUM] to refer to geofeed data, and how to prudently use them. In all places inetnum: is used, inet6num: should also be assumed [INET6NUM].

An optional, but utterly awesome, means for authenticating geofeed data is also defined.

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. Geofeed Files

Geofeed files are described in [[RFC8805](#)]. They provide a facility for an IP address resource 'owner' to associate those IP addresses to geographic locale(s).

Content providers and other parties who wish to locate an IP address to a geographic locale need to find the relevant geofeed data. In [Section 3](#) this document specifies how to find the relevant geofeed file given an IP address.

Geofeed data for large providers with significant horizontal scale and high granularity can be quite large. The size of a file can be even larger if an unsigned geofeed file combines data for many prefixes, as may be likely if the location data are maintained by a different department than address management, dual IPv4/IPv6 spaces are represented, etc.

This document also suggests optional data for geofeed files to provide stronger authenticity to the data.

3. inetnum: Class

RPSL, [[RFC2622](#)], as used by the Regional Internet Registries (RIRs), has been augmented with the inetnum: [[INETNUM](#)] and the inet6num: [[INET6NUM](#)] classes; each of which describes an IP address range and its attributes.

Ideally, RPSL would be augmented to define a new RPSL geofeed: attribute in the inetnum: class. Until such time, this document defines the syntax of a Geofeed remarks: attribute which contains an HTTPS URL of a geofeed file. The format MUST be as in this example, "remarks: Geofeed " followed by a URL which will vary.

```
inetnum: 192.0.2.0/24 # example
remarks: Geofeed https://example.com/geofeed.csv
```

Any particular inetnum: object MAY have, at most, one geofeed reference, whether a remark: or a proper geofeed: attribute when one is defined.

inetnum: objects form a hierarchy, see [[INETNUM](#)] [Section 4.2.4.1](#), Hierarchy of INETNUM Objects. Geofeed references SHOULD be at the lowest applicable inetnum: object. When fetching, the most specific inetnum: object with a geofeed reference MUST be used.

When geofeed references are provided by multiple inetnum: objects which have identical address ranges, then the geofeed reference on

the inetnum: with the most recent last-modified: attribute SHOULD be preferred.

It is significant that geofeed data may have finer granularity than the inetnum: which refers to them.

Currently, the registry data published by ARIN is not the same RPSL as the other registries; therefore, when fetching from ARIN via FTP [[RFC0959](#)], whois [[RFC3912](#)], RDAP [[RFC7482](#)], or whatever, the "NetRange" attribute/key must be treated as "inetnum" and the "Comment" attribute must be treated as "remarks".

4. Authenticating Geofeed Data

The question arises of whether a particular geofeed data set is valid, i.e. authorized by the 'owner' of the IP address space and is authoritative in some sense. The inetnum: which points to the geofeed file provides some assurance. Unfortunately the RPSL in many repositories is weakly authenticated at best. An approach where RPSL was signed a la [[RFC7909](#)] would be good, except it would have to be deployed by all RPSL registries.

An optional authenticator MAY be appended to a geofeed file. It would be essentially a digest of the main body of the file signed by the private key of the relevant RPKI certificate for the covering address range. One needs a format that bundles the relevant RPKI certificate with the signature and the digest of the geofeed text.

Borrowing detached signatures from [[RFC5485](#)], after text file canonicalization (Sec 2.2), the Cryptographic Message Syntax (CMS) [[RFC3852](#)] would be used to create a detached DER encoded signature which is then BASE64 encoded and line wrapped to 72 or fewer characters.

Both the address ranges of the signing certificate and of the inetnum: MUST cover all prefixes in the geofeed file; and the address range of the signing certificate must cover that of the inetnum:.

An address range A 'covers' address range B if the range of B is identical to or a subset of A. 'Address range' is used here because inetnum: objects and RPKI certificates need not align on CIDR prefix boundaries, while those of geofeed lines must.

As the signer would need to specify the covered RPKI resources relevant to the signature, the RPKI certificate covering the inetnum: object's address range would be included in the [[RFC3852](#)] CMS SignedData certificates field.

Identifying the private key associated with the certificate, and getting the department with the HSM to sign the CMS blob is left as an exercise for the implementor. On the other hand, verifying the signature requires no complexity; the certificate, which can be validated in the public RPKI, has the needed public key.

Until [\[RFC8805\]](#) is updated to formally define such an appendix, it MUST be 'hidden' as a series of "#" comments at the end of the geofeed file. This is a cryptographically incorrect, albeit simple example. A correct and full example is in [Appendix A](#).

```
# RPKI Signature: 192.0.2.0/24
# MIIGlwYJKoZIhvcNAQcCoIIGiDCCBoQCAQMxDALBgIghkgBZQMEAgEwDQYLKoZ
# IhvcNAQkQAS+gggSxMIIErTCCA5WgAwIBAgIUJ605QIPX8rW5m4Zwx3WyuW7hZu
...
# imwYkXpiMxw44EZqDjl36MiWsRDLdgoijBBcGbibwyAfGeR46k5raZCGvxG+4xa
# 08PDTxTfIYwAnBjRBKAqAZ7yX5xHfm58jUXsZJ7Ileq1S7G6Kk=
# End Signature: 192.0.2.0/24
```

5. Operational Considerations

To create the needed inetnum: objects, an operator wishing to register the location of their geofeed file needs to coordinate with their RIR/NIR and/or any provider LIR which has assigned prefixes to them. RIRs/NIRs provide means for assignees to create and maintain inetnum: objects. They also provide means of [sub-]assigning IP address resources and allowing the assignee to create whois data, including inetnum: objects, and thereby referring to geofeed files.

The geofeed files SHOULD be published over and fetched using https [\[RFC2818\]](#).

When using data from a geofeed file, one MUST ignore data outside of the referring inetnum: object's inetnum: attribute address range.

Iff the geofeed file is not signed per [Section 4](#), then multiple inetnum: objects MAY refer to the same geofeed file, and the consumer MUST use only geofeed lines where the prefix is covered by the address range of the inetnum: object they have followed.

To minimize the load on RIR whois [\[RFC3912\]](#) services, use of the RIR's FTP [\[RFC0959\]](#) services SHOULD be the preferred access. This also provides bulk access instead of fetching with a tweezers.

Currently, geolocation providers have bulk whois data access at all the RIRs. An anonymized version of such data is openly available for all RIRs except ARIN, which requires an authorization. However, for users without such authorization the same result can be achieved with

extra RDAP effort. There is open source code to pass over such data across all RIRs, collect all geofeed references, and process them [[geofeed-finder](#)].

An entity fetching geofeed data using these mechanisms MUST NOT do frequent real-time look-ups to prevent load on RPSL and geofeed servers. [[RFC8805](#)] [Section 3.4](#) suggests use of the [[RFC7234](#)] HTTP Expires Caching Header to signal when geofeed data should be refetched. As the data change very infrequently, in the absence of such an HTTP Header signal, collectors MUST NOT fetch more frequently than weekly. It would be wise not to fetch at magic times such as midnight UTC, the first of the month, etc., because too many others are likely to do the same.

6. Security Considerations

It would be generally prudent for a consumer of geofeed data to also use other sources to cross-validate the data. All of the Security Considerations of [[RFC8805](#)] apply here as well.

As mentioned in [Section 4](#), many RPSL repositories have weak if any authentication. This would allow spoofing of inetnum: objects pointing to malicious geofeed files. [Section 4](#) suggests an unfortunately complex method for stronger authentication based on the RPKI.

If an inetnum: for a wide prefix (e.g. a /16) points to an RPKI-signed geofeed file, a customer or attacker could publish a unsigned equal or narrower (e.g. a /24) inetnum: in a whois registry which has weak authorization.

The RPSL providers have had to throttle fetching from their servers due to too-frequent queries. Usually they throttle by the querying IP address or block. Similar defenses will likely need to be deployed by geofeed file servers.

7. IANA Considerations

IANA is asked to register object identifiers for one content type in the "SMI Security for S/MIME CMS Content Type (1.2.840.113549.1.9.16.1)" registry as follows:

Description	OID	Specification
id-ct-geofeedCSVwithCRLF	1.2.840.113549.1.9.16.1.47	[RFC-TBD]

8. Acknowledgements

Thanks to Rob Austein for CMS and detached signature clue. George Michaelson for the first, and a substantial, external review. Erik Kline who was too shy to agree to co-authorship. Additionally, we express our gratitude to early implementors, including Menno Schepers, Flavio Luciani, Eric Dugas, and Kevin Pack. Also to geolocation providers that are consuming geofeeds with this described solution, Jonathan Kosgei (ipdata.co), and Ben Dowling (ipinfo.io).

9. References

9.1. Normative References

- [INET6NUM] RIPE, "Description of the INET6NUM Object", <<https://www.ripe.net/manage-ips-and-asns/db/support/documentation/ripe-database-documentation/rpsl-object-types/4-2-descriptions-of-primary-objects/4-2-3-description-of-the-inet6num-object>>.
- [INETNUM] RIPE, "Description of the INETNUM Object", <<https://www.ripe.net/manage-ips-and-asns/db/support/documentation/ripe-database-documentation/rpsl-object-types/4-2-descriptions-of-primary-objects/4-2-4-description-of-the-inetnum-object>>.
- [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>>.
- [RFC2622] Alaettinoglu, C., Villamizar, C., Gerich, E., Kessens, D., Meyer, D., Bates, T., Karrenberg, D., and M. Terpstra, "Routing Policy Specification Language (RPSL)", [RFC 2622](#), DOI 10.17487/RFC2622, June 1999, <<https://www.rfc-editor.org/info/rfc2622>>.
- [RFC2818] Rescorla, E., "HTTP Over TLS", [RFC 2818](#), DOI 10.17487/RFC2818, May 2000, <<https://www.rfc-editor.org/info/rfc2818>>.
- [RFC3852] Housley, R., "Cryptographic Message Syntax (CMS)", [RFC 3852](#), DOI 10.17487/RFC3852, July 2004, <<https://www.rfc-editor.org/info/rfc3852>>.

- [RFC5485] Housley, R., "Digital Signatures on Internet-Draft Documents", [RFC 5485](#), DOI 10.17487/RFC5485, March 2009, <<https://www.rfc-editor.org/info/rfc5485>>.
- [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>>.
- [RFC8805] Kline, E., Duleba, K., Szamonek, Z., Moser, S., and W. Kumari, "A Format for Self-Published IP Geolocation Feeds", [RFC 8805](#), DOI 10.17487/RFC8805, August 2020, <<https://www.rfc-editor.org/info/rfc8805>>.

[9.2.](#) Informative References

- [geofeed-finder]
Massimo Candela, "geofeed-finder",
<<https://github.com/massimocandela/geofeed-finder>>.
- [RFC0959] Postel, J. and J. Reynolds, "File Transfer Protocol", STD 9, [RFC 959](#), DOI 10.17487/RFC0959, October 1985, <<https://www.rfc-editor.org/info/rfc959>>.
- [RFC3912] Daigle, L., "WHOIS Protocol Specification", [RFC 3912](#), DOI 10.17487/RFC3912, September 2004, <<https://www.rfc-editor.org/info/rfc3912>>.
- [RFC7234] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Caching", [RFC 7234](#), DOI 10.17487/RFC7234, June 2014, <<https://www.rfc-editor.org/info/rfc7234>>.
- [RFC7482] Newton, A. and S. Hollenbeck, "Registration Data Access Protocol (RDAP) Query Format", [RFC 7482](#), DOI 10.17487/RFC7482, March 2015, <<https://www.rfc-editor.org/info/rfc7482>>.
- [RFC7909] Kisteleki, R. and B. Haberman, "Securing Routing Policy Specification Language (RPSL) Objects with Resource Public Key Infrastructure (RPKI) Signatures", [RFC 7909](#), DOI 10.17487/RFC7909, June 2016, <<https://www.rfc-editor.org/info/rfc7909>>.

[Appendix A.](#) Example

This appendix provides an example, including a trust anchor, a CA certificate subordinate to the trust anchor, an end-entity

certificate subordinate to the CA for signing the geofeed, and a detached signature.

The trust anchor is represented by a self-signed certificate. As usual in the RPKI, the trust anchor has authority over all IPv4 address blocks, all IPv6 address blocks, and all AS numbers.

-----BEGIN CERTIFICATE-----

MIIEPjCCAYagAwIBAgIUPSUFJ4e/7pKZ6E14aBdkbYzms1gwDQYJKoZIhvcNAQEL
BQAwwFTETMBEGA1UEAxMKZXhhbXBsZS10YTAeFw0yMDA5MDMxODU0NTRaFw0zMDA5
MDExODU0NTRaMBUxEzARBgNVBAMTCmV4YW1wbGUtdGEwggEiMA0GCSqGSIb3DQEB
AQUAA4IBDwAwggEKAoIBAQCeLmMMDcGBhqn/a3VrNAoKMr1HVLKxGoG7VF/13HZJ
0tw0bUZlh3Jz+XeD+kNAURhELWTrsgdTkQQfinqOuRemxTL55+x7nLpe5nmwaBH
XqqDOHubmkbAGanGcm6T/rD9KNk1Z46Uc2p7UYu0fwN00mo0aqFL2FSyvvZwziNe
g7ELYZ4a3LvGn81JfP/JvM6pgtoMNuee5RV6TWaz7LV304ICj8Bhphy/HFp0A1rb
09gs8CUMgqz+RroAIA8cV8gbF/fPCz90fl7Gdmib679JxxFrW4wRj0nMjgJmsZXq
jaVc0g70Rc+eIACHw7Uroc6h7Y7lGj0kDZF75j0mLQa3AgMBAAGjggGEMIIIBgDAd
BgNVHQ4EFgQU3hNEuwwUGNCHY1TBatcUR03pNdYwHwYDVR0jBBgwFoAU3hNEuwwU
GNCHY1TBatcUR03pNdYwDwYDVR0TAQH/BAUwAwEB/zA0BgNVHQ8BAf8EBAMCAQYw
GAYDVR0gAQH/BA4wDDAKBggrBgEFBQc0AjCBuQYIKwYBBQUHAQsEgawwgakwPgYI
KwYBBQUHMAQGMnJzew5j0i8vcnBraS5leGFtcGxlLm5ldC9yZXBvc2l0b3J5L2V4
YW1wbGUtdGEubWZ0MDUGCCsGAQUFBzAnhilodHRwczovL3JyZHAuZXhhbXBsZS5u
ZXQvbm90aWZpY2F0aW9uLnhtbDAwBggrBgEFBQcwBYykkcnN5bmM6Ly9ycGtpLmV4
YW1wbGUubmV0L3JlcG9zaXRvcnkVMcCcGAQUFBwEHAQH/BBgwFjAJBAIAATAD
AwEAMAKEAgACMAMDAQAwwHgYIKwYBBQUHAQgEEjAQoA4wDDAKAgEAAgUA/////zAN
BgkqhkiG9w0BAQsFAA0CAQEAgZFQ0Sf3CI5Hwev61AUWHY0Fnny69PuDTq+WnhDe
xX5rpjSDRrs5L756KSKJca0J36lz045lf0PSY9fH6x30pnipaqRA7t5rApky24jH
cSUA9iRednzxhVyGjWknfAKyNo2MYfa0AT0db1GjyLKb0ADI9FowtHBUu+60ykcM
Quz66XrzxtmxlrRcAnbv/HtV17q0d4my6q5yjTPR1dmYN9oR/2ChlXtGE6uQVguA
rvNZ5CwiJ1TgGGTB7T8ORHwWU6dGTC0jk2rESAaikmLi1roZSNC21fckhapEit1a
x8CyivXjcVc5e0AmS1rJfL6LIfwmtive/N/eBtIM92HkBA==

-----END CERTIFICATE-----

The CA certificate is issued by the trust anchor. This certificate grants authority over one IPv4 address block (192.0.2.0/24) and two AS numbers (64496 and 64497).

-----BEGIN CERTIFICATE-----

```

MIIFBzCCA++gAwIBAgIUcyCzS10hdfG65kbRq7toQAvRDkQwDQYJKoZIhvcNAQEL
BQAwFTETMBEGA1UEAxMKZXhhbXBsZS10YTAeFw0yMDA5MDMxOTYMTaFw0yMTA5
MDMxOTYMTaMDMxMTAvBgNVBAMTKDNBQ0UyQ0VGNEZCMjFCN0QxMUUzRTE4NEVG
QzFFMjk3QjM3Nzg2NDIwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDc
zz1qwTxC2ocw5rqp8ktm2XyYkl8riBVuqLXwfefTxsR2YFpgz9vkYUd5Az9EVEG7
6wGIyZbtmhK63eEeaqbKz2GHub467498BXeVrYys0+YuIGgCEYKznNDZ4j5aaDbo
j5+4/z0Qvv6HEsxQd0f8br6lKJwgeRM6+fm7796HNPB0aqD7Zj9NRCLXjbB0DCgJ
liH6rXMKR86ofglL9V2mRjesvhdKYgkGb0if9rvxVpLJ/6zdru5CE9yeuJZ59l+n
YH/r6PzdJ4Q7yKrJX8qd6A60j4+biaU4MQ72KpsjhQNTTqF/HRwi0N54GDaknEwE
TnJQHgLDJYqww9yKwtjjAgMBAAGjggIvMIICKzAdBgNVHQ4EFgQUOs4s70+yG30R
4+GE78Hil7N3hkIwHwYDVR0jBBgwFoAU3hNEuuvUGNCHY1TBatCUR03pNdYwDwYD
VR0TAQH/BAUwAwEB/zA0BgNVHQ8BAf8EBAMCAQYwGAYDVR0gAQH/BA4wDDAKBggr
BgEFBQC0AjBhBgNVHR8EWjBYMFAgVKBShlByc3luYzovL3Jwa2kuZXhhbXBsZS5u
ZXQvcnVwb3NpdG9yeS8zQUFMkNFRjRGQjIxQjdEMTFFM0UxODRFRkMxRTI5N0Iz
Nzc4NjQyLmNybDB0BggrBgEFBQcBAQRCEAwPgYIKwYBBQUHMAKGMnJzew5j0i8v
cnBraS5leGFtcGxLLm5ldC9yZXBvc2l0b3J5L2V4YW1wbGUtdGEuY2VyMIG5Bggr
BgEFBQcBCwSBrdCBqTA+BggrBgEFBQcwCoYycnN5bmM6Ly9ycGtpLmV4YW1wbGUu
bmV0L3JlL29zaXRvcnkxZXhhbXBsZS10YTAeFw0yMDA5MDMxOTYMTaFw0yMTA5
MDMxOTYMTaMDMxMTAvBgNVBAMTKDNBQ0UyQ0VGNEZCMjFCN0QxMUUzRTE4NEVG
QzFFMjk3QjM3Nzg2NDIwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDc
0i8vcnJkcC5leGFtcGxLLm5ldC9ub3RpZmljYXRpb24ueG1sMDAGCCsGAQUFBzAF
hiRyc3luYzovL3Jwa2kuZXhhbXBsZS5uZXQvcnVwb3NpdG9yeS8wHwYIKwYBBQUH
AQcBAf8EEDA0MAwEAgaBMAyDBADAAAIwHgYIKwYBBQUHAQgEEjAQoA4wDDAKAgMA
+/ACAwd78TANBgkqhkiG9w0BAQsFAA0CAQEAnLu+d1ZsUTiX3YWGueTHIalw4ad0
Kupi7pYMV2nXbxNGmdJMol9BkzVz9tj55ReMghUU4YLm/ICYe4fz5e0T8o9s/vIm
cGS29+WoGuiznMitpvbS/379gaMezk6KpqjH6Brw6meMqy09phmcmvm3x3WTmx09
mLlQneMptwk8qSYcnMUMGLJs+cVqmk0a3sWRdw8WrGu6QqYtQz3HFZQojF06YzEq
V/dBdCfDEOWTfVl2n2XqhoJl/oEBdC4uu2G0qRk3+WVs+uwVHP0Ttsbt7TzFgZfY
yxqv0g6QoldxZVZmHhncKmETu/BqCDGJot9may31ukrx34Bu+XFMVihm0w==

```

-----END CERTIFICATE-----

The end-entity certificate is issued by the CA. This certificate grants signature authority for one IPv4 address block (192.0.2.0/24). Signature authority for AS numbers is not needed for geofeed data signatures, so no AS numbers are included in the certificate.

-----BEGIN CERTIFICATE-----

MIIERtCCA5WgAwIBAgIUJ605QIPX8rW5m4Zwx3WyuW7hZuMwDQYJKoZIhvcNAQEL
BQAwMzExMC8GA1UEAxMoM0FDRTJDRUY0RkIyMUI3RDExRTNFMTg0RUZDMUUYOTdC
Mzc3ODY0MjAeFw0yMDA5MDMxOTA1MTdaFw0yMTA2MzAxOTA1MTdaMDMxMTAvBgNV
BAMTKDkxNDY1MkEzQkQ1MUMxNDQyNjAxOTg4ODlGNUM0NUFCRjA1M0ExODcwggEi
MA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCycTqrOb/qB2W3i3Ki8PhA/DEW
yii2TgGo9pgCw09lsIRI6Zb/k+aSiWWP9kSczlcQgtPCVwr62hTQZCIowBN0BL0c
K0/5k1imJdi5qdM3nvKswM8CnoR11vB8pQFwruZmr5xphXRvE+mzuJVLgu2V1upm
BXuWloeymudh6WwJ+GDjwPX03RiXBejBr0FNXhaFLe08y4DPfr/S/tXJOBm7QzQp
tmbPLYtGfprYu45LiFFqP94UeLpISfXd36AKGzqTFcc3EW9l5UFE1MFLlnoEog
qtoLoKABt0Ik0FGKeC/EgeaBdWLe469ddC9rQft5w6g6cmxG+aYDdIEB34zrAgMB
AAGjggG3MIIIBszAdBgNVHQ4EFgQUkUZSo71RwUQmAZiIn1xFq/BToYcwHwYDVR0j
BBgwFoAU0s4s70+yG30R4+GE78Hil7N3hkIwDAYDVR0TAQH/BAIwADA0BgNVHQ8B
Af8EBAMCB4AwGAYDVR0gAQH/BA4wDDAKBggrBgEFBQc0AjBhBgNVHR8EwjBYMFag
VKBShlByc3luYzovL3Jwa2kuZXhhbXBsZS5uZXQvcvVwb3NpdG9yeS8zQUFMkNF
RjRGQjIxQjdEMTFM0UxODRFRkMxRTI5N0IzNzc4NjQyLmNyDBsBggrBgEFBQcB
AQRgMF4wXAYIKwYBBQUHMAKGUHZew5j0i8vcnBraS5leGFtcGxlLm5ldC9yZXBv
c2l0b3J5LzNBQ0UyQ0VGNZCMjFCN0QxMUUZrTE4NEVGGZFFMjk3QjM3Nzg2NDIu
Y2VyMCEGCCsGAQUFBwEHAQH/BBIWEDAGBAIAAQUAMAYEAgACBQAwrQYIKwYBBQUH
AQsEOTA3MDUGCCsGAQUFBzANhilodHRwczovL3JyZHAuZXhhbXBsZS5uZXQvbm90
awZpY2F0aw9uLnhthbDANBgkqhkiG9w0BAQsFAA0CAQEABR2T0qT2V1ZlsZjj+yHP
TArIVBECZFScdP+bJTse85TqYiblMsNS9yEu2SNbaZMNLuSSiAffYoooh4nIYq/Rh
6+xGs1n427JZUokoeLtY0UUb2fIsua9JF08YGTnpqDMGe+xnpbJ0SCSoBlJCIj+b
+YS8WXjEht2Kw6wyA/BcNS8adS2pEUwC2cs/WcwzgbttnkcnG7/wkrQ3oqzpc1ar
Kelyz7PGIIXJGy9nF8C3/aaaEpHd7UgIyvXYuCY/lqWTm97jDxgGIYGC7660mtf0
Mk8YF6kUU+td2dDQsMztc0xbzqiGnicmeJfBwG2li600vorW4d5iIOTKpQyqfh4
5Q==

-----END CERTIFICATE-----

The end-entity certificate is displayed below in detail. For brevity, the other two certificates are not.

0 1197: SEQUENCE {
4 917: SEQUENCE {
8 3: [0] {
10 1: INTEGER 2
: }
13 20: INTEGER 27AD394083D7F2B5B99B8670C775B2B96EE166E3
35 13: SEQUENCE {
37 9: OBJECT IDENTIFIER
: sha256WithRSAEncryption (1 2 840 113549 1 1 11)
48 0: NULL
: }
50 51: SEQUENCE {
52 49: SET {
54 47: SEQUENCE {
56 3: OBJECT IDENTIFIER commonName (2 5 4 3)
61 40: PrintableString


```

      :      '3ACE2CEF4FB21B7D11E3E184EFC1E297B3778642'
      :      }
      :      }
      :      }
103  30: SEQUENCE {
105  13:   UTCTime 03/09/2020 19:05:17 GMT
120  13:   UTCTime 30/06/2021 19:05:17 GMT
      :   }
135  51: SEQUENCE {
137  49:   SET {
139  47:     SEQUENCE {
141   3:       OBJECT IDENTIFIER commonName (2 5 4 3)
146  40:       PrintableString
      :       '914652A3BD51C144260198889F5C45ABF053A187'
      :       }
      :     }
      :   }
188 290: SEQUENCE {
192  13:   SEQUENCE {
194   9:     OBJECT IDENTIFIER rsaEncryption
      :     (1 2 840 113549 1 1 1)
205   0:     NULL
      :   }
207 271: BIT STRING, encapsulates {
212 266:   SEQUENCE {
216 257:     INTEGER
      :     00 B2 71 34 2B 39 BF EA 07 65 B7 8B 72 A2 F0 F8
      :     40 FC 31 16 CA 28 B6 4E 01 A8 F6 98 02 C0 EF 65
      :     B0 84 48 E9 96 FF 93 E6 92 89 65 8F F6 44 9C CE
      :     57 10 82 D3 C2 57 0A FA DA 14 D0 64 22 28 C0 13
      :     74 04 BD 1C 2B 4F F9 93 58 A6 25 D8 B9 A9 D3 37
      :     9E F2 AC C0 CF 02 9E 84 75 D6 F0 7C A5 01 70 AE
      :     E6 66 AF 9C 69 85 74 6F 13 E9 B3 B8 95 4B 82 ED
      :     95 D6 EA 66 05 7B 96 96 87 B2 9A E7 61 E9 65 89
      :     F8 60 E3 C0 F5 CE DD 18 97 05 E8 C1 AC E1 4D 5E
      :     16 85 2D ED 3C CB 80 CF 7E BF D2 FE D5 C9 38 19
      :     BB 43 34 29 B6 66 CF 2D 8B 46 7E 9A D8 BB 8E 65
      :     88 51 6A A8 FF 78 51 E2 E9 21 27 D7 77 7E 80 28
      :     6C EA 4C 50 9C 73 71 16 F6 5E 54 14 4D 4C 14 B9
      :     67 A0 4A 20 AA DA 0B A0 A0 01 B7 42 24 38 51 8A
      :     78 2F C4 81 E6 81 75 62 DE E3 AF 5D 74 2F 6B 41
      :     FB 79 C3 A8 3A 72 6C 46 F9 A6 03 74 81 01 DF 8C
      :     EB
477   3:     INTEGER 65537
      :   }
      : }
      : }
482 439: [3] {

```



```
486 435: SEQUENCE {
490 29: SEQUENCE {
492 3: OBJECT IDENTIFIER subjectKeyIdentifier (2 5 29 14)
497 22: OCTET STRING, encapsulates {
499 20: OCTET STRING
      : 91 46 52 A3 BD 51 C1 44 26 01 98 88 9F 5C 45 AB
      : F0 53 A1 87
      : }
      : }
521 31: SEQUENCE {
523 3: OBJECT IDENTIFIER authorityKeyIdentifier (2 5 29 35)
528 24: OCTET STRING, encapsulates {
530 22: SEQUENCE {
532 20: [0]
      : 3A CE 2C EF 4F B2 1B 7D 11 E3 E1 84 EF C1 E2 97
      : B3 77 86 42
      : }
      : }
      : }
554 12: SEQUENCE {
556 3: OBJECT IDENTIFIER basicConstraints (2 5 29 19)
561 1: BOOLEAN TRUE
564 2: OCTET STRING, encapsulates {
566 0: SEQUENCE {}
      : }
      : }
568 14: SEQUENCE {
570 3: OBJECT IDENTIFIER keyUsage (2 5 29 15)
575 1: BOOLEAN TRUE
578 4: OCTET STRING, encapsulates {
580 2: BIT STRING 7 unused bits
      : '1'B (bit 0)
      : }
      : }
584 24: SEQUENCE {
586 3: OBJECT IDENTIFIER certificatePolicies (2 5 29 32)
591 1: BOOLEAN TRUE
594 14: OCTET STRING, encapsulates {
596 12: SEQUENCE {
598 10: SEQUENCE {
600 8: OBJECT IDENTIFIER
      : resourceCertificatePolicy (1 3 6 1 5 5 7 14 2)
      : }
      : }
      : }
      : }
610 97: SEQUENCE {
612 3: OBJECT IDENTIFIER cRLDistributionPoints (2 5 29 31)
```



```
617 90:    OCTET STRING, encapsulates {
619 88:      SEQUENCE {
621 86:        SEQUENCE {
623 84:          [0] {
625 82:            [0] {
627 80:              [6]
                :            'rsync://rpki.example.net/repository/3ACE2CEF4F'
                :            'B21B7D11E3E184EFC1E297B3778642.crl'
                :          }
                :        }
                :      }
                :    }
709 108:   SEQUENCE {
711  8:    OBJECT IDENTIFIER authorityInfoAccess
                :    (1 3 6 1 5 5 7 1 1)
721 96:    OCTET STRING, encapsulates {
723 94:      SEQUENCE {
725 92:        SEQUENCE {
727  8:          OBJECT IDENTIFIER caIssuers (1 3 6 1 5 5 7 48 2)
737 80:          [6]
                :            'rsync://rpki.example.net/repository/3ACE2CEF4F'
                :            'B21B7D11E3E184EFC1E297B3778642.cer'
                :          }
                :        }
                :      }
                :    }
819 33:   SEQUENCE {
821  8:    OBJECT IDENTIFIER ipAddrBlocks (1 3 6 1 5 5 7 1 7)
831  1:    BOOLEAN TRUE
834 18:    OCTET STRING, encapsulates {
836 16:      SEQUENCE {
838  6:        SEQUENCE {
840  2:          OCTET STRING 00 01
844  0:          NULL
                :        }
846  6:        SEQUENCE {
848  2:          OCTET STRING 00 02
852  0:          NULL
                :        }
                :      }
                :    }
854 69:   SEQUENCE {
856  8:    OBJECT IDENTIFIER subjectInfoAccess
                :    (1 3 6 1 5 5 7 1 11)
866 57:    OCTET STRING, encapsulates {
```



```

868 55: SEQUENCE {
870 53: SEQUENCE {
872 8: OBJECT IDENTIFIER '1 3 6 1 5 5 7 48 13'
882 41: [6]
      : 'https://rrdp.example.net/notification.xml'
      : }
      : }
      : }
      : }
      : }
      : }
      : }
      : }
925 13: SEQUENCE {
927 9: OBJECT IDENTIFIER sha256WithRSAEncryption
      : (1 2 840 113549 1 1 11)
938 0: NULL
      : }
940 257: BIT STRING
      : 05 1D 93 D2 A4 F6 57 56 65 B1 98 E3 FB 21 CF 4C
      : 0A C8 54 11 02 64 54 82 74 FF 9B 25 3B 1E F3 94
      : EA 62 26 E5 32 C3 52 F7 21 2E D9 23 5B 69 93 0D
      : 2E E4 92 88 07 DF 62 8A 21 E2 72 18 AB F4 61 EB
      : EC 46 B3 59 F8 DB B2 59 52 89 28 78 BB 58 D1 45
      : 1B D9 F2 2C B9 AF 49 16 8F 18 19 39 E9 A8 33 06
      : 7B EC 67 A5 B2 74 48 24 A8 06 52 42 22 3F 9B F9
      : 84 BC 59 78 C4 1E DD 8A 5B AC 32 03 F0 5C 35 2F
      : 1A 75 2D A9 11 4C 02 D9 CB 3F 59 CC 33 81 BB 6D
      : 9E 47 27 1B BF F0 92 B4 37 A2 AC E9 0B 56 AB 29
      : E9 72 CF B3 C6 20 85 C9 1B 2F 67 17 C0 B7 FD A6
      : 9A 12 91 DD ED 48 08 CA F5 D8 B8 26 3F 96 A5 93
      : 9B DE E3 0F 18 06 21 81 82 EF AE B4 9A D7 CE 32
      : 40 7C 60 5E A4 51 4F AD 77 67 43 42 C3 33 B5 C3
      : B1 6F 3A A2 1A 78 9C 99 E2 5F 07 01 B6 96 2E 8E
      : D2 FA 2B 5B 87 79 88 83 93 2A 94 32 A9 F8 78 E5
      : }

```

To allow reproduction of the signature results, the end-entity private key is provided. For brevity, the other two private keys are not.

-----BEGIN RSA PRIVATE KEY-----

```

MIIEpQIBAAKCAQEAsnE0Kzm/6gdlt4tyovD4QPwxFsootk4BqPaYAsDvZbCES0mW
/5Pmkollj/ZEnM5XEILTwlck+toU0GQiKMATdAS9HctP+ZNYpiXYuanTN57yrMDP
Ap6EddbWfKUBcK7mZq+caYV0bxPps7iVS4LtlDbqZgV7lpaHsprnYellifhg48D1
zt0YlwXowazhTV4WhS3tPMuAz36/0v7VyTgZu0M0KbZmzy2LRn6a2Lu0ZYhRaqj/
eFHi6SEn13d+gChs6kxQnHNxFvZeVBRNTBS5Z6BKIKraC6CgAbdCJDhRingvxIHm
gXVi3u0vXXQva0H7ec0oOnJsRvmmA3SBAd+M6wIDAQABAoIBAQCyB0FeMuKm8bRo
18aKjFGSPEoZi53srIz5bvUgIi92TBLz7ZnzL6Iym26oJ+5th+lCHG0/dqLhXio
pI50C5Yc9TFbbLb/EC0suCuuqKFjZ8CD3GVsHozXKJeMM+/o5YZXQR0Rj6UnwT0z
oL/JE5pIGUCIgsXX6tz9s5BP3lUAvVQHsv6+vEVKLxQ3wj/1vIL80/CN036EV0GJ
mpkwmypGpjfECT9wbWo0yn3jxJb36+M/QjjUP28oNIVn/IKoPZRxnqchEbuuCJ651
IsaFSqtiThm4WZtvCH/IDq+6/dcMucmTjIRcYwW7fdHfjplllVPve9c/OmpWEQvF
t3ArWUt5AoGBANs4764yHxo4mctLIE7G7l/tf9bP4KKUiYw4R4ByEocuqMC4yhmt
MPCf0FLOqet710WckjP2L/7EKUe9yx7G5KmxAHY6j0jvcRkvGsl6lWF0sQ8p126M
Y9hmGzMOjtsdhAiMmOWKzjvm4WqfMgghQe+PnjjSVkgTt+7BxpIuGBAvAoGBANBg
26FF5cDLpix0d3Za1YXsOgguwCaw3Plvi7vUZRpa/zBMELEty0ebfakkIRWNm07l
nE+lAZwXm+29PTD0nqCFE91teyzjnQaL05kkAdJiFuVV3icL0Go399FrnJbKensm
FGSli+3KxQhCNIJJfgWzq4bE0ioAMjdGbYXzIYQFAoGBAM6tuDJ36KDU+hIS6wu6
02TPSfZhf/zPo3pCWQ78/QDb+Zdw4IEiqoBA7F4NPVLg9Y/H8UTx9r/veqe7hP0o
Ok7NpIzSmKTHkc5XfZ60Zn9OLF0kbaQ40a1kXoJdWEu2YR0aUlAe9F6/Rog6PHYz
vLE5qscRbu0XQhLkN+z7bg5bAoGBAKDsDbEdb/dbqbyaAYpmwhH2sdRSkphg7Niwc
DNm9qWa1J6Zw1+M87I6Q8naRREuU1IAVqqWHVLR/ROBQ6NTJ1Uc5/qFeT2XXUgkf
taMKv61tuyjZK3sTmznMh0HfzUpWjEhWnCEuB+ZYVdm052ZGw2A75RdrILL2+9Dc
PvDXVubRAoGAdqXeSwoLxuzZXzl8rsaKrQsTYaXn0WaZieU1SL5vVe8nK257UDqZ
E3ng2j5XPTUwli+aNGFEJGRoNtcQv0600/sFZUhu52sqq9mWVYZNh1TB5aP8X+pV
iFcZ0LUvQEcN6PA+YQK5FU11rAI1M0Gm5RDnVnUl0L2xfCYxb7FzV6Y=

```

-----END RSA PRIVATE KEY-----

Signing of "192.0.2.0/24,US,WA,Seattle," (terminated by CR and LF), yields the following detached CMS signature.


```

# RPKI Signature: 192.0.2.0/24
# MIIGlwYJKoZIhvcNAQcCoIIGiDCCBoQCAQMxDALBgIghkgBZQMEAgEwDQYLKoZ
# IhvcNAQkQAS+gggSxMIIErTCCA5WgAwIBAgIUJ605QIPX8rW5m4Zwx3WyuW7hZu
# MwdQYJKoZIhvcNAQELBQAwmZExMC8GA1UEAxMoM0FDRTJDRUY0RkIyMUI3RDExR
# TNFMTg0RUZDMUuyOTdCMzc3ODY0MjAeFw0yMDA5MDMxOTA1MTdaFw0yMTA2MzAx
# OTA1MTdaMDMxMTAvBgNVBAMTKDkxNDY1MkEzQkQ1MUMxNDQyNjAxOTg4ODlGNUM
# 0NUFCRjA1M0EXODcwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCycT
# QrOb/qB2W3i3Ki8PhA/DEWyii2TgGo9pgCw09lsIRI6Zb/k+aSiWWP9kSczlcQg
# tPCVwr62htQZCIowBN0BL0cK0/5k1imJdi5qdM3nvKswM8CnoR11vB8pQFwruZm
# r5xphXRvE+mzuJVLgu2V1upmBXuWloeymudh6WWJ+GDjwPX03RiXBejBr0FNXha
# FLe08y4DPfr/S/tXJOBm7QzQptmbPLYtGfprYu45liFFqQP94UeLpISfxD36AKG
# zqTFccc3EW9l5UFE1MFLnoEogqtoLoKABt0Ik0FGKeC/EgeaBdWLe469ddC9rQ
# ft5w6g6cmxG+aYDdIEB34zrAgMBAAGjggG3MIIBszAdBgNVHQ4EFgQUKUZSo71R
# wUQmAZiIn1xFq/BToYcwHwYDVR0jBBGwFoAU0s4s70+yG30R4+GE78Hil7N3hkI
# wDAYDVR0TAQH/BAIwADA0BgNVHQ8BAf8EBAMCB4AwGAYDVR0gAQH/BA4wDDAKBg
# grBgEFBQC0AjBhBgNVHR8EwjBYMfagVKBSHlByc3luYzovL3Jwa2kuZXhhbXBsZ
# S5uZXQvcnVwb3NpdG9yeS8zQUFMkNFRjRGQjIxQjdEMTFFM0UxODRFRkMxRTI5
# N0IzNzc4NjQyLmNybDBsBggRBgEFBQCBAQRGMF4wXAYIKwYBBQUHMAKGUHHJzew5
# j0i8vcnBraS5leGfctGxllm5ldC9yZXBvc2l0b3J5LzNBQ0UyQ0VGNEZCMjFCN0
# QxMUUZrTE4NEVGZzFFMjk3QjM3Nzg2NDIuY2VyMCEGCCsGAQUFBWEHAQH/BBiWE
# DAGBAIAAQUAMAYEAgACBQAwRQYIKwYBBQUHAQsEOTA3MDUGCCsGAQUFBzANhilo
# dHRwczovL3JyZHAuZXhhbXBsZS5uZXQvcnVwb3NpdG9wZy2F0aw9uLnhtbDANBgkqhki
# G9w0BAQsFAAOCAQEABR2T0qT2V1ZlsZjj+yHPTArIVBECZFSCdP+bJTse85TqYi
# blMsNS9yEu2SNbaZMNLuSSiAffYooH4nIYq/Rh6+xGs1n427JZUokoeLty0UUB2
# fIsua9JFo8YGtNpqDMGe+xnpbJ0SCSoBlJCIj+b+YS8WXjEht2KW6wyA/BcNS8a
# dS2pEUwC2cs/WcwzgbttnkcnG7/wkrQ3oqzpc1arKelz7PGIIXJGy9nF8C3/aa
# aEpHd7UgIyvXYuCY/lqWtm97jDxgGIYGC7660mtf0Mk8YF6kUU+td2dDQsMztc
# OxbzqiGnicmeJfBwG2li600vorw4d5iIOTKpQyqfh45TGCAaowggGmAgEDgBSRR
# lKjvVHBRcYBmIifXEwr8F0hhzALBgIghkgBZQMEAgGgazAaBkgqhkiG9w0BCQMx
# DQYLKoZIhvcNAQkQAS8wHAYJKoZIhvcNAQkFMQ8XDTIwMDkxMzE4NDUxMFowLwY
# JKoZIhvcNAQkEMSIIEICvi8p5S8ckg2wTRhDBQzGijjyqs5T6I+4vtBHypfcEWMA
# 0GCSqGSIb3DQEBAQUABIIBAHUra4PaJG42BD3hpF8U0usnV3Dg5NQh97SfyKtk7
# YHhhwu/936gkmAew8ODRtCddMvM0bWkj7/XeR+WkffaTF1EAdZ1L6REV+GLV91
# cYnFkT9ldn4wHqNNncfAehk5PCLYUuQ0gqjdJT1hdaoL83b3ttekyYIiwPmHE
# xRaNkSvKenlNqcriaaf3rbQy9dc2d1KxrL2429n134ICqjKeRnHkXXrCWdmyv/3
# imwYkXpiMxw44EZqDj136MiWsRDLdgoijBBcGbibwyAfGeR46k5raZCGvxG+4xa
# 08PDTxTfIYwAnBjRBKAqAZ7yX5xHfm58jUXsZJ7Ileq1S7G6Kk=
# End Signature: 192.0.2.0/24

```

Authors' Addresses

Massimo Candela
 NTT
 Siriusdreef 70-72
 Hoofddorp 2132 WT
 Netherlands

Email: massimo@ntt.net

Randy Bush
IIJ & Arccus
5147 Crystal Springs
Bainbridge Island, Washington 98110
United States of America

Email: randy@psg.com

Warren Kumari
Google
1600 Amphitheatre Parkway
Mountain View, CA 94043
US

Email: warren@kumari.net

Russ Housley
Vigil Security, LLC
516 Dranesville Road
Herndon, VA 20170
USA

Email: housley@vigilsec.com

