| | |
|---|---|
| Stream: | Internet Engineering Task Force (IETF) |
| RFC: | 9092 |
| Category: | Standards Track |
| Published: | July 2021 |
| ISSN: | 2070-1721 |
| Authors: | R. Bush | M. Candela | W. Kumari | R. Housley |
|---|---|---|---|---|
| | *IIJ & Arrcus* | *NTT* | *Google* | *Vigil Security* |

# RFC 9092
# Finding and Using Geofeed Data

## Abstract

This document specifies how to augment the Routing Policy Specification Language inetnum: class to refer specifically to geofeed data comma-separated values (CSV) files and describes an optional scheme that uses the Routing Public Key Infrastructure to authenticate the geofeed data CSV files.

## Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9092.

## Table of Contents



## 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) [RFC2725] inetnum: class to refer specifically to geofeed data CSV files and how to prudently use them. In all places inetnum: is used, inet6num: should also be assumed [RFC4012].

The reader may find [INETNUM] and [INET6NUM] informative, and certainly more verbose, descriptions of the inetnum: database classes.

An optional utterly awesome but slightly complex 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 locales.

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 [RFC8805] 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, if dual IPv4/IPv6 spaces are represented, etc.

Geofeed data do have privacy considerations (see Section 6); this process makes bulk access to those data easier.

This document also suggests an optional signature to strongly authenticate the data in the geofeed files.

## 3. inetnum: Class

The original RPSL specifications starting with [RIPE81], [RIPE181], and a trail of subsequent documents were written by the RIPE community. The IETF standardized RPSL in [RFC2622] and [RFC4012]. Since then, it has been modified and extensively enhanced in the Regional Internet Registry (RIR) community, mostly by RIPE [RIPE-DB]. Currently, change control effectively lies in the operator community.

The RPSL, and [RFC2725] and [RFC4012] used by the Regional Internet Registries (RIRs), specify the inetnum: database class. Each of these objects describes an IP address range and its attributes. The inetnum: objects form a hierarchy ordered on the address space.

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 of the inetnum: geofeed remarks: attribute

**MUST** be as in this example, "remarks: Geofeed ", where the token "Geofeed " **MUST** be case sensitive, followed by a URL that will vary, but it **MUST** refer only to a single geofeed [RFC8805] file.

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

While we leave global agreement of RPSL modification to the relevant parties, we specify that a proper geofeed: attribute in the inetnum: class **MUST** be "geofeed:" and **MUST** be followed by a single URL that will vary, but it **MUST** refer only to a single geofeed [RFC8805] file.

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

Registries **MAY**, for the interim, provide a mix of the remarks: attribute form and the geofeed: attribute form.

The URL uses HTTPS, so the WebPKI provides authentication, integrity, and confidentiality for the fetched geofeed file. However, the WebPKI can not provide authentication of IP address space assignment. In contrast, the RPKI (see [RFC6481]) can be used to authenticate IP space assignment; see optional authentication in Section 4.

Until all producers of inetnum: objects, i.e., the RIRs, state that they have migrated to supporting a geofeed: attribute, consumers looking at inetnum: objects to find geofeed URLs **MUST** be able to consume both the remarks: and geofeed: forms. The migration not only implies that the RIRs support the geofeed: attribute, but that all registrants have migrated any inetnum: objects from remarks: to geofeed: attributes.

Any particular inetnum: object **MUST** have, at most, one geofeed reference, whether a remarks: or a proper geofeed: attribute when it is implemented. If there is more than one, all are ignored.

If a geofeed CSV file describes multiple disjoint ranges of IP address space, there are likely to be geofeed references from multiple inetnum: objects. Files with geofeed references from multiple inetnum: objects are not compatible with the signing procedure in Section 4.

When geofeed references are provided by multiple inetnum: objects that have identical address ranges, then the geofeed reference on the inetnum: with the most recent last-modified: attribute **SHOULD** be preferred.

As inetnum: objects form a hierarchy, geofeed references **SHOULD** be at the lowest applicable inetnum: object covering the relevant address ranges in the referenced geofeed file. When fetching, the most specific inetnum: object with a geofeed reference **MUST** be used.

It is significant that geofeed data may have finer granularity than the inetnum: that refers to them. For example, an INETNUM object for an address range P could refer to a geofeed file in which P has been subdivided into one or more longer prefixes.

Currently, the registry data published by ARIN are not the same RPSL as that of the other registries (see [RFC7485] for a survey of the WHOIS Tower of Babel); therefore, when fetching from ARIN via FTP [RFC0959], WHOIS [RFC3912], the Registration Data Access Protocol (RDAP) [RFC9082], etc., 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 whether a particular geofeed [RFC8805] data set is valid, i.e., is authorized by the "owner" of the IP address space and is authoritative in some sense. The inetnum: that points to the geofeed [RFC8805] file provides some assurance. Unfortunately, the RPSL in many repositories is weakly authenticated at best. An approach where RPSL was signed per [RFC7909] would be good, except it would have to be deployed by all RPSL registries, and there is a fair number of them.

A single optional authenticator **MAY** be appended to a geofeed [RFC8805] file. It is a digest of the main body of the file signed by the private key of the relevant RPKI certificate for a covering address range. One needs a format that bundles the relevant RPKI certificate with the signature of the geofeed text.

The canonicalization procedure converts the data from their internal character representation to the UTF-8 [RFC3629] character encoding, and the <CRLF> sequence **MUST** be used to denote the end of a line of text. A blank line is represented solely by the <CRLF> sequence. For robustness, any non-printable characters **MUST NOT** be changed by canonicalization. Trailing blank lines **MUST NOT** appear at the end of the file. That is, the file must not end with multiple consecutive <CRLF> sequences. Any end-of-file marker used by an operating system is not considered to be part of the file content. When present, such end-of-file markers **MUST NOT** be processed by the digital signature algorithm.

Should the authenticator be syntactically incorrect per the above, the authenticator is invalid.

Borrowing detached signatures from [RFC5485], after file canonicalization, the Cryptographic Message Syntax (CMS) [RFC5652] would be used to create a detached DER-encoded signature that is then padded BASE64 encoded (as per Section 4 of [RFC4648]) and line wrapped to 72 or fewer characters. The same digest algorithm **MUST** be used for calculating the message digest on content being signed, which is the geofeed file, and for calculating the message digest on the SignerInfo SignedAttributes [RFC8933]. The message digest algorithm identifier **MUST** appear in both the SignedData DigestAlgorithmIdentifiers and the SignerInfo DigestAlgorithmIdentifier [RFC5652].

The address range of the signing certificate **MUST** cover all prefixes in the geofeed file it signs.

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 Classless Inter-Domain Routing (CIDR) [RFC4632] prefix boundaries, while those of the CSV lines in a geofeed file do.

As the signer specifies the covered RPKI resources relevant to the signature, the RPKI certificate covering the inetnum: object's address range is included in the [RFC5652] CMS SignedData certificates field.

Identifying the private key associated with the certificate and getting the department that controls the private key (which might be trapped in a Hardware Security Module (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. The trust anchors for the RIRs are expected to already be available to the party performing signature validation. Validation of the CMS signature on the geofeed file involves:

1. Obtaining the signer's certificate from the CMS SignedData CertificateSet [RFC5652]. The certificate SubjectKeyIdentifier extension [RFC5280] **MUST** match the SubjectKeyIdentifier in the CMS SignerInfo SignerIdentifier [RFC5652]. If the key identifiers do not match, then validation **MUST** fail.

   Validation of the signer's certificate **MUST** ensure that it is part of the current [RFC6486] manifest and that the resources are covered by the RPKI certificate.
2. Constructing the certification path for the signer's certificate. All of the needed certificates are expected to be readily available in the RPKI repository. The certification path **MUST** be valid according to the validation algorithm in [RFC5280] and the additional checks specified in [RFC3779] associated with the IP Address Delegation certificate extension and the Autonomous System Identifier Delegation certificate extension. If certification path validation is unsuccessful, then validation **MUST** fail.
3. Validating the CMS SignedData as specified in [RFC5652] using the public key from the validated signer's certificate. If the signature validation is unsuccessful, then validation **MUST** fail.
4. Verifying that the IP Address Delegation certificate extension [RFC3779] covers all of the address ranges of the geofeed file. If all of the address ranges are not covered, then validation **MUST** fail.

All of these steps **MUST** be successful to consider the geofeed file signature as valid.

As the signer specifies the covered RPKI resources relevant to the signature, the RPKI certificate covering the inetnum: object's address range is included in the CMS SignedData certificates field [RFC5652].

Identifying the private key associated with the certificate and getting the department with the Hardware Security Module (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.

The appendix **MUST** be hidden as a series of "#" comments at the end of the geofeed file. The following is a cryptographically incorrect, albeit simple, example. A correct and full example is in Appendix A.

```
# RPKI Signature: 192.0.2.0 - 192.0.2.255
# MIIGlwYJKoZIhvcNAQcCoIIGiDCCBoQCAQMxDTALBglghkgBZQMEAgEwDQYLKoZ
# IhvcNAQkQAS+gggSxMIIErTCCA5WgAwIBAgIUJ605QIPX8rW5m4Zwx3WyuW7hZu
...
# imwYkXpiMxw44EZqDjl36MiWsRDLdgoijBBcGbibwyAfGeR46k5raZCGvxG+4xa
# O8PDTxTfIYwAnBjRBKAqAZ7yX5xHfm58jUXsZJ7Ileq1S7G6Kk=
# End Signature: 192.0.2.0 - 192.0.2.255
```

The signature does not cover the signature lines.

The bracketing "# RPKI Signature:" and "# End Signature:" **MUST** be present following the model as shown. Their IP address range **MUST** match that of the inetnum: URL followed to the file.

[RPKI-RSC] describes and provides code for a CMS profile for a general purpose listing of checksums (a "checklist") for use with the Resource Public Key Infrastructure (RPKI). It provides usable, albeit complex, code to sign geofeed files.

[RPKI-RTA] describes a CMS profile for a general purpose Resource Tagged Attestation (RTA) based on the RPKI. While this is expected to become applicable in the long run, for the purposes of this document, a self-signed root trust anchor is used.

## 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 Regional Internet Registry (RIR) or National Internet Registry (NIR) and/or any provider Local Internet Registry (LIR) that has assigned address ranges to them. RIRs/NIRs provide means for assignees to create and maintain inetnum: objects. They also provide means of assigning or sub-assigning IP address resources and allowing the assignee to create WHOIS data, including inetnum: objects, thereby referring to geofeed files.

The geofeed files **MUST** be published via and fetched using HTTPS [RFC2818].

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

If and only if 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 lines in the geofeed file where the prefix is covered by the address range of the inetnum: object's URL it has followed.

If the geofeed file is signed, and the signer's certificate changes, the signature in the geofeed file **MUST** be updated.

It is good key hygiene to use a given key for only one purpose. To dedicate a signing private key for signing a geofeed file, an RPKI Certification Authority (CA) may issue a subordinate certificate exclusively for the purpose shown in Appendix A.

To minimize the load on RIR WHOIS [RFC3912] services, use of the RIR's FTP [RFC0959] services **SHOULD** be used for large-scale access to gather geofeed URLs. This also provides bulk access instead of fetching by brute-force search through the IP space.

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].

To prevent undue load on RPSL and geofeed servers, entity-fetching geofeed data using these mechanisms **MUST NOT** do frequent real-time lookups. Section 3.4 of [RFC8805] suggests use of the HTTP Expires header [RFC7234] to signal when geofeed data should be refetched. As the data change very infrequently, in the absence of such an HTTP Header signal, collectors **SHOULD NOT** fetch more frequently than weekly. It would be polite 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. Privacy Considerations

[RFC8805] geofeed data may reveal the approximate location of an IP address, which might in turn reveal the approximate location of an individual user. Unfortunately, [RFC8805] provides no privacy guidance on avoiding or ameliorating possible damage due to this exposure of the user. In publishing pointers to geofeed files as described in this document, the operator should be aware of this exposure in geofeed data and be cautious. All the privacy considerations of Section 4 of [RFC8805] apply to this document.

Where [RFC8805] provided the ability to publish location data, this document makes bulk access to those data readily available. This is a goal, not an accident.

## 7. Security Considerations

It is generally prudent for a consumer of geofeed data to also use other sources to cross validate the data. All the security considerations of [RFC8805] apply here as well.

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

For example, if an inetnum: for a wide address range (e.g., a /16) points to an RPKI-signed geofeed file, a customer or attacker could publish an unsigned equal or narrower (e.g., a /24) inetnum: in a WHOIS registry that has weak authorization, abusing the rule that the most-specific inetnum: object with a geofeed reference **MUST** be used.

If signatures were mandatory, the above attack would be stymied, but of course that is not happening anytime soon.

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.

# 8. IANA Considerations

IANA has registered 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:

| Decimal | Description | References |
|---|---|---|
| 47 | id-ct-geofeedCSVwithCRLF | RFC 9092 |

*Table 1*

# 9. References

## 9.1. Normative References

**[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>.

**[RFC2725]** Villamizar, C., Alaettinoglu, C., Meyer, D., and S. Murphy, "Routing Policy System Security", RFC 2725, DOI 10.17487/RFC2725, December 1999, <https://www.rfc-editor.org/info/rfc2725>.

**[RFC2818]** Rescorla, E., "HTTP Over TLS", RFC 2818, DOI 10.17487/RFC2818, May 2000, <https://www.rfc-editor.org/info/rfc2818>.

**[RFC3629]** Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, DOI 10.17487/RFC3629, November 2003, <https://www.rfc-editor.org/info/rfc3629>.

**[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/info/rfc3779>.

**[RFC4012]** Blunk, L., Damas, J., Parent, F., and A. Robachevsky, "Routing Policy Specification Language next generation (RPSLng)", RFC 4012, DOI 10.17487/RFC4012, March 2005, <https://www.rfc-editor.org/info/rfc4012>.

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

**[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/info/rfc5280>.

**[RFC5652]** Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, DOI 10.17487/RFC5652, September 2009, <https://www.rfc-editor.org/info/rfc5652>.

**[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>.

**[RFC6486]** Austein, R., Huston, G., Kent, S., and M. Lepinski, "Manifests for the Resource Public Key Infrastructure (RPKI)", RFC 6486, DOI 10.17487/RFC6486, February 2012, <https://www.rfc-editor.org/info/rfc6486>.

**[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>.

**[RFC8933]** Housley, R., "Update to the Cryptographic Message Syntax (CMS) for Algorithm Identifier Protection", RFC 8933, DOI 10.17487/RFC8933, October 2020, <https://www.rfc-editor.org/info/rfc8933>.

## 9.2. Informative References

**[GEOFEED-FINDER]** "geofeed-finder", commit 5f557a4, June 2021, <https://github.com/massimocandela/geofeed-finder>.

**[INET6NUM]** RIPE NCC, "Description of the INET6NUM Object", October 2019, <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 NCC, "Description of the INETNUM Object", June 2020, <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>.

**[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>.

**[RFC4632]** Fuller, V. and T. Li, "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan", BCP 122, RFC 4632, DOI 10.17487/RFC4632, August 2006, <https://www.rfc-editor.org/info/rfc4632>.

**[RFC5485]** Housley, R., "Digital Signatures on Internet-Draft Documents", RFC 5485, DOI 10.17487/RFC5485, March 2009, <https://www.rfc-editor.org/info/rfc5485>.

**[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>.

**[RFC7485]** Zhou, L., Kong, N., Shen, S., Sheng, S., and A. Servin, "Inventory and Analysis of WHOIS Registration Objects", RFC 7485, DOI 10.17487/RFC7485, March 2015, <https://www.rfc-editor.org/info/rfc7485>.

**[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>.

**[RFC9082]** Hollenbeck, S. and A. Newton, "Registration Data Access Protocol (RDAP) Query Format", STD 95, RFC 9082, DOI 10.17487/RFC9082, June 2021, <https://www.rfc-editor.org/info/rfc9082>.

**[RIPE-DB]** RIPE NCC, "RIPE Database Documentation", <https://www.ripe.net/manage-ips-and-asns/db/support/documentation/ripe-database-documentation>.

**[RIPE181]** RIPE NCC, "Representation Of IP Routing Policies In A Routing Registry", October 1994, <https://www.ripe.net/publications/docs/ripe-181>.

**[RIPE81]** RIPE NCC, "Representation Of IP Routing Policies In The RIPE Database", February 1993, <https://www.ripe.net/publications/docs/ripe-081>.

**[RPKI-RSC]** Snijders, J., Harrison, T., and B. Maddison, "Resource Public Key Infrastructure (RPKI) object profile for Signed Checklist (RSC)", Work in Progress, Internet-Draft, draft-ietf-sidrops-rpki-rsc-04, 31 May 2021, <https://datatracker.ietf.org/doc/html/draft-ietf-sidrops-rpki-rsc-04>.

**[RPKI-RTA]** Michaelson, G. G., Huston, G., Harrison, T., Bruijnzeels, T., and M. Hoffmann, "A profile for Resource Tagged Attestations (RTAs)", Work in Progress, Internet-Draft, draft-ietf-sidrops-rpki-rta-00, 21 January 2021, <https://datatracker.ietf.org/doc/html/draft-ietf-sidrops-rpki-rta-00>.

## Appendix A. Example

This appendix provides an example that includes 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 Autonomous System (AS) numbers.


```
-----BEGIN CERTIFICATE-----
MIIEPjCCAyagAwIBAgIUPsUFJ4e/7pKZ6E14aBdkbYzms1gwDQYJKoZIhvcNAQEL
BQAwFTETMBEGA1UEAxMKZXhhbXBsZS10YTAeFw0yMDA5MDMxODU0NTRaFw0zMDA5
MDExODU0NTRaMBUxEzARBgNVBAMTCmV4YW1wbGUtdGEwggEiMA0GCSqGSIb3DQEB
AQUAA4IBDwAwggEKAoIBAQCelMmMDCGBhqn/a3VrNAoKMr1HVLKxGoG7VF/13HZJ
0twObUZlh3Jz+XeD+kNAURhELWTrsgdTkQQfqinqOuRemxTl55+x7nLpe5nmwaBH
XqqDOHubmkbAGanGcm6T/rD9KNk1Z46Uc2p7UYu0fwNO0mo0aqFL2FSyvzZwziNe
g7ELYZ4a3LvGn81JfP/JvM6pgtoMNuee5RV6TWaz7LV304ICj8Bhphy/HFpOA1rb
O9gs8CUMgqz+RroAIa8cV8gbF/fPCz9Ofl7Gdmib679JxxFrW4wRJ0nMJgJmsZXq
jaVc0g7ORc+eIAcHw7Uroc6h7Y7lGjOkDZF75j0mLQa3AgMBAAGjggGEMIIBgDAd
BgNVHQ4EFgQU3hNEuwvUGNCHY1TBatcUR03pNdYwHwYDVR0jBBgwFoAU3hNEuwvU
GNCHY1TBatcUR03pNdYwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMCAQYw
GAYDVR0gAQH/BA4wDDAKBggrBgEFBQcOAjCBuQYIKwYBBQUHAQsEgawwgakwPgYI
KwYBBQUHMAqGMnJzeW5jOi8vcnBraS5leGFtcGxlLm5ldC9yZXBvc2l0b3J5L2V4
YW1wbGUtdGEubWZ0MDUGCCsGAQUFBzANhilodHRwczovL3JyZHAuZXhhbXBsZS5u
ZXQvbm90aWZpY2F0aW9uLnhtbDAwBggrBgEFBQcwBYYkcnN5bmM6Ly9ycGtpLmV4
YW1wbGUubmV0L3JlcG9zaXRvcnkvMCcGCCsGAQUFBwEHAQH/BBgwFjAJBAIAATAD
AwEAMAkEAgACMAMDAQAwHgYIKwYBBQUHAQgEEjAQoA4wDDAKAgEAAgUA/////zAN
BgkqhkiG9w0BAQsFAAOCAQEAgZFQ0Sf3CI5Hwev61AUWHYOFniy69PuDTq+WnhDe
xX5rpjSDRrs5L756KSKJcaOJ36lzO45lfOPSY9fH6x30pnipaqRA7t5rApky24jH
cSUA9iRednzxhVyGjWKnfAKyNo2MYfaOAT0db1GjyLKbOADI9FowtHBUu+60ykcM
Quz66XrzxtmxlrRcAnbv/HtV17qOd4my6q5yjTPR1dmYN9oR/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++gAwIBAgIUcyCzS10hdfG65kbRq7toQAvRDKowDQYJKoZIhvcNAQEL
BQAwFTETMBEGA1UEAxMKZXhhbXBsZS10YTAeFw0yMDA5MDMxOTAyMTlaFw0yMTA5
MDMxOTAyMTlaMDMxMTAvBgNVBAMTKDNBQ0UyQ0VGNEZCMjFCN0QxMUUzRTE4NEVG
QzFFMjk3QjM3Nzg2NDIwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDc
zz1qwTxC2ocw5rqp8ktm2XyYkl8riBVuqlXwfefTxsR2YFpgz9vkYUd5Az9EVEG7
6wGIyZbtmhK63eEeaqbKz2GHub467498BXeVrYysO+YuIGgCEYKznNDZ4j5aaDbo
j5+4/z0Qvv6HEsxQd0f8br6lKJwgeRM6+fm7796HNPB0aqD7Zj9NRCLXjbB0DCgJ
liH6rXMKR86ofgll9V2mRjesvhdKYgkGbOif9rvxVpLJ/6zdru5CE9yeuJZ59l+n
YH/r6PzdJ4Q7yKrJX8qD6A60j4+biaU4MQ72KpsjhQNTTqF/HRwi0N54GDaknEwE
TnJQHgLJDYqww9yKWtjjAgMBAAGjggIvMIICKzAdBgNVHQ4EFgQUOs4s70+yG30R
4+GE78Hil7N3hkIwHwYDVR0jBBgwFoAU3hNEuwvUGNCHY1TBatcUR03pNdYwDwYD
VR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMCAQYwGAYDVR0gAQH/BA4wDDAKBggr
BgEFBQcOAjBhBgNVHR8EWjBYMFagVKBShlByc3luYzovL3Jwa2kuZXhhbXBsZS5u
ZXQvcmVwb3NpdG9yeS8zQUNFMkNFRjRGQjIxQjdEMTFFM0UxODRFRkMxRTI5N0Iz
Nzc4NjQyLmNybDBOBggrBgEFBQcBAQRCMEAwPgYIKwYBBQUHMAKGMnJzeW5jOi8v
cnBraS5leGFtcGxlLm5ldC9yZXBvc2l0b3J5L2V4YW1wbGUtdGEuY2VyMIG5Bggr
BgEFBQcBCwSBrDCBqTA+BggrBgEFBQcwCoYycnN5bmM6Ly9ycGtpLmV4YW1wbGUu
bmV0L3JlcG9zaXRvcnkvZXhhbXBsZS1jYS5tZnQwNQYIKwYBBQUHMA2GKWh0dHBz
Oi8vcnJkcC5leGFtcGxlLm5ldC9ub3RpZmljYXRpb24ueG1sMDAGCCsGAQUFBzAF
hiRyc3luYzovL3Jwa2kuZXhhbXBsZS5uZXQvcmVwb3NpdG9yeS8wHwYIKwYBBQUH
AQcBAf8EEDAOMAwEAgABMAYDBADAAAIwHgYIKwYBBQUHAQgEEjAQoA4wDDAKAgMA
+/ACAwD78TANBgkqhkiG9w0BAQsFAAOCAQEAnLu+d1ZsUTiX3YWGueTHIalW4ad0
Kupi7pYMV2nXbxNGmdJMol9BkzVz9tj55ReMghUU4YLm/ICYe4fz5e0T8o9s/vIm
cGS29+WoGuiznMitpvbS/379gaMezk6KpqjH6Brw6meMqy09phmcmvm3x3WTmx09
mLlQneMptwk8qSYcnMUmGLJs+cVqmkOa3sWRdw8WrGu6QqYtQz3HFZQojF06YzEq
V/dBdCFdEOwTfVl2n2XqhoJl/oEBdC4uu2G0qRk3+WVs+uwVHP0Ttsbt7TzFgZfY
yxqvOg6QoldxZVZmHHncKmETu/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-----
MIIEpTCCA42gAwIBAgIUJ605QIPX8rW5m4Zwx3WyuW7hZuQwDQYJKoZIhvcNAQEL
BQAwMzExMC8GA1UEAxMoM0FDRTJDRUY0RkIyMUI3RDExRTNFMTg0RUZDMUUyOTdC
Mzc3ODY0MjAeFw0yMTA1MjAxNjA1NDVaFw0yMjAzMTYxNjA1NDVaMDMxMTAvBgNV
BAMTKDkxNDY1MkEzQkQ1MUMxNDQyNjAxOTg4ODlGNUM0NUFCRjA1M0ExODcwggEi
MA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCycTQrOb/qB2W3i3Ki8PhA/DEW
yii2TgGo9pgCwO9lsIRI6Zb/k+aSiWWP9kSczlcQgtPCVwr62hTQZCIowBN0BL0c
K0/5k1imJdi5qdM3nvKswM8CnoR11vB8pQFwruZmr5xphXRvE+mzuJVLgu2V1upm
BXuWloeymudh6WWJ+GDjwPXO3RiXBejBrOFNXhaFLe08y4DPfr/S/tXJOBm7QzQp
tmbPLYtGfprYu45liFFqqP94UeLpISfXd36AKGzqTFCcc3EW9l5UFE1MFLlnoEog
qtoLoKABt0IkOFGKeC/EgeaBdWLe469ddC9rQft5w6g6cmxG+aYDdIEB34zrAgMB
AAGjggGvMIIBqzAdBgNVHQ4EFgQUkUZSo71RwUQmAZiIn1xFq/BToYcwHwYDVR0j
BBgwFoAUOs4s70+yG30R4+GE78Hil7N3hkIwDAYDVR0TAQH/BAIwADAOBgNVHQ8B
Af8EBAMCB4AwGAYDVR0gAQH/BA4wDDAKBggrBgEFBQcOAjBhBgNVHR8EWjBYMFag
VKBShlByc3luYzovL3Jwa2kuZXhhbXBsZS5uZXQvcmVwb3NpdG9yeS8zQUNFMkNF
RjRGQjIxQjdEMTFFM0UxODRFRkMxRTI5N0IzNzc4NjQyLmNybDBsBggrBgEFBQcB
AQRgMF4wXAYIKwYBBQUHMAKGUHJzeW5jOi8vcnBraS5leGFtcGxlLm5ldC9yZXBv
c2l0b3J5LzNBQ0UyQ0VGNEZCMjFCN0QxMUUzRTE4NEVGQzFFMjk3QjM3Nzg2NDIu
Y2VyMBkGCCsGAQUFBwEHAQH/BAowCDAGBAIAAQUAMEUGCCsGAQUFBwELBDkwNzA1
BggrBgEFBQcwDYYpaHR0cHM6Ly9ycmRwLmV4YW1wbGUubmV0L25vdGlmaWNhdGlv
bi54bWwwDQYJKoZIhvcNAQELBQADggEBAEjC98gVp0Mb7uiKaHylP0453mtJ+AkN
07fsK/qGw/e90DJv7cp1hvjj4uy3sgf7PJQ7cKNGrgybq/lE0jce+ARgVjbi2Brz
ZsWAnB846Snwsktw6cenaif6Aww6q00NspAepMBd2Vg/9sKFvOwJFVOgNcqiQiXP
5rGJPWBcOMv52a/7adjfXwpnOijiTOgMloQGmC2TPZpydZKjlxEATdFEQssa33xD
nlpp+/r9xuNVYRtRcC36oWraVA3jzN6F6rDE8r8xs3ylISVz6JeCQ4YRYwbMsjjc
/tiJLM7ZYxIe5IrYz1ZtN6n/SEssJAswRIgps2EhCt/HS2xAmGCOhgU=
-----END CERTIFICATE-----
```

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

```
   0 1189: SEQUENCE {
   4  909:  SEQUENCE {
   8    3:   [0] {
  10    1:    INTEGER 2
         :     }
  13   20:   INTEGER 27AD394083D7F2B5B99B8670C775B2B96EE166E4
  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 20/05/2021 16:05:45 GMT
 120   13:    UTCTime 16/03/2022 16:05:45 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  431:   [3] {
 486  427:    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   25:     SEQUENCE {
821    8:      OBJECT IDENTIFIER ipAddrBlocks (1 3 6 1 5 5 7 1 7)
831    1:      BOOLEAN TRUE
834   10:      OCTET STRING, encapsulates {
836    8:       SEQUENCE {
838    6:        SEQUENCE {
840    2:         OCTET STRING 00 01
844    0:         NULL
        :          }
        :         }
        :        }
        :       }
846   69:     SEQUENCE {
848    8:      OBJECT IDENTIFIER subjectInfoAccess
        :       (1 3 6 1 5 5 7 1 11)
858   57:      OCTET STRING, encapsulates {
860   55:       SEQUENCE {
862   53:        SEQUENCE {
864    8:         OBJECT IDENTIFIER '1 3 6 1 5 5 7 48 13'
874   41:         [6]
        :          'https://rrdp.example.net/notification.xml'
        :          }
        :         }
        :        }
        :       }
        :      }
        :     }
        :    }
917   13:  SEQUENCE {
919    9:   OBJECT IDENTIFIER sha256WithRSAEncryption
        :    (1 2 840 113549 1 1 11)
930    0:   NULL
        :    }
```

```
 932  257:  BIT STRING
          :   48 C2 F7 C8 15 A7 43 1B EE E8 8A 68 7C A5 3F 4E
          :   39 DE 6B 49 F8 09 0D D3 B7 EC 2B FA 86 C3 F7 BD
          :   D0 32 6F ED CA 75 86 F8 E3 E2 EC B7 B2 07 FB 3C
          :   94 3B 70 A3 46 AE 0C 9B AB F9 44 D2 37 1E F8 04
          :   60 56 36 E2 D8 1A F3 66 C5 80 9C 1F 38 E9 29 F0
          :   B2 4B 70 E9 C7 A7 6A 27 FA 03 0C 3A AB 4D 0D B2
          :   90 1E A4 C0 5D D9 58 3F F6 C2 85 BC EC 09 15 53
          :   A0 35 CA A2 42 25 CF E6 B1 89 3D 60 5C 38 CB F9
          :   D9 AF FB 69 D8 DF 5F 0A 67 3A 28 E2 4C E8 0C 96
          :   84 06 98 2D 93 3D 9A 72 75 92 A3 97 11 00 4D D1
          :   44 42 CB 1A DF 7C 43 9E 5A 69 FB FA FD C6 E3 55
          :   61 1B 51 70 2D FA A1 6A DA 54 0D E3 CC DE 85 EA
          :   B0 C4 F2 BF 31 B3 7C A5 21 25 73 E8 97 82 43 86
          :   11 63 06 CC B2 38 DC FE D8 89 2C CE D9 63 12 1E
          :   E4 8A D8 CF 56 6D 37 A9 FF 48 4B 2C 24 0B 30 44
          :   88 29 B3 61 21 0A DF C7 4B 6C 40 98 60 8E 86 05
          :   }
```

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/6gdlt4tyovD4QPwxFsootk4BqPaYAsDvZbCESOmW
/5Pmkollj/ZEnM5XEILTwlcK+toU0GQiKMATdAS9HCtP+ZNYpiXYuanTN57yrMDP
Ap6EddbwfKUBcK7mZq+caYV0bxPps7iVS4LtldbqZgV7lpaHsprnYellifhg48D1
zt0YlwXowazhTV4WhS3tPMuAz36/0v7VyTgZu0M0KbZmzy2LRn6a2LuOZYhRaqj/
eFHi6SEn13d+gChs6kxQnHNxFvZeVBRNTBS5Z6BKIKraC6CgAbdCJDhRingvxIHm
gXVi3uOvXXQva0H7ecOoOnJsRvmmA3SBAd+M6wIDAQABAoIBAQCyB0FeMuKm8bRo
18aKjFGSPEoZi53srIz5bvUgIi92TBLez7ZnzL6Iym26oJ+5th+lCHGO/dqlhXio
pI50C5Yc9TFbblb/ECOsuCuuqKFjZ8CD3GVsHozXKJeMM+/o5YZXQrORj6UnwT0z
ol/JE5pIGUCIgsXX6tz9s5BP3lUAvVQHsv6+vEVKLxQ3wj/1vIL8O/CN036EV0GJ
mpkwmygPjfECT9wbWo0yn3jxJb36+M/QjjUP28oNIVn/IKoPZRXnqchEbuuCJ651
IsaFSqtiThm4WZtvCH/IDq+6/dcMucmTjIRcYwW7fdHfjplllVPve9c/OmpWEQvF
t3ArWUt5AoGBANs4764yHxo4mctLIE7G7l/tf9bP4KKUiYw4R4ByEocuqMC4yhmt
MPCfOFLOQet71OWCkjP2L/7EKUe9yx7G5KmxAHY6jOjvcRkvGsl6lWFOsQ8p126M
Y9hmGzMOjtsdhAiMmOWKzjvm4WqfMgghQe+PnjjSVkgTt+7BxpIuGBAvAoGBANBg
26FF5cDLpixOd3Za1YXsOgguwCaw3Plvi7vUZRpa/zBMELEtyOebfakkIRWNm07l
nE+lAZwxm+29PTD0nqCFE91teyzjnQaLO5kkAdJiFuVV3icLOGo399FrnJbKensm
FGSli+3KxQhCNIJJfgWzq4bE0ioAMjdGbYXzIYQFAoGBAM6tuDJ36KDU+hIS6wu6
O2TPSfZhF/zPo3pCWQ78/QDb+Zdw4IEiqoBA7F4NPVLg9Y/H8UTx9r/veqe7hPOo
Ok7NpIzSmKTHkc5XfZ60Zn9OLFoKbaQ40a1kXoJdWEu2YROaUlAe9F6/Rog6PHYz
vLE5qscRbu0XQhLkN+z7bg5bAoGBAKDsbDEb/dbqbyaAYpmwhH2sdRSkphg7Niwc
DNm9qWa1J6Zw1+M87I6Q8naRREuU1IAVqqWHVLr/ROBQ6NTJ1Uc5/qFeT2XXUgkf
taMKv61tuyjZK3sTmznMh0HfzUpWjEhWnCEuB+ZYVdmO52ZGw2A75RdrILL2+9Dc
PvDXVubRAoGAdqXeSWoLxuzZXzl8rsaKrQsTYaXnOWaZieU1SL5vVe8nK257UDqZ
E3ng2j5XPTUWli+aNGFEJGRoNtcQvO60O/sFZUhu52sqq9mWVYZNh1TB5aP8X+pV
iFcZOLUvQEcN6PA+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 - 192.0.2.255
# MIIGjwYJKoZIhvcNAQcCoIIGgDCCBnwCAQMxDTALBglghkgBZQMEAgEwDQYLKoZ
# IhvcNAQkQAS+gggSpMIIEpTCCA42gAwIBAgIUJ605QIPX8rW5m4Zwx3WyuW7hZu
# QwDQYJKoZIhvcNAQELBQAwMzExMC8GA1UEAxMoM0FDRTJDRUY0RkIyMUI3RDExR
# TNFMTg0RUZDMUUyOTdCMzc3ODY0MjAeFw0yMTA1MjAxNjA1NDVaFw0yMjAzMTYx
# NjA1NDVaMDMxMTAvBgNVBAMTKDkxNDY1MkEzQkQ1MUMxNDQyNjAxOTg4ODlGNUM
# 0NUFCRjA1M0ExODcwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCycT
# QrOb/qB2W3i3Ki8PhA/DEWyii2TgGo9pgCwO9lsIRI6Zb/k+aSiWWP9kSczlcQg
# tPCVwr62hTQZCIowBN0BL0cK0/5k1imJdi5qdM3nvKswM8CnoR11vB8pQFwruZm
# r5xphXRvE+mzuJVLgu2V1upmBXuWloeymudh6WWJ+GDjwPXO3RiXBejBrOFNXha
# FLe08y4DPfr/S/tXJOBm7QzQptmbPLYtGfprYu45liFFqqP94UeLpISfXd36AKG
# zqTFCcc3EW9l5UFE1MFLlnoEogqtoLoKABt0IkOFGKeC/EgeaBdWLe469ddC9rQ
# ft5w6g6cmxG+aYDdIEB34zrAgMBAAGjggGvMIIBqzAdBgNVHQ4EFgQUkUZSo71R
# wUQmAZiIn1xFq/BToYcwHwYDVR0jBBgwFoAUOs4s70+yG30R4+GE78Hil7N3hkI
# wDAYDVR0TAQH/BAIwADAOBgNVHQ8BAf8EBAMCB4AwGAYDVR0gAQH/BA4wDDAKBg
# grBgEFBQcOAjBhBgNVHR8EWjBYMFagVKBShlByc3luYzovL3Jwa2kuZXhhbXBsZ
# S5uZXQvcmVwb3NpdG9yeS8zQUNFMkNFRjRGQjIxQjdEMTFFM0UxODRFRkMxRTI5
# N0IzNzc4NjQyLmNybDBsBggrBgEFBQcBAQRgMF4wXAYIKwYBBQUHMAKGUHJzeW5
# jOi8vcnBraS5leGFtcGxlLm5ldC9yZXBvc2l0b3J5LzNBQ0UyQ0VGNEZCMjFCN0
# QxMUUzRTE4NEVGQzFFMjk3QjM3Nzg2NDIuY2VyMBkGCCsGAQUFBwEHAQH/BAowC
# DAGBAIAAQUAMEUGCCsGAQUFBwELBDkwNzA1BggrBgEFBQcwDYYpaHR0cHM6Ly9y
# cmRwLmV4YW1wbGUubmV0L25vdGlmaWNhdGlvbi54bWwwDQYJKoZIhvcNAQELBQA
# DggEBAEjC98gVp0Mb7uiKaHylP0453mtJ+AkN07fsK/qGw/e90DJv7cp1hvjj4u
# y3sgf7PJQ7cKNGrgybq/lE0jce+ARgVjbi2BrzZsWAnB846Snwsktw6cenaif6A
# ww6q00NspAepMBd2Vg/9sKFvOwJFVOgNcqiQiXP5rGJPWBcOMv52a/7adjfXwpn
# OijiTOgMloQGmC2TPZpydZKjlxEATdFEQssa33xDnlpp+/r9xuNVYRtRcC36oWr
# aVA3jzN6F6rDE8r8xs3ylISVz6JeCQ4YRYwbMsjjc/tiJLM7ZYxIe5IrYz1ZtN6
# n/SEssJAswRIgps2EhCt/HS2xAmGCOhgUxggGqMIIBpgIBA4AUkUZSo71RwUQmA
# ZiIn1xFq/BToYcwCwYJYIZIAWUDBAIBoGswGgYJKoZIhvcNAQkDMQ0GCyqGSIb3
# DQEJEAEvMBwGCSqGSIb3DQEJBTEPFw0yMTA1MjAxNjI4MzlaMC8GCSqGSIb3DQE
# JBDEiBCAr4vKeUvHJINsE0YQwUMxoo48qrOU+iPuFbQR8qX3BFjANBgkqhkiG9w
# 0BAQEFAASCAQB85HsCBrU3EcVOcf4nC6Z3jrOjT+fVlyTDAObF6GTNWgrxe7jSA
# Inyf51UzuIGqhVY3sQiiXbdWcVYtPb4118KvyeXh8A/HLp4eeAJntl9D3igt38M
# o84q5pf9pTQXx3hbsm51ilpOip/TKVMqzE42s6OPox3M0+6eKH3/vBKnw1s1ayM
# 0MUnPDTBfZL3JJEGPWfIZHEcrypevbqR7Jjsz5vp0qyF2D9v+w+nyhZOPmuePm7
# YqLyOw/E99PVBs9uI+hmBiCz/BK2Z3VRjrrlrUU+49eldSTkZ2sJyhCbbV2Ufgi
# S2FOquAgJzjilyN3BDQLV8Rp9cGh0PpVslKH2na
# End Signature: 192.0.2.0 - 192.0.2.255
```


# Acknowledgments

Thanks to Rob Austein for CMS and detached signature clue, George Michaelson for the first and substantial external review, and Erik Kline who was too shy to agree to coauthorship. Additionally, we express our gratitude to early implementors, including Menno Schepers; Flavio Luciani; Eric Dugas; Job Snijders, who provided running code; and Kevin Pack. Also, thanks to the following geolocation providers who are consuming geofeeds with this described solution: Jonathan Kosgei (ipdata.co), Ben Dowling (ipinfo.io), and Pol Nisenblat (bigdatacloud.com). For an amazing number of helpful reviews, we thank Adrian Farrel, Antonio Prado, Francesca

Palombini, Jean-Michel Combes (INTDIR), John Scudder, Kyle Rose (SECDIR), Martin Duke, Murray Kucherawy, Paul Kyzivat (GENART), Rob Wilton, and Roman Danyliw. The authors also thank George Michaelson, the awesome document shepherd.

## Authors' Addresses

**Randy Bush**
IIJ & Arrcus
5147 Crystal Springs
Bainbridge Island, Washington 98110
United States of America
Email: randy@psg.com

**Massimo Candela**
NTT
Siriusdreef 70-72
2132 WT Hoofddorp
Netherlands
Email: massimo@ntt.net

**Warren Kumari**
Google
1600 Amphitheatre Parkway
Mountain View, CA 94043
United States of America
Email: warren@kumari.net

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