Internet Engineering Task Force (IETF)
Request for Comments: 9632
Obsoletes: 9092
Category: Standards Track
ISSN: 2070-1721

R. Bush
IIJ Research & Arrcus
M. Candela
NTT
W. Kumari
Google
R. Housley
Vigil Security
August 2024

Finding and Using Geofeed Data

Abstract

This document specifies how to augment the Routing Policy Specification Language (RPSL) inetnum: class to refer specifically to geofeed comma-separated values (CSV) data files and describes an optional scheme that uses the Resource Public Key Infrastructure (RPKI) to authenticate the geofeed data files. This document obsoletes RFC 9092.

Status of This Memo

Copyright Notice

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, which may not point to a user; see Section 14 of [RFC6269] in particular. Also, administrators of infrastructure and other services might wish to publish the locale of said infrastructure or services. 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 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 in Section 5.

This document obsoletes [RFC9092]. Changes from [RFC9092] include the following:

* RIPE has implemented the geofeed: attribute.

* This document allows, but discourages, an inetnum: to have both a geofeed remarks: attribute and a geofeed: attribute.

* The Authentication section (Section 5) has been rewritten to be more formal.

* Geofeed files are only UTF-8 CSV.

* This document stresses that authenticating geofeed data is optional.

* IP Address Delegation extensions must not use "inherit".

* If geofeed data are present, geographic location hints in other data should be ignored.

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.

Per [RFC8805], geofeed files consist of comma-separated values (CSV) in UTF-8 text format, not HTML, richtext, or other formats.

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 7); 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]. At the time of publishing this document, change control of the RPSL effectively lies in the operator community.

The inetnum: database class is specified by the RPSL, as well as Routing Policy System Security [RFC2725] and RPSLng [RFC4012], which are used by the Regional Internet Registries (RIRs). Each of these objects describes an IP address range and its attributes. The inetnum: objects form a hierarchy ordered on the address space.

Ideally, the RPSL would be augmented to define a new RPSL geofeed: attribute in the inetnum: class. Absent implementation of the geofeed: attribute in a particular RIR database, 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
```

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
```

The URL uses HTTPS, so the WebPKI provides authentication, integrity, and confidentiality for the fetched geofeed file. However, the WebPKI cannot 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 5.

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 SHOULD have, at most, one geofeed reference, whether a remarks: or a proper geofeed: attribute when it is implemented. As the remarks: form cannot be formally checked by the RIR, this cannot be formally enforced. A geofeed: attribute is preferred, of course, if the RIR supports it. If there is more than one type of attribute in the intetnum: object, the geofeed: attribute MUST be used.

For inetnum: objects covering the same address range, a signed geofeed file MUST be preferred over an unsigned file. If none are signed, or more than one is signed, the (signed) inetnum: with the most recent last-modified: attribute MUST be preferred.

If a geofeed 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 5.

An unsigned, and only an unsigned, geofeed file MAY be referenced by multiple inetnum: objects and MAY contain prefixes from more than one registry.

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.

4. Fetching Geofeed Data

This document provides a guideline for how interested parties should fetch and read geofeed files.

Historically, before [RFC9092], this was done in varied ways, at the discretion of the implementor, often without consistent authentication, where data were mostly imported from email without formal authorization or validation.

To minimize the load on RIRs' WHOIS [RFC3912] services, the RIR's FTP [RFC0959] services SHOULD be used for large-scale access to gather inetnum: objects with geofeed references. This uses efficient bulk access instead of fetching via brute-force search through the IP space.

When reading data from an unsigned geofeed file, one MUST ignore data outside the referring inetnum: object's address range. This is to avoid importing data about ranges not under the control of the operator. Note that signed files MUST only contain prefixes within the referring inetnum:'s range as mandated in Section 5.

If geofeed files are fetched, other location information from the inetnum: MUST be ignored.

Given an address range of interest, the most specific inetnum: object with a geofeed reference MUST be used to fetch the geofeed file. For example, if the fetching party finds the following inetnum: objects:

```
inetnum: 192.0.0.0/22 # example
remarks: Geofeed https://example.com/geofeed_1

inetnum: 192.0.2.0/24 # example
remarks: Geofeed https://example.com/geofeed_2
```

An application looking for geofeed data for 192.0.2.0/29 MUST ignore data in geofeed_1 because 192.0.2.0/29 is within the more specific 192.0.2.0/24 inetnum: covering that address range and that inetnum: does have a geofeed reference.

Hints in inetnum: objects such as country:, geoloc:, etc., tend to be administrative, and not deployment specific. Consider large, possibly global, providers with headquarters very far from most of their deployments. Therefore, if geofeed data are specified, either as a geofeed: attribute or in a geofeed remarks: attribute, other geographic hints such as country:, geoloc:, DNS geoloc RRsets, etc., for that address range MUST be ignored.

There is open-source code to traverse the RPSL data across all of the RIRs, collect all geofeed references, and process them [GEOFEED-FINDER]. It implements the steps above and of all the Operational Considerations described in Section 6, including caching. It produces a single geofeed file, merging all the geofeed files found. This open-source code can be run daily by a cron job, and the output file can be directly used.

RIRs are converging on Registration Data Access Protocol (RDAP) support, which includes geofeed data; see [RDAP-GEOFEED]. This SHOULD NOT be used for bulk retrieval of geofeed data.

5. Authenticating Geofeed Data (Optional)

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 some repositories is weakly authenticated at best. An approach where the 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.

The remainder of this section specifies an optional authenticator for the geofeed data set that follows "Signed Object Template for the Resource Public Key Infrastructure (RPKI)" [RFC6488].

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. The following format bundles the relevant RPKI certificate with a signature over 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 each 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 covered by the digital signature.

If the authenticator is not in the canonical form described above, then the authenticator is invalid.

Borrowing detached signatures from [RFC5485], after file canonicalization, the Cryptographic Message Syntax (CMS) [RFC5652] is used to create a detached DER-encoded signature that is then Base64 encoded with padding (as defined in Section 4 of [RFC4648]) and line wrapped to 72 or fewer characters. The same digest algorithm MUST be used for calculating the message digest of the 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 CMS SignedData DigestAlgorithmIdentifiers and the SignerInfo DigestAlgorithmIdentifier [RFC5652]. The RPKI certificate covering the geofeed inetnum: object’s address range is included in the CMS SignedData certificates field [RFC5652].

The address range of the signing certificate MUST cover all prefixes in the signed geofeed file. If not, the authenticator is invalid.

The signing certificate MUST NOT include the Autonomous System Identifier Delegation certificate extension [RFC3779]. If it is present, the authenticator is invalid.

As with many other RPKI signed objects, the IP Address Delegation certificate extension MUST NOT use the "inherit" capability defined in Section 2.2.3.5 of [RFC3779]. If "inherit" is used, the authenticator is invalid.

An IP Address Delegation extension using "inherit" would complicate processing. The implementation would have to build the certification path from the end entity to the trust anchor, then validate the path from the trust anchor to the end entity, and then the parameter would have to be remembered when the validated public key was used to validate a signature on a CMS object. Having to remember things from certification path validation for use with CMS object processing would be quite complex and error-prone. Additionally, the certificates do not get that much bigger by repeating the information.

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 lines in a geofeed file do align.

The Certification Authority (CA) SHOULD sign only one geofeed file with each generated private key and SHOULD generate a new key pair for each new version of a particular geofeed file. The CA MUST generate a new end entity (EE) certificate for each signing of a particular geofeed file. An associated EE certificate used in this fashion is termed a "one-time-use" EE certificate (see Section 3 of [RFC6487]).

Identifying the private key associated with the certificate and getting the department that controls the private key (which might be stored in a Hardware Security Module (HSM)) to generate the CMS signature is left as an exercise for the implementor. On the other hand, verifying the signature has no similar complexity; the certificate, which is validated in the public RPKI, contains the needed public key. The RPKI trust anchors for the RIRs are expected to already be available to the party performing signature validation. Validation of the CMS signature over 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.

2. Validating the signer’s certificate MUST ensure that it is part of the current [RFC9286] manifest and that all resources are covered by the RPKI certificate.

3. 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.

4. 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.

5. Confirming that the eContentType object identifier (OID) is id-ct-geofeedCSVwithCRLF (1.2.840.113549.1.9.16.1.47). This OID MUST appear within both the eContentType in the encapContentInfo object and within the ContentType signed attribute in the signerInfo object (see [RFC6488]).

6. 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 the above steps MUST be successful to consider the geofeed file signature as valid.

The authenticator MUST be hidden as a series of "#" comments at the end of the geofeed file. The following simple example is cryptographically incorrect:

```
# 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
```

A correct and full example is in Appendix A.

The CMS signature does not cover the signature lines.

The bracketing "# RPKI Signature:" and "# End Signature:" MUST be present as shown in the example. The RPKI Signature’s IP address range MUST match that of the geofeed URL in the inetnum: that points to the geofeed file.

6. 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 [RFC9110].

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 5, 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.

Harvesting and publishing aggregated geofeed data outside of the RPSL model should be avoided as it could lead to detailed data of one aggregatee undesirably affecting the less detailed data of a different aggregatee. Moreover, publishing aggregated geofeed data prevents the reader of the data from performing the checks described in Sections 4 and 5.

At the time of publishing this document, 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 [RFC9111] 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.

7. 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.

8. Implementation Status

At the time of publishing this document, the geofeed: attribute in inetnum objects has been implemented in the RIPE and APNIC databases.

Registrants in databases that do not yet support the geofeed: attribute are using the remarks: attribute, or equivalent.

At the time of publishing this document, 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 RDAP [RFC9082], etc., the "NetRange" attribute/key must be treated as "inetnum", and the "Comment" attribute must be treated as "remarks".

[rpki-client] can be used to authenticate a signed geofeed file.

9. 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.

The consumer of geofeed data SHOULD fetch and process the data themselves. Importing data sets produced and/or processed by a third-party places significant trust in the third-party.

As mentioned in Section 5, some RPSL repositories have weak, if any, authentication. This allows spoofing of inetnum: objects pointing to malicious geofeed files. Section 5 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.

10. IANA Considerations

In the SMI Security for S/MIME CMS Content Type (1.2.840.113549.1.9.16.1) in the Structure of Management Information (SMI) Numbers (MIB Module Registrations) registry group (located at <https://www.iana.org/assignments/smi-numbers/>), the reference for this registration has been updated to this document:

Decimal	Description	Reference
47	id-ct-geofeedCSVwithCRLF	RFC 9632

Table 1: From SMI Security for S/MIME Module Identifier (1.2.840.113549.1.9.16.1)

11. References

11.1. 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>.

[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>.

[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>.

[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>.

[RFC9110] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Semantics", STD 97, RFC 9110, DOI 10.17487/RFC9110, June 2022, <https://www.rfc-editor.org/info/rfc9110>.

[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>.

11.2. Informative References

[GEOFEED-FINDER] "geofeed-finder", commit 5f557a4, March 2024, <https://github.com/massimocandela/geofeed-finder>.

[INET6NUM] RIPE NCC, "RIPE Database Documentation: Description of the INET6NUM Object", <https://apps.db.ripe.net/docs/RPSL-Object-Types/Descriptions-of-Primary-Objects/#description-of-the-inet6num-object>.

[INETNUM] RIPE NCC, "RIPE Database Documentation: Description of the INETNUM Object", <https://apps.db.ripe.net/docs/RPSL-Object-Types/Descriptions-of-Primary-Objects/#description-of-the-inetnum-object>.

[RDAP-GEOFEED] Singh, J. and T. Harrison, "An RDAP Extension for Geofeed Data", Work in Progress, Internet-Draft, draft-ietf-regext-rdap-geofeed-07, 6 August 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-regext-rdap-geofeed-07>.

[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>.

[RFC6269] Ford, M., Ed., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing", RFC 6269, DOI 10.17487/RFC6269, June 2011, <https://www.rfc-editor.org/info/rfc6269>.

[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>.

[RFC9092] Bush, R., Candela, M., Kumari, W., and R. Housley, "Finding and Using Geofeed Data", RFC 9092, DOI 10.17487/RFC9092, July 2021, <https://www.rfc-editor.org/info/rfc9092>.

[RFC9111] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Caching", STD 98, RFC 9111, DOI 10.17487/RFC9111, June 2022, <https://www.rfc-editor.org/info/rfc9111>.

[RIPE-DB] RIPE NCC, "RIPE Database Documentation", September 2023, <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-client] Snijders, J., "Example on how to use rpki-client to authenticate a signed Geofeed", September 2023, <https://sobornost.net/~job/using_geofeed_authenticators.txt>.

Appendix A. Example

This appendix provides an example, including a trust anchor, a Certificate Revocation List (CRL) signed by the trust anchor, a CA certificate subordinate to the trust anchor, a CRL signed by the CA, 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-----
MIIEQTCCAymgAwIBAgIUEggycNoFVRjAuN/Fw7URu0DEZNAwDQYJKoZIhvcNAQEL
BQAwFTETMBEGA1UEAxMKZXhhbXBsZS10YTAeFw0yMzA5MTkyMDMzMzlaFw0zMzA5
MTYyMDMzMzlaMBUxEzARBgNVBAMTCmV4YW1wbGUtdGEwggEiMA0GCSqGSIb3DQEB
AQUAA4IBDwAwggEKAoIBAQDQprR+g/i4JyObVURTp1JpGM23vGPyE5fDKFPqV7rw
M1Amm7cnew66U02IzV0X5oiv5nSGfRX5UxsbR+vwPBMceQyDgS5lexFiv4fB/Vjf
DT2qX/UjsLL9QOeaSOh7ToJSLjmtpa0D9iz7ful3hdxRjpMMZiE/reX9/ymdpW/E
dg0F6+T9WGZE1miPeIjl5OZwnmLHCftkN/aaYk1iPNjNniHYIOjC1jSpABmoZyTj
sgrwLE2F1fIRkVkwASqToq/D5v9voXaYYaXUNJb4H/5wenRuvT5O/n6PXh70rMQy
F5yzLs96ytxqg5gGX9kabVnvxFU8nHfPa0rhlwfTJnljAgMBAAGjggGHMIIBgzAd
BgNVHQ4EFgQUwL1SXb7SeLIW7LOjQ5XSBguZCDIwHwYDVR0jBBgwFoAUwL1SXb7S
eLIW7LOjQ5XSBguZCDIwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMCAQYw
```

GAYDVR0gAQH/BA4wDDAKBggrBgEFBQcOAjCBuQYIKwYBBQUHAQsEgawwgakwPgYI
KwYBBQUHMAqGMnJzeW5jOi8vcnBraS5leGFtcGxlLm5ldC9yZXBvc2l0b3J5L2V4
YW1wbGUtdGEubWZ0MDUGCCsGAQUFBzANhilodHRwczovL3JyZHAuZXhhbXBsZS5u
ZXQvbm90aWZpY2F0aW9uLnhtbDAwBggrBgEFBQcwBYYkcnN5bmM6Ly9ycGtpLmV4
YW1wbGUubmV0L3JlcG9zaXRvcnkvMCcGCCsGAQUFBwEHAQH/BBgwFjAJBAIAATAD
AwEAMAkEAgACMAMDAQAwIQYIKwYBBQUHAQgBAf8EEjAQoA4wDDAKAgEAAgUA////
/zANBgkqhkiG9w0BAQsFAAOCAQEAa9eLY9QAmnlZOIyOzbpta5wqcOUQV/yR7o/0
1zkEZaSavKBt19lMK6AXZurx1T5jyjIwG7bEtZZThjtH2m80V5kc2tsFjSq/yp7N
JBclMHVd3tXse9If3nXYF4bxRIcir1lXlAbYN+Eo1U3i5qJO+fxouzt7Merk2Dih
nsenTeXKzN7tfmuCYZZHCC8viCoJWdH+o1uRM4TiQApZsUJ8sF4TABrrRJmA/Ed5
v0CTBbgqTx7yg0+VarFLPdnjYgtpoCJqwE2C1UpX15rZSaLVuGXtbwXd/cHEg5vF
W6QTsMeMQFEUa6hkicDGtxLTUdhckBgmCGoF2nlZii5f1BTWAg==
-----END CERTIFICATE-----

The CRL is issued by the trust anchor.

-----BEGIN X509 CRL-----
MIIBjjB4AgEBMA0GCSqGSIb3DQEBCwUAMBUxEzARBgNVBAMTCmV4YW1wbGUtdGEX
DTIzMDkyMzE1NTUzOFoXDTIzMTAyMzE1NTUzOFqgLzAtMB8GA1UdIwQYMBaAFMC9
Ul2+0niyFuyzo0OV0gYLmQgyMAoGA1UdFAQDAgEEMA0GCSqGSIb3DQEBCwUAA4IB
AQCngOu+Nq3WC4y/pHtLoheAOtNg32WWsKPNiEyL+QalmOtURUsWMzOq41bmoPzQ
NDQoRmXe9mvohAVRe0CnM7A07HOtSfjw5aoouPXGTtfwEomHG2CYk+2U1bvxgZyA
E1c5TvyhkabFMO0+857wqxRP+ht9NV0lMX6kUFlEOCw3ELVd9oNNRBwKQtXj1huM
6Sf26va2a1tnC5zP01hN+EY3S9T5T1gcgPGBcqRWKoXJEbRzCrLsb/TMj5cMpIje
AHZoBojVAmvL1AIH/BnGAQj0+XqaJ0axHvlqJa8iX8QwKqhp+o6sv/atY2QDDRmE
Yjq/VrBVKu5VsDY2Lr29HszA
-----END X509 CRL-----

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-----
MIIE7DCCA9SgAwIBAgIUcyCzS10hdfG65kbRq7toQAvRDLkwDQYJKoZIhvcNAQEL
BQAwFTETMBEGA1UEAxMKZXhhbXBsZS10YTAeFw0yMzA5MjMxNTU1MzhaFw0yNDA5
MjIxNTU1MzhaMDMxMTAvBgNVBAMTKDNBQ0UyQ0VGNEZCMjFCN0QxMUUzRTE4NEVG
QzFFMjk3QjM3Nzg2NDIwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDc
zz1qwTxC2ocw5rqp8ktm2XyYkl8riBVuqlXwfefTxsR2YFpgz9vkYUd5Az9EVEG7
6wGIyZbtmhK63eEeaqbKz2GHub467498BXeVrYysO+YuIGgCEYKznNDZ4j5aaDbo
j5+4/z0Qvv6HEsxQd0f8br6lKJwgeRM6+fm7796HNPB0aqD7Zj9NRCLXjbB0DCgJ
liH6rXMKR86ofgll9V2mRjesvhdKYgkGbOif9rvxVpLJ/6zdru5CE9yeuJZ59l+n
YH/r6PzdJ4Q7yKrJX8qD6A60j4+biaU4MQ72KpsjhQNTTqF/HRwi0N54GDaknEwE
TnJQHgLJDYqww9yKWtjjAgMBAAGjggIUMIICEDAdBgNVHQ4EFgQUOs4s70+yG30R
4+GE78Hil7N3hkIwHwYDVR0jBBgwFoAUwL1SXb7SeLIW7LOjQ5XSBguZCDIwDwYD
VR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMCAQYwGAYDVR0gAQH/BA4wDDAKBggr
BgEFBQcOAjBDBgNVHR8EPDA6MDigNqA0hjJyc3luYzovL3Jwa2kuZXhhbXBsZS5u
ZXQvcmVwb3NpdG9yeS9leGFtcGxlLXRhLmNybDBOBggrBgEFBQcBAQRCMEAwPgYI
KwYBBQUHMAKGMnJzeW5jOi8vcnBraS5leGFtcGxlLm5ldC9yZXBvc2l0b3J5L2V4
YW1wbGUtdGEuY2VyMIG5BggrBgEFBQcBCwSBrDCBqTA+BggrBgEFBQcwCoYycnN5
bmM6Ly9ycGtpLmV4YW1wbGUubmV0L3JlcG9zaXRvcnkvZXhhbXBsZS1jYS5tZnQw
NQYIKwYBBQUHMA2GKWh0dHBzOi8vcnJkcC5leGFtcGxlLm5ldC9ub3RpZmljYXRp
b24ueG1sMDAGCCsGAQUFBzAFhiRyc3luYzovL3Jwa2kuZXhhbXBsZS5uZXQvcmVw
b3NpdG9yeS8wHwYIKwYBBQUHAQcBAf8EEDAOMAwEAgABMAYDBADAAAIwIQYIKwYB
BQUHAQgBAf8EEjAQoA4wDDAKAgMA+/ACAwD78TANBgkqhkiG9w0BAQsFAAOCAQEA
arIrZWb22wFmP+hVjhdg3IsKHB6fQdMuUR0u2DyZTVvbL6C+HyGAH32pi5mR/QLX
FAfdqALaB7r68tQTGLIW6bGljT+BqUPJmZcj56x3cBLJlltxwFatTloypjFt3cls
xFCuuD9J2iBxc6odTKi6u0mhQjD+C9m4xkbe8XXWWx85IHm1s6rYbpGgiMWxBC80
qqAzmBHGROWKUEvh00EYIYdiAvyFcrj7QtDiRJL5TDOySVd9pWJkerDzhqwE1IaZ
rpHck+lkYTS7jTD++6v32HG62GdsmryOQUk3aU1rLb3kS8vzaGbrgHpGPid0Hd0x
ZSl1AoIMpp5mZ7/h9aW5+A==
-----END CERTIFICATE-----

The CRL is issued by the CA.

-----BEGIN X509 CRL-----
MIIBrTCBlgIBATANBgkqhkiG9w0BAQsFADAzMTEwLwYDVQQDEygzQUNFMkNFRjRG

```
   QjIxQjdEMTFFM0UxODRFRkMxRTI5N0IzNzc4NjQyFw0yMzA5MjMxNTU1MzhaFw0y
   MzEwMjMxNTU1MzhaoC8wLTAfBgNVHSMEGDAWgBQ6zizvT7IbfRHj4YTvweKXs3eG
   QjAKBgNVHRQEAwIBATANBgkqhkiG9w0BAQsFAAOCAQEACwCNzcAoqbMcUL1kBY65
   YhL95OnBqAcuc99pD4i9c1BmVOl7bXU3cJqLaOZ6Z8CmN0kBbcHyqlHBJ9oA/aYD
   ByhxsjzKk7jxtM2IlTpEvCEqvnGLSVihgS3h0NA+sgWqHGL3Rhcj6hVsi+j9GENc
   T6F9np1mxbI3i2xhgeDJG1pryvH0hWXh7yJiYS8ItNEaIIXDT3szK/J9wnPjukTR
   5MITiK9P3TCFujawb3O7rIT5PPgkM6eiCdwDgt6gjmw6cow5+rMjNHSRa+GOviSd
   gXljVDfJvF4tKHmw59Jc2aFnSGfX1/ITDNiNfXYpUYFOcsqxkYf8F0uO7AtbRmTF
   2w==
   -----END X509 CRL-----
```


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 end entity certificate.


```
   -----BEGIN CERTIFICATE-----
   MIIEVjCCAz6gAwIBAgIUJ605QIPX8rW5m4Zwx3WyuW7hZvAwDQYJKoZIhvcNAQEL
   BQAwMzExMC8GA1UEAxMoM0FDRTJDRUY0RkIyMUI3RDExRTNFMTg0RUZDMUUyOTdC
   Mzc3ODY0MjAeFw0yMzA5MjMxNTU1MzhaFw0yNDA3MTkxNTU1MzhaMDMxMTAvBgNV
   BAMTKDkxNDY1MkEzQkQ1MUMxNDQyNjAxOTg4ODlGNUM0NUFCRjA1M0ExODcwggEi
   MA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCycTQrOb/qB2W3i3Ki8PhA/DEW
   yii2TgGo9pgCwO9lsIRI6Zb/k+aSiWWP9kSczlcQgtPCVwr62hTQZCIowBN0BL0c
   K0/5k1imJdi5qdM3nvKswM8CnoR11vB8pQFwruZmr5xphXRvE+mzuJVLgu2V1upm
   BXuWloeymudh6WWJ+GDjwPXO3RiXBejBrOFNXhaFLe08y4DPfr/S/tXJOBm7QzQp
   tmbPLYtGfprYu45liFFqqP94UeLpISfXd36AKGzqTFCcc3EW9l5UFE1MFLlnoEog
   qtoLoKABt0IkOFGKeC/EgeaBdWLe469ddC9rQft5w6g6cmxG+aYDdIEB34zrAgMB
   AAGjggFgMIIBXDAdBgNVHQ4EFgQUkUZSo71RwUQmAZiIn1xFq/BToYcwHwYDVR0j
   BBgwFoAUOs4s70+yG30R4+GE78Hil7N3hkIwDgYDVR0PAQH/BAQDAgeAMBgGA1Ud
   IAEB/wQOMAwwCgYIKwYBBQUHDgIwYQYDVR0fBFowWDBWoFSgUoZQcnN5bmM6Ly9y
   cGtpLmV4YW1wbGUubmV0L3JlcG9zaXRvcnkvM0FDRTJDRUY0RkIyMUI3RDExRTNF
   MTg0RUZDMUUyOTdCMzc3ODY0Mi5jcmwwbAYIKwYBBQUHAQEEYDBeMFwGCCsGAQUF
   BzAChlByc3luYzovL3Jwa2kuZXhhbXBsZS5uZXQvcmVwb3NpdG9yeS8zQUNFMkNF
   RjRGQjIxQjdEMTFFM0UxODRFRkMxRTI5N0IzNzc4NjQyLmNlcjAfBggrBgEFBQcB
   BwEB/wQQMA4wDAQCAAEwBgMEAMAAAjANBgkqhkiG9w0BAQsFAAOCAQEAlxt25FUe
   e0+uCidTH+4p7At3u2ncgHcGTsag3UcoPjcE/I1JgQJRu9TiM4iNB1C7Lbdd131g
   MdliL5GQ3P4QfKnfkuPR6S1V8suq6ZT1KQRyLJx+EPgDN2rb/iji0TOK6RKPNBdG
   lXVLjth4x/uu1O4V54GLEhDAPQC8IUm5intL/Hx1M1x2ptN/+j5HD3XUXd3x13yi
   s6u758nbA7ND40JNhGG5JNGQgDchL4IQzIhylMNC+bKUiyyMHz3MqoVAklIB86IW
   Ucv72Mekq+i46T/w3RnaGn4x7RAJctVJWw3e5YMrFnQcuuaGOs0QcoxW7Bi4W7Eg
   8fK1fd/f6fjZ9w==
   -----END CERTIFICATE-----
```


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

```
      0 1110: SEQUENCE {
      4  830:  SEQUENCE {
      8    3:   [0] {
     10    1:    INTEGER 2
            :     }
     13   20:   INTEGER
            :    27 AD 39 40 83 D7 F2 B5 B9 9B 86 70 C7 75 B2 B9
            :    6E E1 66 F0
     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 23/09/2023 15:55:38 GMT
 120   13:     UTCTime 19/07/2024 15:55:38 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  352:    [3] {
 486  348:     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   14:       SEQUENCE {
556    3:        OBJECT IDENTIFIER keyUsage (2 5 29 15)
561    1:        BOOLEAN TRUE
564    4:        OCTET STRING, encapsulates {
566    2:         BIT STRING 7 unused bits
          :          '1'B (bit 0)
          :          }
          :        }
570   24:       SEQUENCE {
572    3:        OBJECT IDENTIFIER certificatePolicies (2 5 29 32)
577    1:        BOOLEAN TRUE
580   14:        OCTET STRING, encapsulates {
582   12:         SEQUENCE {
584   10:          SEQUENCE {
586    8:           OBJECT IDENTIFIER
          :            resourceCertificatePolicy (1 3 6 1 5 5 7 14 2)
          :            }
          :           }
          :          }
          :        }
596   97:       SEQUENCE {
598    3:        OBJECT IDENTIFIER
          :         cRLDistributionPoints (2 5 29 31)
603   90:        OCTET STRING, encapsulates {
605   88:         SEQUENCE {
607   86:          SEQUENCE {
609   84:           [0] {
611   82:            [0] {
613   80:             [6]
          :            'rsync://rpki.example.net/repository/3ACE'
          :            '2CEF4FB21B7D11E3E184EFC1E297B3778642.crl'
          :              }
          :             }
          :            }
          :           }
          :          }
          :        }
695  108:       SEQUENCE {
697    8:        OBJECT IDENTIFIER
          :         authorityInfoAccess (1 3 6 1 5 5 7 1 1)
707   96:        OCTET STRING, encapsulates {
709   94:         SEQUENCE {
711   92:          SEQUENCE {
713    8:           OBJECT IDENTIFIER
          :            caIssuers (1 3 6 1 5 5 7 48 2)
723   80:           [6]
          :            'rsync://rpki.example.net/repository/3ACE'
          :            '2CEF4FB21B7D11E3E184EFC1E297B3778642.cer'
          :            }
          :           }
          :          }
          :        }
805   31:       SEQUENCE {
807    8:        OBJECT IDENTIFIER
          :         ipAddrBlocks (1 3 6 1 5 5 7 1 7)
817    1:        BOOLEAN TRUE
820   16:        OCTET STRING, encapsulates {
822   14:         SEQUENCE {
824   12:          SEQUENCE {
826    2:           OCTET STRING 00 01
830    6:           SEQUENCE {
832    4:            BIT STRING
          :             '0100000000000000000000011'B
          :             }
```

```
            :           }
            :          }
            :         }
            :        }
            :       }
            :      }
            :     }
    838   13:   SEQUENCE {
    840    9:    OBJECT IDENTIFIER
            :     sha256WithRSAEncryption (1 2 840 113549 1 1 11)
    851    0:    NULL
            :     }
    853  257:   BIT STRING
            :    97 1B 76 E4 55 1E 7B 4F AE 0A 27 53 1F EE 29 EC
            :    0B 77 BB 69 DC 80 77 06 4E C6 A0 DD 47 28 3E 37
            :    04 FC 8D 49 81 02 51 BB D4 E2 33 88 8D 07 50 BB
            :    2D B7 5D D7 7D 60 31 D9 62 2F 91 90 DC FE 10 7C
            :    A9 DF 92 E3 D1 E9 2D 55 F2 CB AA E9 94 F5 29 04
            :    72 2C 9C 7E 10 F8 03 37 6A DB FE 28 E2 D1 33 8A
            :    E9 12 8F 34 17 46 95 75 4B 8E D8 78 C7 FB AE D4
            :    EE 15 E7 81 8B 12 10 C0 3D 00 BC 21 49 B9 8A 7B
            :    4B FC 7C 75 33 5C 76 A6 D3 7F FA 3E 47 0F 75 D4
            :    5D DD F1 D7 7C A2 B3 AB BB E7 C9 DB 03 B3 43 E3
            :    42 4D 84 61 B9 24 D1 90 80 37 21 2F 82 10 CC 88
            :    72 94 C3 42 F9 B2 94 8B 2C 8C 1F 3D CC AA 85 40
            :    92 52 01 F3 A2 16 51 CB FB D8 C7 A4 AB E8 B8 E9
            :    3F F0 DD 19 DA 1A 7E 31 ED 10 09 72 D5 49 5B 0D
            :    DE E5 83 2B 16 74 1C BA E6 86 3A CD 10 72 8C 56
            :    EC 18 B8 5B B1 20 F1 F2 B5 7D DF DF E9 F8 D9 F7
            :    }
```

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-----
```


The 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
```


```
# MIIGQAYJKoZIhvcNAQcCoIIGMTCCBi0CAQMxDTALBglghkgBZQMEAgEwDQYLKoZ
# IhvcNAQkQAS+gggRaMIIEVjCCAz6gAwIBAgIUJ605QIPX8rW5m4Zwx3WyuW7hZv
# AwDQYJKoZIhvcNAQELBQAwMzExMC8GA1UEAxMoM0FDRTJDRUY0RkIyMUI3RDExR
# TNFMTg0RUZDMUUyOTdCMzc3ODY0MjAeFw0yMzA5MjMxNTU1MzhaFw0yNDA3MTkx
# NTU1MzhaMDMxMTAvBgNVBAMTKDkxNDY1MkEzQkQ1MUMxNDQyNjAxOTg4ODlGNUM
# 0NUFCRjA1M0ExODcwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCycT
# QrOb/qB2W3i3Ki8PhA/DEWyii2TgGo9pgCwO9lsIRI6Zb/k+aSiWWP9kSczlcQg
# tPCVwr62hTQZCIowBN0BL0cK0/5k1imJdi5qdM3nvKswM8CnoR11vB8pQFwruZm
# r5xphXRvE+mzuJVLgu2V1upmBXuWloeymudh6WWJ+GDjwPXO3RiXBejBrOFNXha
# FLe08y4DPfr/S/tXJOBm7QzQptmbPLYtGfprYu45liFFqqP94UeLpISfXd36AKG
# zqTFCcc3EW9l5UFE1MFLlnoEogqtoLoKABt0IkOFGKeC/EgeaBdWLe469ddC9rQ
# ft5w6g6cmxG+aYDdIEB34zrAgMBAAGjggFgMIIBXDAdBgNVHQ4EFgQUkUZSo71R
# wUQmAZiIn1xFq/BToYcwHwYDVR0jBBgwFoAUOs4s70+yG30R4+GE78Hil7N3hkI
# wDgYDVR0PAQH/BAQDAgeAMBgGA1UdIAEB/wQOMAwwCgYIKwYBBQUHDgIwYQYDVR
# 0fBFowWDBWoFSgUoZQcnN5bmM6Ly9ycGtpLmV4YW1wbGUubmV0L3JlcG9zaXRvc
# nkvM0FDRTJDRUY0RkIyMUI3RDExRTNFMTg0RUZDMUUyOTdCMzc3ODY0Mi5jcmww
# bAYIKwYBBQUHAQEEYDBeMFwGCCsGAQUFBzAChlByc3luYzovL3Jwa2kuZXhhbXB
# sZS5uZXQvcmVwb3NpdG9yeS8zQUNFMkNFRjRGQjIxQjdEMTFFM0UxODRFRkMxRT
# I5N0IzNzc4NjQyLmNlcjAfBggrBgEFBQcBBwEB/wQQMA4wDAQCAAEwBgMEAMAAA
# jANBgkqhkiG9w0BAQsFAAOCAQEAlxt25FUee0+uCidTH+4p7At3u2ncgHcGTsag
# 3UcoPjcE/I1JgQJRu9TiM4iNB1C7Lbdd131gMdliL5GQ3P4QfKnfkuPR6S1V8su
# q6ZT1KQRyLJx+EPgDN2rb/iji0TOK6RKPNBdGlXVLjth4x/uu1O4V54GLEhDAPQ
# C8IUm5intL/Hx1M1x2ptN/+j5HD3XUXd3x13yis6u758nbA7ND40JNhGG5JNGQg
# DchL4IQzIhylMNC+bKUiyyMHz3MqoVAklIB86IWUcv72Mekq+i46T/w3RnaGn4x
# 7RAJctVJWw3e5YMrFnQcuuaGOs0QcoxW7Bi4W7Eg8fK1fd/f6fjZ9zGCAaowggG
# mAgEDgBSRRlKjvVHBRCYBmIifXEWr8FOhhzALBglghkgBZQMEAgGgazAaBgkqhk
# iG9w0BCQMxDQYLKoZIhvcNAQkQAS8wHAYJKoZIhvcNAQkFMQ8XDTIzMDkyMzE1N
# TUzOFowLwYJKoZIhvcNAQkEMSIEICvi8p5S8ckg2wTRhDBQzGijjyqs5T6I+4Vt
# BHypfcEWMA0GCSqGSIb3DQEBAQUABIIBAKZND7pKdVdfpB6zaJN89wTt+sXd0io
# 0WULMc+o6gRJFt3wmKNW2nYPrDbocJ+Q/rDMGxbp4QetJ0MQtn1+AYAS8v5jPDO
# 4a63U4/mJ2D3wSnQsDP0lUVknqRzfnS66HgHqiOVdHB0U+OnMEJuqHNTLx0dknb
# L3zwxyDJTHdo+dMB0U9xdcjwpsPM3xqg57EXj5EIQK5JbardXCjrsysAnEdktUY
# oyayGNbbQelANYJcOmuHhSXArR+qqzvNP2MDRqqKEcpd65YW6FSnqlVMIBH2M3P
# D2F0p3sdm4IeGAZWaERVB4AXO1PUFDNdhamr4XpIwqIoAig7xiLm7j8qu5Oc=
```


```
# End Signature: 192.0.2.0/24
```

Acknowledgments

Thanks to Rob Austein for the 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, 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 Job Snijders, who also found an ASN.1 'inherit' issue, Adrian Farrel, Antonio Prado, Francesca Palombini, Jean-Michel Combes (INTDIR), John Scudder, Kyle Rose (SECDIR), Martin Duke, Mohamed Boucadair, Murray Kucherawy, Paul Kyzivat (GENART), Rob Wilton, Roman Danyliw, and Ties de Kock.

Authors' Addresses

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

Massimo Candela
NTT
Veemweg 23

3771 MT Barneveld
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