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.

Copyright Notice
Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.
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.

```
inетnum: 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
# MIIGlwYJKoZIhvcNAQcCoIIGiDCCBoQCAQMxDTALBgglghkgBZQMEAgEwDQYKoZ
# IhvcNAQkQAS+ggsxMIIErTCCASWgAwIBAgIUJ685QIPX8rW5m4zWx3WyW7hZu
# imwYkXpiMxw44EZqDj136MiWsRDLdgoijBBcGbibwyAfGeR46k5raZCGvxG+4xa
# 08PDxTfIywAnBjRBKAqAZyX5CHfm58jUXsZJ7Ieq1S7G6Kk=
# 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:

<table>
<thead>
<tr>
<th>Decimal</th>
<th>Description</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>47</td>
<td>id-ct-geofeedCSVwithCRLF</td>
<td>RFC 9092</td>
</tr>
</tbody>
</table>

Table 1

9. References

9.1. Normative References


Bush, et al. Standards Track Page 9
9.2. Informative References


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-----
MIIEpjCCayagAwIBAgIUPIUsUFJ4e/7pKZ6E14aBdkbYzms1gwDQYJKoZIhvcNAQEL
BQAwFTETMBAEKUEAxMKXhbxBXsZS18YTaeFw0yMADA5MDMxODU0NTRaFw0zMDA5
MDEXDU0NTRaMBUeXaZARBgNBVMBAMCMv4YW1wbGUtdGWggeEiMA0GCSqGSIb3DQE
AQUAA4IBDwAwggEKAQCEI1MmMDGCbhq/a3VrNaOKMr1HVLKxGoG7VF/13HZJ
0twobUzI3Jz+XeD+knAURelMLWtrsgdkTQFqfing0pRemxrl55+X7nLpe5nwwaBH
XqddOLubmbkbaGanGcm6T/r9DNKk1Z46uc2p7YUo8wN08mo0aqFL2FSyzzzWzine
E7LYZ4a3LvGnB1jFp/JvM6pgrtoMVuee5RV6Twa7Lw384ICjBBphfy/HFPO1arb
09gsBCUmqz+RvroAia8c8vgBF/fPCz9o17GdMib679JxxFrw4rj08nMjgsXq
jaVCG7o9Rc+eIAChw7Uroc6h7Y7lG0kDZ75i8mlQa3AgMBAAGjgGEMIIbgDad
BgNVQ4EFqGU3hNEuwuUGYNCHY1TbetcUR83pNdYwHWYDVR8jBBwfoUAUI3hNEuwuU
GNCHY1TbetcUR83pNdYwDwYDVR8TAQH/BAUuAwEB/zAOBgNVHQ8BAf8EBAMCAQYw
GAYDVR0gAQH/YBAwFkA8AOwggBQMBfA8EAMCAQQgOgYDVR0gAQIwDBBwYOUuem
KwYBBQUHMAQgGmJzeW5jOi8vcnBraS5eGFCtGx1Lm5dG9yZXVc21ob3J5L2V4
YW1wbGUtdGEBwZ0DUCQGAQQFBzANhilodHRwczovL3JyZG4yMwYwDQYJKoZIhvc
-----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++gAwIBAgIUcyCzS10hdf665kbRg7toQAVrDKowDQYJKoZIhvcNAQEL
BQAwETMBEGA1UEAxMKXhhbXBSzS10YTAeFw0yMDA5MDk0OTyMTIafw8yMTA5
MDMxOTyMTIaMDMxOTyMTAvBngV8VMTDEBMDQwOj4yQVNECMjFNC0BxMUZsRTE4NEVG
QzFFMjk3QjM3Nzg2NDWggEIaMA0GCSqGSIb3DQEB6QUAA4IBdwBwggEKAoIBAQCq
zz1qwTX2c0c5r8q8tkm2YyEk18riBVu1dWwefNTx82RFpg9vUYd5A9ZOEVE7G
6wGwGZbmhkK663eeaqbfKZ2GHub6474988BExcYrs04+YuGwCEYznnND4Z5aaDbo
j5/4/z0Qv6HExyQdf0f8br61KJwgeRM6+fm7796HNBP0aqD7Zj9NRCXJybBOCDUj
liH6XMNR6ofg1i9VzmRjesvdhKYyGb0if9rvxVpLJ/6zdru5C9aye9UZS51+n
YH/6PzdJ4Q7yKrJX8q6D6A8j4+biaU4M472Kpsj7QNTtQf/HrwN6N54GdaknWE
TnJQHgcLJYqw9ywKttj/JagMBAAGjggCMIICKzzAdBgNVH4EFgQU0s4s68+yG39R
4+EG78Hl17n3hIzwvYDvRo7jBbwfoA03hNeuwvUGNCYT1BatcUR83pdYwDwYD
VR8TQAfH/BAuwAEBz/zA0BnNvH8QBAf8EBAMCAQYwGAYDVR80aGQH/B4awDDAKBgr
BgEFBQoCAjBhBgNVHR8EByMFagVKBSh1baj31uYzovL3JwawkuZXhhbXBdBzzSuu
ZXQvcmWvb3NpdG9yeb8zQUNMFNRFrRGQjIxjdeMTFvF80UXQDFRkMxRTIS0Iz
Nzc4njyLmNyBDB0B0rgBrEFBQcBAQRQCEAwPgy1KwYBBQUHMAKGMnJzeW5j0i8v
bnvRa516EfGgCQ011m51dC9y2XBvc2l0b3J5L2J4YW1wbGtlbGdUEyYyZMIg5bgr
BgEFBQcBBwSeD3BqTA+BggBrEFQcCwYcCmnvN5bmM6Ly9ycGtpLm4YW1wbGUu
bmV0L3JjLaG9zXRvcmvXZ0hBbGS3Zj55tZnQ0YKwYBBQUHAQGIJAhWkHdHz
0i8vcnkc5C16EfGtGcX11m51dC9ub3RpmjYXRpb24ueG1sMDAGCCs5GAFBFzA
htrc31uYzovL3Jw2kuZXhhbXzSBzzSuRQvE8icvE3Npdg9yeSb8wHYKwYBBQH
AQCnBAb85EDEODAuwEAeABMAYDADBADAAAIHgwYKwYBBQUHAgEjAQoAw4wDFAQGMA
+ACAwD7TNABqkhk1G9wBBAQsFAQOCQAEANLu+d1ZsuUTi31XWGuCThlaiW4ad0
Kupi7pMVn2XbcHmgM3oC19Kb0vZ2t5j55MehgUH4YLmICYe4fz5e8089s/vIm
cG529+W0uGizn7mp5v/vS/379g4aMzK6KpqhH6Brw6mEmQy89hpmcmv3X3Wmx89
mL1L1neMptwkBqSycnM0UMjLjcs8C9vk0a3sSWrdw88W6QyTXo8ZHzFZQoj86YFzEq
V/dBdCfedEwTFy112n2Xqho1/fo/eBdC4u2GdqRk3+MV5s+uHw+/P7tst777FgFfY
yxqvG6q6ldxZV2mHmCkmKuTu/BqCDGjot9may31xkr345Bu+XFMVihm90w==
-----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-----
MIIEpTCCA42gAwIBAgIUI665QIPX8rW5m4zwx3WyuW7zuQwDQYJKoZIhvcNAQEL
BQAwMzExNCSC8A1UEAxMoM0FDRTJDRUY0RkIyMUI3RDExRTNFMTg0RUZMUMyOTdc
Mzc3ODY0MjAeFwWyMTA1MjAxNjA1NDAwFwyMyMjAzMTYxNjA1NDAwMDMxMTAvBgNV
BAMTKDQxMODY1MkEzOk1MUNxNDYyNjAxOTg4ODI1GNM0UmFdRjA1MDE0M0DwggEi
MA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCycTQrOb/k+aSIW9P9kScz1cQgT
PCVwr62hTQZC1owBNO8L8cK0/5k1imJdi5qdm3nvKswM8CnoR11vB8pQFwrUZmr5x
phXRVe+azuJVLgu2V1upmBXuWl0eymudh6WJJ+GDjwPX03RiXBejBrOFNXhaFLe80y
4Dfpr/S/tXJO8m7QzQp
tmBPlyTgfPryU451lffqqP94uEAPISfxd36AKGzqTFCCm39w51UFE1MFlnoEog
qtoLoKAbt0IKOFGKeCE/geeaBdWLe469ddC9Qf5w6g6cuniG+aYDdBEB34zAgMB
AAAgggGvMIIbqzAdBgnVHQ4EFGQOQUKUZSo71RwUQmAzIn1xQf/BToYcwhhYVDR0j
BBgwoFAU0sI8s70+yG38R4+GE78H117N3kJWdAYDVR8TAQH/BAiwaAaOBgNVHQ9BB
Af8EBEAMBC4awGAYDVR0gAQH/BA4wDADAQbhBgEFBQcOAJBhBgnVH8REW8yMYFMa
VKBSh1yc3luYzovL3j2u2kXzhbxXbSsZ5u1QX0vcmW3g3p49yE8ZQUNF9k
RjAGGyjQHjDEMTFMMU0uXQDVRkMrXRl5N0fizNzc4nYjQyLMjybD8bsBgrgEBFQcCB
AQRgMf4xXAYIKYkBBQMAHMGKnXJHe5sJ688vbn3JLsGfc6xL1M3C9yZBv
c210b3J5LzNB0uQyO0VgNEZCMjFCN0QxMUIzRTE4NEVQzFMMj3QjM3ng2D1u
Y2VyMBK6CcsGAQUF8weEahQH/BAowDAAGAIAAQAMEUCGscGAQUF6WELkovNzaA1
BgrgEBFQcwyDyhpHR8cHM6Ly9ycmRwLmV4YW1wbGUubmV0L25vC1maWNhdGlv
bi54bWwTDQYJKoZIhvcNAQELBQADQgBAEjC98gVr9pM07uikHaHyP0453tmJt+AkN
07fsk/gQw/e90DJ7c1pvhj4uy3s7g7F70C7.requestFocusf/JEOjce+ArgVjbi2Brz
ZSwAn8B46Sw3ktt6cenaif6Aw6q0NspAePMBd2Vg/9sKFW0vJFVQvNQci1XiP
5rGJwpBcOMv5a/7adjfXwpon0iji70GloQmC2TPZpydKZjIxExATdfEQsa33xD
nlpp/+9xKuNVYRTcC360WraAV3jN6f6rDE8r8x3ylISV6JeCQ4YRYwBmsjJc
/tiJLM7ZxyEiBZlzn6/SessJAswRIGps2HeCt/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
38   16:      sha256WithRSAEncryption (1 2 840 113549 1 1 11)
48   0:      NULL
50   51:    SEQUENCE {
52   49:      SET {
54   47:        SEQUENCE {
56   45:          OBJECT IDENTIFIER commonName (2 5 4 3)
61   40:            PrintableString
62   36:              '3ACE2CEF4FB21B7D11E3E184EFC1E297B3778642'
63   34:          }
64   32:        }
96  130:    SEQUENCE {
98  128:      UTCTime 20/05/2021 16:05:45 GMT
103  127:      UTCTime 16/03/2022 16:05:45 GMT
109  123:    }
135  51:    SEQUENCE {
137  49:      SET {
139  47:        SEQUENCE {
141  45:          OBJECT IDENTIFIER commonName (2 5 4 3)
146  40:            PrintableString
147  36:              '914652A3BD51C144260198889F5C45ABF053A187'
148  34:          }
149  32:        }
181  290:    SEQUENCE {
183  288:      SEQUENCE {
185  286:        OBJECT IDENTIFIER rsaEncryption
194  13:          (1 2 840 113549 1 1 1)
205   0:          NULL
207  271:        BIT STRING, encapsulates {
212  266:          SEQUENCE {
216  257:            INTEGER
217  256:              00 B2 71 34 2B 39 BF EA 07 65 B7 8B 72 A2 F0 F8
221  252:              40 FC 31 16 CA 28 B6 4E 01 A8 F6 98 02 C0 EF 65
225  251:              B0 48 49 96 FF 93 E6 92 89 65 8F F6 44 9C CE
229  248:              57 18 82 D3 C2 57 0A FA DA 14 D0 64 22 28 C0 13
233  247:              74 04 BD 1C 2B 4F F9 93 58 A6 25 D8 B9 A9 D3 37
237  242:              9E F2 AC 0C CF 02 9E 84 75 D6 F0 7C A5 01 70 AE
241  238:              E6 66 AF 9C 69 85 74 6F 13 E9 B3 B8 95 4B 82 ED
245  234:              95 D6 EA 66 05 7B 96 96 87 B2 9A E7 61 E9 65 B9
249  230:              F8 60 E3 0C F5 CE DD 18 97 05 E8 C1 AC E1 4D 5E
253  226:              16 85 2D ED 3C CB 80 CF 7E BF D2 FE D5 C9 38 19
257  222:              BB 43 34 29 B6 66 CF 2D 8B 46 7E 9A D8 BB 8E 65
261  218:              88 51 6A A8 FF 78 51 E2 E9 21 27 D7 77 7E 80 28
265  214:              6C EA 4C 58 9C 73 71 16 F6 5E 54 14 4D 4C 14 B9
269  209:              67 A8 4A 29 AA DA 0B A6 A0 01 B7 42 24 38 51 8A
273  205:              78 2F C4 81 E6 81 75 62 DE E3 AF 5D 74 2F 6B 41
277  201:            }
281  199:          }
283  197:        }
285  195:    }
290  192:  }
297  191: }

:     FB 79 C3 A8 3A 72 6C 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
501  16:     91 46 52 A3 BD 51 C1 44 26 01 98 88 9F 5C 45 AB
505  13:     F9 53 A1 87
507  18:     }
521  31:     SEQUENCE {
523  3:     OBJECT IDENTIFIER authorityKeyIdentifier (2 5 29 35)
528  24:     OCTET STRING, encapsulates {
530  22:     SEQUENCE {
532  20:     [0]
534  16:     3A CE 2C EF 4F B2 1B 7D 11 E3 E1 84 EF C1 E2 97
538  15:     B3 77 86 42
540  17:     }
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
582  1:     '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
602  7:         resourceCertificatePolicy (1 3 6 1 5 7 14 2)
604  12:         }
606  16:         }
608  97:     SEQUENCE {
610  3:     OBJECT IDENTIFIER cRLDistributionPoints (2 5 29 31)
612  90:     OCTET STRING, encapsulates {
614  88:     SEQUENCE {
616  86:     SEQUENCE {
618  84:     SEQUENCE {
620  82:     SEQUENCE {
622  80:     SEQUENCE {
624  78:     SEQUENCE {
626  76:     SEQUENCE {
628  74:     SEQUENCE {
630  72:     SEQUENCE {
632  70:     SEQUENCE {
634  68:     SEQUENCE {
636  66:     SEQUENCE {
638  64:     SEQUENCE {
640  62:     SEQUENCE {
642  60:     SEQUENCE {
644  58:     SEQUENCE {
646  56:     SEQUENCE {
648  54:     SEQUENCE {
650  52:     SEQUENCE {
652  50:     SEQUENCE {
654  48:     SEQUENCE {
656  46:     SEQUENCE {
658  44:     SEQUENCE {
660  42:     SEQUENCE {
662  40:     SEQUENCE {
664  38:     SEQUENCE {
666  36:     SEQUENCE {
668  34:     SEQUENCE {
670  32:     SEQUENCE {
672  30:     SEQUENCE {
674  28:     SEQUENCE {
676  26:     SEQUENCE {
678  24:     SEQUENCE {
680  22:     SEQUENCE {
682  20:     SEQUENCE {
684  18:     SEQUENCE {
686  16:     SEQUENCE {
688  14:     SEQUENCE {
690  12:     SEQUENCE {
692  10:     SEQUENCE {
694  8:     SEQUENCE {
696  6:     SEQUENCE {
698  4:     SEQUENCE {
700  2:     SEQUENCE {
702  0:     SEQUENCE {}
704  2:     BIT STRING 4 unused bits
706  1:     '1'B (bit 0)
708  4:     OCTET STRING, encapsulates {
710  2:     BIT STRING 7 unused bits
712  1:     '1'B (bit 0)
714  4:     OCTET STRING, encapsulates {
716  2:     BIT STRING 7 unused bits
718  1:     '1'B (bit 0)
720  4:     OCTET STRING, encapsulates {
722  2:     BIT STRING 7 unused bits
724  1:     '1'B (bit 0)
726  4:     OCTET STRING, encapsulates {
728  2:     BIT STRING 7 unused bits
730  1:     '1'B (bit 0)
732  4:     OCTET STRING, encapsulates {
734  2:     BIT STRING 7 unused bits
736  1:     '1'B (bit 0)
738  4:     OCTET STRING, encapsulates {
740  2:     BIT STRING 7 unused bits
742  1:     '1'B (bit 0)
744  4:     OCTET STRING, encapsulates {
746  2:     BIT STRING 7 unused bits
748  1:     '1'B (bit 0)
750  4:     OCTET STRING, encapsulates {
752  2:     BIT STRING 7 unused bits
754  1:     '1'B (bit 0)
756  4:     OCTET STRING, encapsulates {
758  2:     BIT STRING 7 unused bits
760  1:     '1'B (bit 0)
762  4:     OCTET STRING, encapsulates {
764  2:     BIT STRING 7 unused bits
766  1:     '1'B (bit 0)
SEQUENCE {
  OBJECT IDENTIFIER authorityInfoAccess (1 3 6 1 5 5 7 1 1)
  OCTET STRING, encapsulates {
    SEQUENCE {
      SEQUENCE {
        OBJECT IDENTIFIER caIssuers (1 3 6 1 5 5 7 48 2)
        [6]
        'rsync://rpki.example.net/repository/3ACE2CEF4F'
        'B21B7D11E3184EFC1E297B3778642.cer'
      }
    }
  }
}

SEQUENCE {
  OBJECT IDENTIFIER ipAddrBlocks (1 3 6 1 5 5 7 1 7)
  BOOLEAN TRUE
  OCTET STRING, encapsulates {
    SEQUENCE {
      SEQUENCE {
        OCTET STRING 00 01
        NULL
      }
    }
  }
}

SEQUENCE {
  OBJECT IDENTIFIER subjectInfoAccess (1 3 6 1 5 5 7 1 11)
  OCTET STRING, encapsulates {
    SEQUENCE {
      OBJECT IDENTIFIER '1 3 6 1 5 5 7 48 13'
      [6]
      'https://rrdp.example.net/notification.xml'
    }
  }
}

SEQUENCE {
  OBJECT IDENTIFIER sha256WithRSAEncryption (1 2 840 113549 1 1 11)
  NULL
}
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-----
MIIEpQIBAAKAQEAsnE0Kzm/6gdIt4tyovD4QPwxFsootk4BqPaYAsDvZbCESoWm /5Pmkollj/ZenM5XElTwlcKetoU0GQIKMAtdAS9HcTPz+ZNYpiXYuanTN5yrdMDP Ap6EddbfkUBck7m7zpq+caywO8bxPps71V54LtdbqzV71paHspprnyellifiq48D1 zt0Y1w/XowahzTV4Whs3pMuAz36/0v7vyTgZu0M0kBzmyz2Lr6a2LuOZyRaqj/ eFH165En13d+wCh66kJk0HnxFvZveVBRTNBSS6BkIrCraC6GcAbdCJDhRtgvrXHm gXV3u0vXXqva8H7ecOO8jRVmm33ASAbd+M6wIDAQABo1BAQcyBo8Fflm88BoRo 18AkjFGSPEoZisr1z5bUVl1i92TELe7zn6L61ym260J/5th+iChG0/dq1hXio p156C5yc97Fbb12EO8sCuuuQKfJ8ZC03GyoSv0oxKJeMm+/-5oYzXrOjR6unW70z o1/Je5pLGUC1g5X6tz95s5BP3/1uAavQHsv6+eVdKgLQ3mvj/1vILBO/CN036eV6GJ mpkvmgPyfJcET9wBw0a0yn3jxJb36+M/QjUP28oNIvN/IkoPzRzn8qChubCuJ651 IsaFsgoiThm4ZtvCh/IDq6+dcMucmtJiRcyw7Wfdhjflfl1vPve9c/OpwQEVF t3ArWUt5aGBANs4704yHx04mctLIE7G71/tf9bP4KkIuY4wR48ByEcucqMc4yhmt MCPfOFLOQOe710wCKj2P/L/7EKue9yx7G5KmAHY6j0jvcRkvG516W80jQ8P126M Y9hmGzMojtsdha1MmQwKjvjmw4WqMgyGeq+PnjjSVkgTt7+BpxiUGBAAvoGABANGb 26FF5cDLpx0d3zA1yX05ugwCaw3Plv17uZRPa/zbMEmLETy6eBakknIRWmn01l e+iA2xwm+929PDTO8qCEF91teyznqLa05kkAdjiFuVvV3icLoGO399FJr4hKensm FGsII+S3XkqvHNIjJfFwzq4aB8EioAmjDGyBzYQrYQaOBGAM6tUJd36KDU+hiS6w6 0ZTPSrzFh/pZp03pCW78/Q8bzd2w4E1i6qgBAR74NPvlq9/yHBuTT9/xv3e7phO0 Os7NpLz5mKH4hk5F2f6z2n9lOFk0abqa0l1xuJoDEU2YRoALae9F6/Rog6Phyz vLe5QscrbU0XlKhNe+7zB5aBoBAGDKsDEb/dbqybaAYpmwhH2dRskphg71nwc DNm9qWlaJ6zWrl+MB716q8raRNelUA1avqWv6HLr/R0BQG6N1U1ac5/qfEt2UXxgk ftaMvk61tujoyK3tsmNh8HzupWk6wHcEb+YVdmm52ZGw2A75rD11lL+9+2dc PwDXVubRAOGAdxeSwoluxZz18rsrAkrCStaXn0WazXeUI5Lsve8nKZ7UdQZ E3nq2j5XPTUW11+a1gNFEGR0NtcQV0680/sFZUhu52sqq5MwVvYNh1TB5aP8x+pV iFc2OUvUQcEn6PA+YQK5FU11rA1Tm05Sm5DrvNvU10L2fxY7z8Fzv6Y =-----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
# MIIGjwYJKoZIhvcNAQcCoIIGGnCwCaQMxDTALBglghkgBZQMEAwDQYJKKoZ
# wvDM0Z8MDUyOTdCi6M230YwMjAeF8yYMTA1MjAxNjA1NDV0F8yMjAeF8y
# NAI1MDMxMTAvBZNvMQB85MDUyOTdCi6M230YwMjAeF8yYMTA1MjAxNjA1NDV0F8yMjAeF8y
# 0NUFCrajA1M0ExODc4wEiNA6GCSqGSIb3DQEBAQUAA4IBDwAggEoAwDQYJKKoZ
# Rq0b/q2W3S13N18pa/DEWy125qGo9pgCw901sIR16Bk/k+aSi1W9P9kScz1CQg
# tPVCwvrh2TQZCIowBN0BLcK8/5ki1imDj165mdNSW88r0FQWFrW2m
# r5XphXRVr+muzJVLj2v1UpBWX0eomud6WWL+jGd/jWPX3RixBejFr0FNXha
# Flee8by4DFPr/S/txJ0Bm7Q29qtpmbPFT9gpYa4511FqqP949eLP1s46XAKG
# qzTFCCw9E9S15UFE1MFLnoEogrtloKAA6tikOFGKeC/EgbeAdWle469dcd9RQ
# ft5w6g6cmXGa+YdIDEB34zrAgMBAAGggGwMIIAqZadBGrWVHQ4EFQgQKUZ2O71R
# wUQmAZ1xnuFq/BTOycwHzYDvR07jBBQwFoAU0s4s7+yG30R4+4E78H17N3hK1
# wDAyYDvR8TAQY/BAlwDA0BNgWVHQ8BAF8EMBC4hAwGAYDvR8AqH/Ah4wDADKBg
# grBegEFBCoQjA1BhBnVWHR8EjW1YQmaVWKBSh1ByC31uVZovalJ3w26uZV4hbbSBZ
# S5uZQvcmYwB3npDGYyE8ZQUNFMKNF8rQ0jixQjD7EFTF3MFOxfD8FRKjXRIT1
# N8iZnzC4nJOyLmYyDBsBbgrBEGFBCoBAOQRMFy4XAYIKyV8B0UHMAUKHAJUWZ5
# j0I8vcbnEraS56EFgcxl5m5dC9yEZBcv310e3J35LazNB00Uy0QVNEZCMJFCN0
# XmuuzRE4NEVGVZ2FMfKj3QjMzq2NDiuY2YVBmgcGCAsgGJQGfW8EBAHQ8/BAowC
# DABGIAAAUAEQUbCGCggGACFQwEBLQkwNzA18gBrgBFBBC6ywy0h6CHM6L6y9y
# cmRWlmV4YWtb0uubmW8L2vd5dimaWmhdG1vi54ubWwDQYJIoVhcNAQcMEAhuQ/BAowC
# DggeEAEjC9q8V0pM7uKiAhPyIP0P453mtJ+Akn07fsk/9Qw/e90P7Cr31hvyju4
# y3sgf37Q0jK1NGrqyBq/01+1-osB329rs+zanB4s6swnstw6cenaifa6E
# ww6q08m0spAepMBd2Vg/9sFkFowJFvQogCq1iX5r5GJPWBC0M52a/77dfjXpwn
# 0iijTOGnIoQmStPZ2qzjK1jXeADTeEQeS633XnlDcdn+/r9xUNYRTcC36Ed0Rw
# aVAsj3x6F6rDEr8xss3Y1LStZ6JeQ4YRBmMsjhc/tiJLM7ZYXe5r1rZYTn6
# n/SessJasWkg2sEpC+EC/HHXZAMO49hUqggXgMIEpIBA44AUZKuko71RwQMa
# ZiIn1xfq/BoYcwwCwYjYIIZIANWDBAIoBoGwGyYKoZIhvcNAQdMD8E64CGqGSIb3
# DQEJEAEBMB6C8QsS6ib3DQEBTl7F8yYMTA1MjAxNjA1NDV0F8yMjAeF8y
# JdBIE8CAr4KveulkHJINSE0QyLyMxxo48q90rU+iPufBo8R8X3BFJ1ANBqkghk1G9w
# 0BAQEFaASCQB5hS6Q8URU3ECVOCf4nC63j3r0j+NFlYTDABsFGc6TcN4rxe77SA
# Iny5if3u1uoGqhVYs0jiOixnYpFt4118KvyeXh8A/HLP4eAeocJnt3Dc3gt38M
# o48q5q999TPQX3hbsm516x0i+TKVMQzE42s0POX3M0+6eK3h/vBkwN1s1ayM
# 0MUnPDTRFBZLJ3EJGPWfIHZHcreypvebR7jI8s5vpsqFv2D9w+w+yheZP0uePm7
# YqL6w0/E99P9Be9u1h+IhmBicZ/2BkZ3VRjrlUU+49edSTKZ2sJyhCbbV2Ufgi
# SF0QagJzzjilyN3BDQLV8R9pGh0P0P5IS1KKh2na
```

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; Rob Snijders, who provided running code; and Kevin Pack. Also, thanks to the following geolocation providers who are consuming geofeeds with this described solution: Jonathan Koege (ipdata.co), Ben Dwolingo (ipinfo.io), and Pol Nisenblat (bigdatacloud.com). For an amazing number of helpful reviews, we thank Adrian Farrel, Antonio Prado, Francesca

---

**End Signature:** 192.0.2.0 - 192.0.2.255
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