Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 18, 2015

T. Bruijnzeels
O. Muravskiy
RIPE NCC
B. Weber
Cobenian
R. Austein
Dragon Research Labs
D. Mandelberg
BBN Technologies
February 16, 2015

RPKI Repository Delta Protocol
draft-ietf-sidr-delta-protocol-00

Abstract

In the Resource Public Key Infrastructure (RPKI), certificate authorities publish certificates, including end entity certificates, and CRLs to repositories on publication servers. Relying Parties (RP) retrieve the published information from the repository and MAY store it in a cache. This document specifies a delta protocol which provides relying parties with a mechanism to query a repository for changes, thus enabling the RP to keep its state in sync with the repository.

Status of this Memo

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

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

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

This Internet-Draft will expire on August 18, 2015.

Table of Contents

1. Requirements notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. Introduction

In the Resource Public Key Infrastructure (RPKI), certification authorities (CAs) publish certificates [RFC6487], RPKI signed objects [RFC6488] , manifests [RFC6486] and CRLs to repositories. CAs may have an embedded mechanism to publish to these repositories, or they may use a separate publication server and communication protocol. RPKI repositories are currently accessible using rsync, allowing Relying Parties (RPs) to synchronise a local copy of the RPKI repository used for validation with the central repositories using the rsync protocol [RFC6481].

This document specifies an alternative repository access protocol based on notification, snapshot and delta files that an RP can retrieve over http(s). This allows RPs to perform a full (re-)synchronisation of their local copy of the repository using snapshot files. However, typically RPs will use delta files to keep their local repository updated after initial synchronisation.

This protocol is designed to be consistent with the publication protocol [I-D.ietf-sidr-publication] and treats publication events of one or more repository objects as immutable events that can be communicated to relying parties. This approach helps to minimize the amount of data that traverses the network and thus helps minimize the amount of time until repository convergence occurs. This protocol also provides a standards based way to obtain consistent, point in time views of a single repository eliminating a number of consistency related issues. Finally, this approach allows for caching infrastructure to be used to serve this immutable data, and thus helps to reduce the load on a publication server when a large a number of relying parties are querying it.

3. RPKI Repository Delta Protocol Implementation

3.1. Informal Overview

Certification Authorities (CA) in the RPKI use a publication server to publish their RPKI products, such as manifests, CRLs, signed certificates and RPKI signed objects. This publication server may be remote, or embedded in the CA engine itself. Certificates in the RPKI that use a publication server that supports this delta protocol include a special Subject Information Access (SIA) pointer referring to a notification file.

The notification file includes a globally unique session_id in the form of a version 4 UUID, and serial number that can be used by the Relying Party (RP) to determine if it and the repository are synchronised. Furthermore it includes a link to the most recent complete snapshot of current objects that are published by the publication servers, and a list of links to delta files, for each revision starting at a point determined by the publication server, up to the current revision of the repository.

This notification file is intended to be small so that is can easily be fetched over HTTP(S). The publication server may use HTTP caching infrastructure to reduce its load. The publication server should should avoid using a long caching interval, since the length of this interval determines when RPs will receive updated notification files, and thereby new products produced by Certification Authorities using this publication server. It is recommended that of no longer than five minutes is used for caching this file. If the caching infrastructure supports it another useful approach would be to expire the cache for the notification file URI as soon as a new notification file is known to be published.

An RP that first learns about a notification file location can download it, and then proceed to download the latest snapshot file, and thus create a local copy of the repository that is in sync with the publication server. The RP should remember the location of this notification file, the session_id and current serial number.

RPs are encouraged to re-fetch this notification file at regular intervals, but should not try to fetch the same file more frequently than once per minute. After re-fetching the notification file, the RP may find that there are one or more delta files available that allow it to synchronise with the current state.

If no contiguous chain of updates is available, or if the session_id has changed, the latest snapshot should be used instead. In this case the RP should then add the objects found in the latest snapshot to its local repository.

As soon as the RP fetches new content in this way it should start a validation process using its local repository. An example of a reason why an RP may not do this immediately is because it has learned of more than one notification location and it prefers to complete all its updates before validating.

The publication server may use http caching infrastructure to reduce its load. It should be noted that snapshots and deltas for any given session_id and serial number contain an immutable record of the state of the publication server at a certain point in time. For this reason these files can be cached indefinitely. To support this the publication server must use a globally unique URL for the location of

each of these snapshot and delta files. It is recommended that old versions of snapshot and delta files remain available for download for some time after they have last appeared on a notification file to provide some resiliency in case relying parties are slow to process.

3.2. Update Notification File

3.2.1. Purpose

The update notification file is used by RPs to discover whether any changes exist between the state of the publication server's repository and the RP's cache. It describes the location of the files containing the snapshot and incremental deltas which can be used by the RP to synchronize with the repository.

3.2.2. Cache Concerns

A repository server MAY use caching infrastructure to cache the notification file and reduce the load of http(s) requests to a central repository server. However, since this file is used by RPs to determine whether any updates are available it is strongly RECOMMENDED to use a short interval for caching, to avoid unnecessary delays. A maximum of delay of 5 minutes after a new notification file has been published seems like a reasonable compromise. This delay should not cause major problems for RPs and routing since a similar human time scale is expected to be involved in updating the contents of the RPKI on the one hand, i.e. creating and publishing new ROAs or router certificates, and updating actual BGP announcements in routers on the other. That said, real world measurements are needed on this subject, so this recommended maximum time may be subject to change in future.

There are various ways to ensure that the notification file is only cached for a certain time in caching infrastructure and different solutions, such as commercial Content Delivery Networks (CDNs), may provide different ways of achieving this. For example some CDNs have custom support to cache a file such as this notification file indefinetely, but allow a central server to notify the CDN through some protocol that an update is available and trigger the CDN to then refresh this file. In general a publication server may find certain HTTP headers to be useful, such as: Cache-Control: max-age=300

Finally it shoulf be noted that snapshot and delta files are intended to be cache-able for a much longer longer time. In support of this the URIs for each snapshot and delta file for a given session_id and serial number MUST be unique and the contents of those files MUST NOT change.

3.2.3. File Format and Validation

Example notification file:

```
   <notification xmlns="HTTP://www.ripe.net/rpki/rrdp" version="1" session_id="9
df4b597-af9e-4dca-bdda-719cce2c4e28" serial="2">
     <snapshot uri="HTTP://rpki.ripe.net/rpki-ca/rrdp/EEEA7F7AD96D85BBD1F7274FA7
DA0025984A2AF3D5A0538F77BEC732ECB1B068.xml" hash="EEEA7F7AD96D85BBD1F7274FA7DA00
25984A2AF3D5A0538F77BEC732ECB1B068"/>
     <delta serial="2" uri="HTTP://rpki.ripe.net/rpki-ca/rrdp/198BD94315E9372D7F
15688A5A61C7BA40D318210CDC799B6D3F9F24831CF21B.xml" hash="198BD94315E9372D7F1568
8A5A61C7BA40D318210CDC799B6D3F9F24831CF21B"/>
     <delta serial="1" uri="HTTP://rpki.ripe.net/rpki-ca/rrdp/8DE946FDA8C6A6E431
DFE3622E2A3E36B8F477B81FAFCC5E7552CC3350C609CC.xml" hash="8DE946FDA8C6A6E431DFE3
622E2A3E36B8F477B81FAFCC5E7552CC3350C609CC"/>
   </notification>
```

The following validation rules must be observed when creating or parsing notification files:

- A RP MUST NOT process any update notification file that is not well formed, or which does not conform to the RELAX NG schema outlined in Section 5 of this document.
- The XML namespace MUST be HTTP://www.ripe.net/rpki/rrdp
- The encoding MUST be us-ascii
- The version attribute in the notification root element MUST be 1
- The session_id attribute MUST be a random version 4 UUID unique to this session
- The serial attribute must be an unbounded, unsigned positive integer indicating the current version of the repository.
- The notification file MUST contain exactly one ’snapshot’ element for the current repository version.
- If delta elements are included they MUST form a contiguous sequence starting at a revision determined by the publication server, up to the current version of the repository.
- The hash attribute in snapshot and delta elements must be the hexadecimal encoding of the SHA-256 hash of the referenced file. The RP SHOULD verify this hash when the file is retrieved and reject it if it does not match.

3.2.4. Publication Server Initialisation

When the publication server (re-) initialises it MUST generate a new random version 4 UUID to be used as the session_id. Furthermore it MUST then generate a snapshot file for serial number ONE for this new session that includes all currently known published objects that the publication server is responsible for. This snapshot file MUST be made available at a URL that is unique to this session and version, so that it can be cached indefinitely. The format and caching concerns for snapshot files are explained in more detail below in Section 3.3. After the snapshot file has been published the publication server MUST publish a new notification file that contains the new session_id, has serial number ONE, has one reference to the snaphot file that was just published, and that contains no delta references.

3.2.5. Publishing Updates

Whenever the publication server receives updates from a CA it SHOULD generate an update as follows.

The new repository serial MUST be one greater than the current repository serial. A new delta file MUST be generated for this new serial, that contains all the updates, i.e. new, replaced and withdrawn objects, as a single change set. This delta file MUST be made available at a URL that is unique to this session and version, so that it can be cached indefinitely. The format and caching concerns for delta files are explained in more detail below in Section 3.4.

The publication server MUST also generate a new snaphost file for this new serial, that contains all current objects for this new serial. In other words it should include all publish elements found in this update, and it should exclude all previous publish elements for objects that have been withdrawn or updated. As above this new file MUST be made available at a URL that is unique to this session_id and new version before proceeding.

Finally an updated notification file MUST be created by the publication server. This new notification file MUST include a reference to the new snaphot file. The file SHOULD also include available delta files for this and previous updates. However, the server MUST not include more delta files than, when combined, exceed the size of the current snapshot.

The publication server MAY also choose to include fewer delta files if it is found that the efficiency gain in keeping notification files small outweighs the overhead of forcing a small number of relying parties to process full snapshot files. At the time of this writing it is not completely clear what would constitute reasonable parameters to determine this balance. Real world measurements are needed to help this discussion. Possible approaches are:

- The publication server may learn the retrieval distribution of old delta files by RPs over time, and decide to exclude deltas from the point where less than e.g. 0.1% of RPs would retrieve them.
- The server may decide to support deltas only for a limited time, e.g. 6 hours. So that RPs can recover easily from restart or reasonably short outage scenarios, and are only forced to do a full re-sync in case of prolonged outages.

If the publication server is not capable of performing the above for some reason, then it MUST perform a full re-initialisation, as explained above in Section 3.2.4.

3.3. Snapshot File

3.3.1. Purpose

A snapshot is intended to reflect the complete and current contents of the repository. There it MUST contain all objects from the repository current as of the time of the publication.

3.3.2. Cache Concerns

A repository server MAY use caching infrastructure to cache snapshot files and reduce the load of http(s) requests to a central repository server. To support this it is important that snaphot files for a specific session_id and serial have a unique URL. The files themselves reflect the content of the repository at a specific point in time, and for that reason they never change. Aside from space concerns this means that these files MAY therefore be cached indefinitely.

To support RPs that are slow to process old, possibly cached, notification files, the publication server SHOULD ensure that old snapshot files remain available for some time after have last appeared on a notification file. It is RECOMMENDED that these files are kept for at least two times as long as the notification file cache period, i.e. 10 minutes. However, space permitting, the publication server is welcome to keep these files available for longer.

3.3.3. File Format and Validation

Example snapshot file:

```
   <snapshot xmlns="HTTP://www.ripe.net/rpki/rrdp" version="1" session_id="9df4b
597-af9e-4dca-bdda-719cce2c4e28" serial="5932">
     <publish uri="rsync://bandito.ripe.net/repo/671570f06499fbd2d6ab76c4f22566f
e49d5de60.cer">
       MIIFNDCCBBygAwIBAgIBAjANBgkqhkiG9w0BAQsFADANMQswCQYDVQQDEwJUQTAeFw0xNDExM
TMw
       MzU4MjlaFw0xNTExMTMwMzU4MjlaMDMxMTAvBgNVBAMTKDY3MTU3MGYwNjQ5OWZiZDJkNmFiN
zZj
       NGYyMjU2NmZlNDlkNWRlNjAwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDOlUYxD
Pwu
       hqVSG5VXcg96qTYt9aKOH8qV2lAU/jnY1rRl2W5Uoa8RrAIseou8ltLKonMcVulHyoyY+J9Gq
rzN
       45vRSgBaOuvLn6nTuoD0LQsD/m8c/wEmFjQllirxQykLGJLXn1eKdUs/OXGgrAUPzgvkciJds
g69
       6X44deHcbCU0ZQZSLxZBZEqjfgyoYgww9n/hK5Sfkb44LsBK1lESdBSRrTpFizrCxl22ptsH0
eW4
       ek80CV5YgCg4F4u9xlzS2DvB+1X3Nl1vvTZ6TJlpVjIVcvE+sKQ50ntUwWG1+lOJc+twRehhi
CAb
       yHhfaxID4B+7h5Rcpkh1Q1AUMG9JAgMBAAGjggJ3MIICczAdBgNVHQ4EFgQUZxVw8GSZ+9LWq
3bE
       8iVm/knV3mAwHwYDVR0jBBgwFoAUd4IboVLl+9bEbD6VrCsnqRClFNUwDwYDVR0TAQH/BAUwA
wEB
       /zAOBgNVHQ8BAf8EBAMCAQYwRQYIKwYBBQUHAQEEOTA3MDUGCCsGAQUFBzAChilodHRwOi8vY
mFu
       ZGl0by5yaXBlLm5ldC9ycGtpLWNhL3RhL3RhLmNlcjCCATAGCCsGAQUFBwELBIIBIjCCAR4wV
wYI
       KwYBBQUHMAWGS3JzeW5jOi8vYmFuZGl0by5yaXBlLm5ldC9yZXBvLzNhODdhNGIxLTZlMjItN
GE2
       My1hZDBmLTA2ZjgzYWQzY2ExNi9kZWZhdWx0LzCBgwYIKwYBBQUHMAqGd3JzeW5jOi8vYmFuZ
Gl0
       by5yaXBlLm5ldC9yZXBvLzNhODdhNGIxLTZlMjItNGE2My1hZDBmLTA2ZjgzYWQzY2ExNi9kZ
WZh
       dWx0LzY3MTU3MGYwNjQ5OWZiZDJkNmFiNzZjNGYyMjU2NmZlNDlkNWRlNjAubWZ0MD0GCCsGA
QUF
       BzANhjFodHRwOi8vYmFuZGl0by5yaXBlLm5ldC9ycGtpLWNhL25vdGlmeS9ub3RpZnkueG1sM
FsG
       A1UdHwRUMFIwUKBOoEyGSnJzeW5jOi8vYmFuZGl0by5yaXBlLm5ldC9yZXBvLzc3ODIxYmExN
TJl
       NWZiZDZjNDZjM2U5NWFjMmIyN2E5MTBhNTE0ZDUuY3JsMBgGA1UdIAEB/wQOMAwwCgYIKwYBB
QUH
       DgIwHgYIKwYBBQUHAQcBAf8EDzANMAsEAgABMAUDAwDAqDANBgkqhkiG9w0BAQsFAAOCAQEAk
Anl
       E+Fm1r3cmW8EEwhq4Wo37j7qC8ciU/E/zJqptROd8M8+2PDjCF8K7plf/SqYNUWjCk8zQv7Si
ala
       DP3JNI7oWkJ5K9zSU/qPGD8UbrfK5EF4g+++OAsxsOf/qeMVdZ6FlPIUV0wYj2s9w1zz/r16H
FV6
       QO785ajB50foqo/oQ74BSRbrlYkWrM8U45rdSiAMlyr0lHgv0OCqNK6AVR6y9Sp6bBUi7RotZ
5FN
       x0TgBRTA6xp4pjG5FimX1SanMaW1hgYqdc4X5aZ9gPiyqvBcOtFq91WnNTsm5Ox0cPNDCkMPL
AwW
       pHOiFA0PlD0vBPrvTR1hsgfKGd318Qzq+w==
     </publish>
     <publish uri="rsync://bandito.ripe.net/repo/77821ba152e5fbd6c46c3e95ac2b27a
910a514d5.mft">
       MIAGCSqGSIb3DQEHAqCAMIACAQMxDzANBglghkgBZQMEAgEFADCABgsqhkiG9w0BCRABGqCAJ
IAE
       gd0wgdoCAguWGA8yMDE0MTIwMzE4MDgzMloYDzIwMTQxMjA0MTgwODMyWgYJYIZIAWUDBAIBM
IGm
       MFEWLDY3MTU3MGYwNjQ5OWZiZDJkNmFiNzZjNGYyMjU2NmZlNDlkNWRlNjAuY2VyAyEAh0nT6
uSg
       nJQhGAnKgjxb9TDeGu9AEd8QK+GHXYop0U8wURYsNzc4MjFiYTE1MmU1ZmJkNmM0NmMzZTk1Y
WMy
       YjI3YTkxMGE1MTRkNS5jcmwDIQAZ658FmRCmFfxCpTfE8hZN00MnUEdohOiISZflCPbrUwAAA
```

```
AAA
        AKCAMIIEcjCCA1qgAwIBAgICC5gwDQYJKoZIhvcNAQELBQAwDTELMAkGA1UEAxMCVEEwHhcNM
TQx
        MjAzMTgwODMyWhcNMTQxMjEwMTgwODMyWjAzMTEwLwYDVQQDEyhlY2Y0NjhkMDY1MTMyNzFmN
Tkz
        MjZhNjQ2MGZmOTFhYTNiNGU2Njk4MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAk
iYz
        EpnsqHIPNEl/LvJmfZfOYzRlhv0Ewqg/RLi6XsE5dhWi0YAiFLbz0v/PfAjmJJFO6STsXkmc5
Cpp
        PoAl2+Ffx9Zujzy95hCNMqNgPSsqA92eAStLJALlvWrlYgTQEBV/hIjetDOEY/fL49gajyuKg
hOh
        +zgeEUCVhdIArEj/4j5E1vI7flwJjLP8SI36IWlKoz6cd88Gm8bLQRURAfe2lKW0quJk0RHOn
PZk
        babWuiiByoU24DCSy1+TBY4mEK6bi1R0iONqeYfaSUrxvWcDh8V6gNikiB+tfwRxrIOO1RTnK
nlI
        hIe2OC5mkP2gMY5ZUyynJZnS3Or+CY3IcQIDAQABo4IBtDCCAbAwHQYDVR0OBBYEFOz0aNBlE
ycf
        WTJqZGD/kao7TmaYMB8GA1UdIwQYMBaAFHeCG6FS5fvWxGw+lawrJ6kQpRTVMA4GA1UdDwEB/
wQE
        AwIHgDBFBggrBgEFBQcBAQQ5MDcwNQYIKwYBBQUHMAKGKWh0dHA6Ly9iYW5kaXRvLnJpcGUub
mV0
        L3Jwa2ktY2EvdGEvdGEuY2VyMGYGCCsGAQUFBwELBFowWDBWBggrBgEFBQcwC4ZKcnN5bmM6L
y9i
        YW5kaXRvLnJpcGUubmV0L3JlcG8vNzc4MjFiYTE1MmU1ZmJkNmM0NmMzZTk1YWMyYjI3YTkxM
GE1
        MTRkNS5tZnQwWwYDVR0fBFQwUjBQoE6gTIZKcnN5bmM6Ly9iYW5kaXRvLnJpcGUubmV0L3Jlc
G8v
        Nzc4MjFiYTE1MmU1ZmJkNmM0NmMzZTk1YWMyYjI3YTkxMGE1MTRkNS5jcmwwGAYDVR0gAQH/B
A4w
```

```
        DDAKBggrBgEFBQcOAjAhBggrBgEFBQcBBwEB/wQSMBAwBgQCAAEFADAGBAIAAgUAMBUGCCsGA
QUF
        BwEIAQH/BAYwBKACBQAwDQYJKoZIhvcNAQELBQADggEBAAjCpzNzjj7QGhmIG3Elt49cHUJe8
65w
        y2Uq3ZKW2aZgA5It29D07XlsHO8tM0EWVXTxsBbpdkiEnzQ4G8Zx/ZI09vLSJ8ZjzSh42QeMa
Nt6
        6zslilaw9rQcm/5jwxN18BRniwU/oavfRbn36AhfCmpegII/4DTZWjI63wucRrYHTHZm6Zajn
HKU
        DT1viomKZoZZDAUB4oQ7pN/Mw+t1K9F50VKz+9i3tnVhyt5wVaoEn/4sGRAL680A8Su0MKiyc
69t
        3DYqnvSgYtFNiBbHhNYooBpraylh5r7WngBxfm+VJYkSaPxU8T6sSz/Capt+1S2UWJGTcFaZl
251
        bm8nmXcAADGCAawwggGoAgEDgBTs9GjQZRMnH1kyamRg/5GqO05mmDANBglghkgBZQMEAgEFA
KBr
        MBoGCSqGSIb3DQEJAzENBgsqhkiG9w0BCRABGjAcBgkqhkiG9w0BCQUxDxcNMTQxMjAzMTgwO
DMy
        WjAvBgkqhkiG9w0BCQQxIgQgOUbuFjfSw4aMeIgLlDmT5xI7D05/mH6zVETECtMzWb0wDQYJK
oZI
        hvcNAQEBBQAEggEAbhfERg8rgzy0GAIPDKj5kNk+owpm7WnRDiUo+6Y30zfKKjFhh1L+N0Ei7
b6q
        r934eqEoac23wycF/A1e3+d4PoLzvFrm1n9rIia4BaD8GiUle6FEHd5njS7jOt5Kuej64yDFC
Htv
        ipt8tGFik4MpvEmP5EOhZ1cU/sErvlpdEsxQCaLsb6JUbIvoIHnWGXHE54QXkBvlucUSxypRo
qW3
        SnAX0vo0F1YNrSDe05So3pjjSmNHOuFFnxZMja+lIMMWTFylbKQJNpLIrb9a/uarfiL9BrGOD
WqE
        dzQh+k3QkTAUojq+YADL+ixO0eg2zpPm+eEU1F2+bGP2M5rbaUfqngAAAAAAAA==
      </publish>
      <publish uri="rsync://bandito.ripe.net/repo/77821ba152e5fbd6c46c3e95ac2b27a
910a514d5.crl">
        MIIBnzCBiAIBATANBgkqhkiG9w0BAQsFADANMQswCQYDVQQDEwJUQRcNMTQxMjAzMTgwODMyW
hcN
        MTQxMjA0MTgwODMyWjAVMBMCAguXFw0xNDEyMDMxODA4MzJaoDAwLjAfBgNVHSMEGDAWgBR3g
huh
        UuX71sRsPpWsKyepEKUU1TALBgNVHRQEBAICC5YwDQYJKoZIhvcNAQELBQADggEBAFQHr/2id
s7e
        hmfnX+PmyePSN2EM1fBMLwMud6dqyBF42iNa8N0H/jxMAkgm7SS98TUupZg1aIwqxLwGakFS6
VeD
        +zCnCGEeMUlXTpZaICDxMxJuJLBOVbqP2amPxWJ22g0+gTXM9KPAoWlNyAiMaNUP+nawjyfMz
Q4c
        WJjIi1kRHnIhiu9cZwEh9Ns/sC3adPJ8NV6LPpMkQDQvIynxV/fbTf/EwwwRfLy1szGLZSdml
4G0
        gHohkWaosr4R2A7sZOc/PZGtstqpBRTD8RwVJx0pseC6Zp/01WH/FjzNpXahFPgR1QXy3qBGE
HRh
        xA08g0+QiGSz+QX5PPQ2dBkTRIY=
      </publish>
   </snapshot>
```

The following validation rules must be observed when creating or parsing snapshot files:

- A RP MUST NOT process any snapshot file that is not well formed, or which does not conform to the RELAX NG schema outlined in Section 5 of this document.
- The XML namespace MUST be HTTP://www.ripe.net/rpki/rrdp.
- The encoding MUST be us-ascii.
- The version attribute in the notification root element MUST be 1
- The session_id attribute MUST match the expected session_id in the reference in the notification file.
- The serial attribute MUST match the expected serial in the reference in the notification file.
- The hexadecimal encoding of the SHA-256 hash of this snapshot file MUST match the hash attribute in the reference in the notification file.

3.4. Delta File

3.4.1. Purpose

An incremental delta file contains all changes for exactly one serial increment of the publication server. In other words a single delta will typically include all the new objects, updated objects and withdrawn objects that a Certification Authority sent to the publication server. In its simplest form the update could concern only a single object, but it is recommended that CAs send all changes for one of their key pairs: i.e. updated objects as well as a new manifest and CRL as one atomic update message.

3.4.2. Cache Concerns

A repository server MAY use caching infrastructure to cache delta files and reduce the load of http(s) requests to a central repository server. To support this it is important that delta files for a specific session_id and serial have a unique URL. The files themselves reflect the content of the repository at a specific point in time, and for that reason they never change. Aside from space concerns this means that these files MAY therefore be cached indefinitely.

To support RPs that are slow to process old, possibly cached, notification files, the publication server SHOULD ensure that old delta files remain available for some time after have last appeared on a notification file. It is RECOMMENDED that these files are kept for at least two times as long as the notification file cache period, i.e. 10 minutes. However, space permitting, the publication server is welcome to keep these files available for longer.

3.4.3. File Format and Validation

Example snapshot file:

```
   <delta xmlns="HTTP://www.ripe.net/rpki/rrdp" version="1" session_id="9df4b597
-af9e-4dca-bdda-719cce2c4e28" serial="5932">
     <publish uri="rsync://bandito.ripe.net/repo/3a87a4b1-6e22-4a63-ad0f-06f83ad
3ca16/default/671570f06499fbd2d6ab76c4f22566fe49d5de60.mft" hash="226AB8CD3C887A
6EBDDDF317F2FAFC9CF3EFC5D43A86347AC0FEFFE4DC0F607E">
       MIAGCSqGSIb3DQEHAqCAMIACAQMxDzANBglghkgBZQMEAgEFADCABgsqhkiG9w0BCRABGqCAJ
IAE
       gYkwgYYCAguWGA8yMDE0MTIwMzE4MDg0MFoYDzIwMTQxMjA0MTgwODQwWgYJYIZIAWUDBAIBM
FMw
       URYsNjcxNTcwZjA2NDk5ZmJkMmQ2YWI3NmM0ZjIyNTY2ZmU0OWQ1ZGU2MC5jcmwDIQD1h9mcw
KzN
       70He/gIMVxszGJlIXLh/TGzkaNTXxLixmwAAAAAAAKCAMIIFGjCCBAKgAwIBAgICC5YwDQYJK
oZI
       hvcNAQELBQAwMzExMC8GA1UEAxMoNjcxNTcwZjA2NDk5ZmJkMmQ2YWI3NmM0ZjIyNTY2ZmU0O
WQ1
       ZGU2MDAeFw0xNDEyMDMxODA4NDBaFw0xNDEyMTAxODA4NDBaMDMxMTAvBgNVBAMTKDMzMTI1Y
zA4
       NmEwOWEyOTJkYzQxMWI1MzkwODA2ZDA4NTMxM2E1MTQwggEiMA0GCSqGSIb3DQEBAQUAA4IBD
wAw
       ggEKAoIBAQCak1G7912fuezSWPpPqER3aZaP4HShGiIiRWeJLOYKpklAeSO8kRa9R+yDCa9CM
i1B
       lewW/C5Coomb9teVZRg31YkpZlXqqNGg8GVesNMJX3ryuizQ+WRUcgwJoakqWH7wPu5zdfFj4
Cpk
       BpgJF+6TBYwXTjAmxfFP0hm0QLWCLxEd1gBGEdBmOogHqfOZHU95GLjllzsPRmR13kyh7BbYM
ie+
       ENJAqqKBlQvW86xPEDMJKUc0uQDnTPCZQBqFwE1xrgUAuSCfJMUguAE8clsshOFe8ROF9t6NI
BxK
       oxA+PTYCpcthCBtdCFyWn/SLp1pb3gIA6xP9ESGlRHNumPL3AgMBAAGjggI2MIICMjAdBgNVH
Q4E
       FgQUMxJcCGoJopLcQRtTkIBtCFMTpRQwHwYDVR0jBBgwFoAUZxVw8GSZ+9LWq3bE8iVm/knV3
mAw
       DgYDVR0PAQH/BAQDAgeAMGcGCCsGAQUFBwEBBFswWTBXBggrBgEFBQcwAoZLcnN5bmM6Ly9iY
W5k
       aXRvLnJpcGUubmV0L3JlcG8vM2E4N2E0YjEtNmUyMi00YTYzLWFkMGYtMDZmODNhZDNjYTE2L
2Rl
       ZmF1bHQvMIGWBggrBgEFBQcBCwSBiTCBhjCBgwYIKwYBBQUHMAuGd3JzeW5jOi8vYmFuZGl0b
y5y
       aXBlLm5ldC9yZXBvLzNhODdhNGIxLTZlMjItNGE2My1hZDBmLTA2ZjgzYWQzY2ExNi9kZWZhd
Wx0
       LzY3MTU3MGYwNjQ5OWZiZDJkNmFiNzZjNGYyMjU2NmZlNDlkNWRlNjAubWZ0MIGJBgNVHR8Eg
YEw
       fzB9oHugeYZ3cnN5bmM6Ly9iYW5kaXRvLnJpcGUubmV0L3JlcG8vM2E4N2E0YjEtNmUyMi00Y
TYz
       LWFkMGYtMDZmODNhZDNjYTE2L2RlZmF1bHQvNjcxNTcwZjA2NDk5ZmJkMmQ2YWI3NmM0ZjIyN
TY2
       ZmU0OWQ1ZGU2MC5jcmwwGAYDVR0gAQH/BA4wDDAKBggrBgEFBQcOAjAhBggrBgEFBQcBBwEB/
wQS
       MBAwBgQCAAEFADAGBAIAAgUAMBUGCCsGAQUFBwEIAQH/BAYwBKACBQAwDQYJKoZIhvcNAQELB
QAD
       ggEBAEO1dSFDN4wZqtZ0fWo5G0YVN+mtk6tKhHPFwX7ydTofnHZkE2pO7C93XcgPcP4zLUBPt
5kS
       aH+0vcBxs9Vg//58cHRUEHhls9O/XcS8RXCVkNiga+9NB5s4oi0+i/gDU3eOUqE/jqSJAJAS+
Ehi
       tvNh0LuLrW92NrOfbYDk29how3uxK4JucIAQ05i63l7EAeQp3WeI8nVzB9Rfrkv+PSV+57mSX
XtJ
       /jWu3kyjvsxRjeUL3Im2Z1F48zfVF6pVaDT7ib4YbKOyAQTMpi4W6NZwgQskda9B8/0qV/d+2
JrC
       m3Ozm0t2laoH8xKP/OC33bBXLCxUvkVqvB/Y+TUXfAEAADGCAawwggGoAgEDgBQzElwIagmik
txB
       G1OQgG0IUxOlFDANBglghkgBZQMEAgEFAKBrMBoGCSqGSIb3DQEJAzENBgsqhkiG9w0BCRABG
jAc
       BgkqhkiG9w0BCQUxDxcNMTQxMjAzMTgwODQwWjAvBgkqhkiG9w0BCQQxIgQgdNPMbp9lJJHNM
mIz
```

```
       00ff73VkVFYWo2Uf6/b4zIzFZucwDQYJKoZIhvcNAQEBBQAEggEAXHNHm+DUD1s9IQMewvKso
NGi
       fXL2jG3yfuGys5x1aJji3bIKGiU+weHmnP9aoH9UFRLk6pW1wFOS0+6M87UD8cU17w9F10e02
58S
       9p7xHMgbrYqXrX9OucMqiN4M+ThDzyDXnfNAOgw5XNJu9KRndS9vyXS6lcvD7JTOhkyqKsrqH
XlM
       0pX+rYFtrF2RNjB54veooSkcKGojXReLttZbvVKWKwkVg2RJy4tt7MOGU0Q6qa/J5S7O6xvwP
jkY
       yCFvrHm+CgeXoR/3Hg/Rk/NdsK4K1u5dXhRh3KYv4P/hnGSD83aFE9t/DTicvl6SjaXFCtLtJ
lTX
       BqSW7wgZ6OoLxwAAAAAAAA==
     </publish>
     <publish uri="rsync://bandito.ripe.net/repo/3a87a4b1-6e22-4a63-ad0f-06f83ad
3ca16/default/671570f06499fbd2d6ab76c4f22566fe49d5de60.crl" hash="2B551A6C10CCA0
4C174B0CEB3B64652A5534D1385BEAA40A55A68CB06055E6BB">
       MIIBxTCBrgIBATANBgkqhkiG9w0BAQsFADAzMTEwLwYDVQQDEyg2NzE1NzBmMDY0OTlmYmQyZ
DZh
       Yjc2YzRmMjI1NjZmZTQ5ZDVkZTYwFw0xNDEyMDMxODA4NDBaFw0xNDEyMDQxODA4NDBaMBUwE
wIC
       C5UXDTE0MTIwMzE4MDg0MFqgMDAuMB8GA1UdIwQYMBaAFGcVcPBkmfvS1qt2xPIlZv5J1d5gM
AsG
       A1UdFAQEAgILljANBgkqhkiG9w0BAQsFAAOCAQEAIbL+8connmKLeypzs/P6FOHv8elmLp6dF
lId
       SDpZT7p6y9xLZkvuow39XOs6NB1AOA+92uao9hEV1XuEBGP98nsx0frL8HJtKcEn0q5LGqA4Y
eBG
       n28+Ldvlh4DetiKvFpsKW/VYqjRumHcgTdWpESY/f9hH3xW6JCggH5cFGFF/dCsCdGT1v+m53
zf4
       Dlz8KhRDEaok3UMycX9XUWMB5HSwf05Qrha2LIFf66uk6AQQEmV9ZiBq3IdbkdNd90TIVDMvn
SW/
       p9Xygdx8azaE2+hsOc9J7+E2kBuu4isLhvfZmChtFpxIUrljQRD4iUil8/xmB6MAIptoF1Esl
pAI
       aw==
     </publish>
     <withdraw uri="rsync://bandito.ripe.net/repo/3a87a4b1-6e22-4a63-ad0f-06f83a
d3ca16/default/example.roa" hash="2B551A6C10CCA04C174B0CEB3B64652A5534D1385BEAA4
0A55A68CB06055E6BB"/>
   </delta>
```

Note that a formal RELAX NG specification of this file format is included later in this document. A RP MUST NOT process any update notification file that is incomplete or not well formed.

The following validation rules must be observed when creating or parsing snapshot files:

- o A RP MUST NOT process any delta file that is not well formed, or which does not conform to the RELAX NG schema outlined in Section 5 of this document.
- o The XML namespace MUST be HTTP://www.ripe.net/rpki/rrdp.
- o The encoding MUST be us-ascii.
- o The version attribute in the notification root element MUST be 1
- o The session_id attribute MUST be a random version 4 UUID unique to this session
- o The session_id attribute MUST match the expected session_id in the reference in the notification file.
- o The serial attribute MUST match the expected serial in the reference in the notification file.
- o The hexadecimal encoding of the SHA-256 hash of this snapshot file MUST match the hash attribute in the reference in the notification file.
- o A publish element MUST include a hash attribute, if the object is intended to replace another object in the RPKI, and its value MUST be the hexadecimal encoding of the SHA-256 hash of the replaced object. If the published object does not replace another object the hash attribute MUST NOT be included. Note that this is an extension to the publication protocol that is not, yet, reflected in [I-D.ietf-sidr-publication].
- o Similarly a withdraw element MUST contain a hash attribute with the hexadecimal encoding of the SHA-256 hash of the withdrawn object. Including the hashes in this manner allows relying parties to identify specific objects by their hash rather than the URI where they are found.

3.5. SIA for CA certificates

Certificate Authorities that use this delta protocol MUST have an instance of an SIA AccessDescription in addition to the ones defined in [RFC6487],

```
AccessDescription ::= SEQUENCE {
  accessMethod OBJECT IDENTIFIER,
  accessLocation GeneralName }
```

This extension MUST use an accessMethod of id-ad-rpkiNotify, see: [IANA-AD-NUMBERS],

```
id-ad OBJECT IDENTIFIER ::= { id-pkix 48 }
id-ad-rpkiNotify OBJECT IDENTIFIER ::= { id-ad 13 }
```

The accessLocation MUST be a URI [RFC3986], using the 'HTTP' or 'HTTPS' protocol, that will point to the update notification file for the publication server that publishes the products of this CA certificate.

Relying Parties that do not support this delta protocol MUST MUST NOT reject a CA certificate merely because it has an SIA extension containing this new kind of AccessDescription.

4. Relying Party Use

4.1. Full Synchronisation

When a Relying Party first encounters a notification file URI as an SIA of a certificate that it has validated it SHOULD retrieve the notification file and download the latest snapshot to get in sync with the current version of the publication server.

The RP SHOULD reject the snapshot file and raise an operator alert, if its hash does not match the hash listed in the notification file. However, if the RP does not have any prior state it may choose to process this snapshot file anyway. It should be noted that the RPKI objects are protected by object security, so problems or attacks on the publication server or transport can result in witholding or replaying old objects, but it cannot force the RP to accept invalid objects. Using the remaining or old objects for validation is probably better than rejecting everything, since the latter could be used as a denial of service vector on relying parties.

4.2. Processing Deltas

It is RECOMMENDED that the RP notes the URI, session_id and serial number when it first learns about a notification file. The RP MAY then poll the file to discover updates. How frequently the RP does this is largely up to local policy. The polling frequency determines in part what propogation time that a RP is willing to accept between the moment that a change is published in the RPKI, and the moment that those changes are processed and validated. As discussed in Section 3.2.2 there does not seem to be a need to have a propogation time that is below five minutes. Since the publication server infrastructure MAY cache the notification file for up to five minutes a slightly more frequent polling strategy may be useful, however the RP SHOULD NOT poll more frequently than once per minute. More frequent polling would only result in marginal gains in propagation, while causing unnecessary load on the caching infrastructure.

If the RP finds that the session_id has changed, or if it cannot find a contiguous chain of links to delta files from its current serial to publication server's current serial, then it MUST perform a full synchronisation instead of continuing to process deltas.

If the RP finds a contiguous chain of links to delta files from its current serial to the publication server's current serial, and the session_id has not changed, it should download all missing delta files. If any delta file cannot be downloaded, or its hash does not match the hash listed on the notification file, or if no such chain of deltas is availabe, or the session_id has changed, then the RP MUST perform a full synchronisation instead.

New objects found in delta files can be added to the RPs local copy of the repository. However, it is RECOMMENDED that the RP treats object updates and withdraws with some skepticism. A compromised publication server may not have access to the certification authorities' keys, but it can pretend valid objects have been withdrawn. Therefore it may be preferred to use a strategy where local copies of objects are only discarded when the RP is sure that they are no longer relevant, e.g. the CA has explicitly revoked them, removed the objects from a valid manifest that it issued, or they have expired.

5. XML Schema

The following is a RELAX NG compact form schema describing version 1 of this protocol.

```
#
# RelaxNG schema for RPKI Repository Delta Protocol (RRDP).
#

default namespace = "HTTP://www.ripe.net/rpki/rrdp"

version = xsd:positiveInteger   { maxInclusive="1" }
serial  = xsd:nonNegativeInteger
uri     = xsd:anyURI
uuid    = xsd:string            { pattern = "[\-0-9a-fA-F]+" }
hash    = xsd:string            { pattern = "[0-9a-fA-F]+" }
base64  = xsd:base64Binary

# Notification file: lists current snapshots and deltas

start |= element notification {
  attribute version    { version },
  attribute session_id { uuid },
  attribute serial     { serial },
  element snapshot {
    attribute uri  { uri },
    attribute hash { hash }
  },
  element delta {
    attribute serial { serial },
    attribute uri    { uri },
    attribute hash   { hash }
  }*
}

# Snapshot segment: think DNS AXFR.

start |= element snapshot {
  attribute version    { version },
  attribute session_id { uuid },
  attribute serial     { serial },
  element publish      {
    attribute uri { uri },
    base64
  }*
}

# Delta segment: think DNS IXFR.

start |= element delta {
  attribute version    { version },
  attribute session_id { uuid },
  attribute serial     { serial },
  delta_element+
}

delta_element |= element publish  {
  attribute uri  { uri },
```

```
    attribute hash { hash }?,
    base64
  }

  delta_element |= element withdraw {
    attribute uri  { uri },
    attribute hash { hash }
  }

  # Local Variables:
  # indent-tabs-mode: nil
  # comment-start: "# "
  # comment-start-skip: "#[ \t]*"
  # End:
```

6. Security Considerations

TBD

7. IANA Considerations

This document has no actions for IANA.

8. Acknowledgements

TBD

9. References

[I-D.ietf-sidr-publication]
Weiler, S., Sonalker, A. and R. Austein, "A Publication Protocol for the Resource Public Key Infrastructure (RPKI)", Internet-Draft draft-ietf-sidr-publication-05, February 2014.

[IANA-AD-NUMBERS]
"SMI Security for PKIX Access Descriptor", , <http://www.iana.org/assignments/smi-numbers/smi-numbers.xhtml#smi-numbers-1.3.6.1.5.5.7.48>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3986] Berners-Lee, T., Fielding, R. and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

[RFC6481] Huston, G., Loomans, R. and G. Michaelson, "A Profile for Resource Certificate Repository Structure", RFC 6481, February 2012.

[RFC6486] Austein, R., Huston, G., Kent, S. and M. Lepinski, "Manifests for the Resource Public Key Infrastructure (RPKI)", RFC 6486, February 2012.

[RFC6487] Huston, G., Michaelson, G. and R. Loomans, "A Profile for X.509 PKIX Resource Certificates", RFC 6487, February 2012.

[RFC6488] Lepinski, M., Chi, A. and S. Kent, "Signed Object Template for the Resource Public Key Infrastructure (RPKI)", RFC 6488, February 2012.

Authors' Addresses

Tim Bruijnzeels
RIPE NCC

Email: tim@ripe.net

Oleg Muravskiy
RIPE NCC

Email: oleg@ripe.net

Bryan Weber
Cobenian

Email: bryan@cobenian.com

Rob Austein
Dragon Research Labs

Email: sra@hactrn.net

David Mandelberg
BBN Technologies

Email: david@mandelberg.org