Internet Engineering Task Force (IETF)
Request for Comments: 6916
BCP: 182
Category: Best Current Practice
ISSN: 2070-1721

R. Gagliano
Cisco Systems
S. Kent
BBN Technologies
S. Turner
IECA, Inc.

April 2013

Algorithm Agility Procedure for the Resource Public Key Infrastructure (RPKI)

Abstract

This document specifies the process that Certification Authorities (CAs) and Relying Parties (RPs) participating in the Resource Public Key Infrastructure (RPKI) will need to follow to transition to a new (and probably cryptographically stronger) algorithm set. The process is expected to be completed over a timescale of several years. Consequently, no emergency transition is specified. The transition procedure defined in this document supports only a top-down migration (parent migrates before children).

Status of This Memo

This memo documents an Internet Best Current Practice.

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 BCPs is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6916.

Table of Contents

1. Introduction

The Resource Public Key Infrastructure (RPKI) must accommodate transitions between the public keys used by Certification Authorities (CAs). Transitions of this sort are usually termed "key rollover". Planned key rollover will occur regularly throughout the life of the RPKI, as each CA changes its public keys, in a non-coordinated fashion. (By non-coordinated we mean that the time at which each CA elects to change its keys is locally determined, not coordinated across the RPKI.) Moreover, because a key change might be necessitated by suspected private key compromise, one can never assume coordination of these events among all of the CAs in the RPKI. In an emergency key rollover, the old certificate is revoked and a new certificate with a new key is issued. The mechanisms to perform a key rollover in RPKI (either planned or in an emergency), while maintaining the same algorithm suite, are covered in [RFC6489].

This document describes the mechanism to perform a key rollover in the RPKI due to the migration to a new signature algorithm suite. It specifies the process that CAs and Relying Parties (RPs) participating in the RPKI will need to follow to transition to a new (and probably cryptographically stronger) algorithm set. The process is expected to be completed over a timescale of months or years. Consequently, no emergency transition is specified. The transition procedure defined in this document supports only a top-down migration (parent migrates before children).

A signature-algorithm suite encompasses both a signature algorithm (with a specified key size range) and a one-way hash algorithm. It is anticipated that the RPKI will require the adoption of updated key sizes and/or different algorithm suites over time. This document treats the adoption of a new hash algorithm while retaining the current signature algorithm as equivalent to an algorithm migration, and requires the CA to change its key. Migration to a new algorithm suite will be required in order to maintain an acceptable level of cryptographic security and protect the integrity of certificates, Certificate Revocation Lists (CRLs), and signed objects in the RPKI. All of the data structures in the RPKI explicitly identify the signature and hash algorithms being used. However, experience has demonstrated that the ability to represent algorithm IDs is not sufficient to enable migration to new algorithm suites (algorithm agility). One also must ensure that protocols, infrastructure elements, and operational procedures also accommodate the migration from one algorithm suite to another. Algorithm migration is expected to be very infrequent, and it will require the support of a "current" and "next" suite for a prolonged interval, probably several years.

This document defines how entities in the RPKI execute a planned CA key rollover when the algorithm suite changes. The description covers actions by CAs, repository operators, and RPs. It describes the behavior required of both CAs and RPs to make such key changes work in the RPKI context, including how the RPKI repository system is used to support key rollover.

This document does not specify any algorithm suite per se. The RPKI Certificate Policy (CP) [RFC6484] mandates the use of the algorithms defined in [RFC6485] by CAs and RPs. When an algorithm transition is initiated, [RFC6485] MUST be updated (as defined in Section 4.1 of this document) to redefine the required algorithms for compliant RPKI CAs and RPs under the CP. The CP will not change as a side effect of algorithm transition, and thus the policy OID in RPKI certificates will not change.

For each algorithm transition, an additional document (the algorithm transition timetable) MUST be published (as a BCP) to define the dates for each milestone defined in this document. It will define dates for the phase transitions consistent with the descriptions provided in Section 4. It also will describe how the RPKI community will measure the readiness of CAs and RPs to transition to each phase. CAs publish certificates, CRLs, and other signed objects under the new algorithm suite as the transition progresses. This provides visibility into the deployment of the new algorithm suite, enabling the community to evaluate deployment progress. The transition procedure allows CAs to remove old certificates, CRLs, and signed products after the twilight date, which provides the ability to observe and measure the withdrawal of the old algorithm suite. Thus, the phases defined in this document enable the community to evaluate the progress of the transition. The timetable document will also describe procedures to amend the timetable if problems arise in implementing later phases of the transition. It is RECOMMENDED that the timetable document be developed by representatives of the RPKI community, e.g., IANA, Internet Registries, and network operators.

2. Requirements Notation

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

3. Terminology

This document assumes that the reader is familiar with the terms and concepts described in "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile" [RFC5280], "X.509 Extensions for IP Addresses and AS Identifiers" [RFC3779], and "A Profile for Resource Certificate Repository Structure" [RFC6481]. Additional terms and conventions used in examples are provided below.

Algorithm migration: A planned transition from one signature and hash algorithm to a new signature and hash algorithm.

Algorithm Suite A: The "current" algorithm suite used for hashing and signing; used in examples in this document.

Algorithm Suite B: The "next" algorithm suite used for hashing and signing; used in examples in this document.

CA X: The CA that issued CA Y's certificate (i.e., CA Y's parent); used in examples in this document.

CA Y: The non-leaf CA; used in examples in this document.

CA Z: A CA that is a "child" of CA Y; used in examples in this document.

Correspond: Two certificates issued under different algorithm suites correspond to one another if they are issued to the same entity by the same CA and bind identical Internet Number Resources (INRs) to that entity. Two CRLs correspond if they are issued by the same CA and enumerate corresponding certificates. Two signed objects (other than manifests) correspond if they are verified using corresponding end-entity (EE) certificates and they contain the same encapsulated Context Info field. Two manifests correspond if they encompass corresponding certificates, Route Origination Authorizations (ROAs), CRLs, and other signed objects. (The term "equivalent" is used synonymously when referring to such RPKI signed products.)

Leaf CA: A CA that issues only EE certificates.

Non-Leaf CA: A CA that issues certificates to other CAs.

PoP (proof of possession): Execution of a protocol that demonstrates to an issuer that a subject requesting a certificate possesses the private key corresponding to the public key in the certificate request submitted by the subject.

ROA: Route Origination Authorization, as defined in [RFC6482].

Signed product set (also called set or product set): A collection of certificates, signed objects, a CRL and a manifest that are associated by virtue of being verifiable under the same parent CA certificate

4. Key Rollover Steps for Algorithm Migration

The "current" RPKI algorithm suite (Suite A) is defined in the RPKI CP document, by reference to [RFC6485]. When a migration of the RPKI algorithm suite is needed, the first step MUST be an update of [RFC6485] to define the new algorithm suite. The algorithm transition timeline document MUST also be published (as a BCP) to inform the community of the dates selected for milestones in the transition process, as described in Section 4.1.

4.1. Milestones Definition

CA Ready Algorithm B Date: After this date, all non-leaf CAs MUST be ready to process a request from a child CA to issue a certificate under the Algorithm Suite B. All CAs publishing an [RFC6490] Trust Anchor Locator (TAL) for Algorithm Suite A MUST also publish the correspondent TAL for Algorithm Suite B.

CA Go Algorithm B Date: After this date, all CAs MUST have reissued all their signed product sets under Algorithm Suite B.

RP Ready Algorithm B Date: After this date, all RPs MUST be prepared to process signed material issued under Algorithm Suite B.

Twilight Date: After this date, a CA MAY cease issuing signed products under Algorithm Suite A. Also, after this date, an RP MAY cease to validate signed materials issued under Algorithm Suite A.

End-Of-Life (EOL) Date: After this date, Algorithm Suite A MUST be deprecated using the process in Section 10, and all Algorithm Suite A TALs MUST be removed from their publication points.

4.2. Process Overview

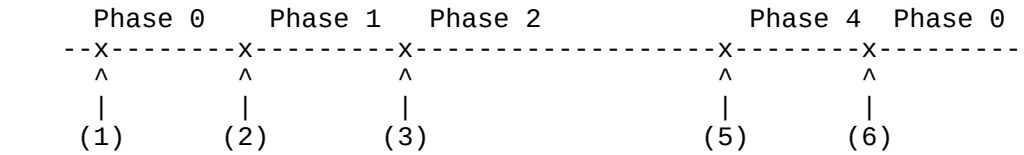
The migration process described in this document involves a series of steps that MUST be executed in chronological order by CAs and RPs. The only milestone at which both CAs and RPs take action at the same time is the EOL Date. Due to the decentralized nature of the RPKI infrastructure, it is expected that an algorithm transition will span several years.

In order to facilitate the transition, CAs will start issuing certificates using Algorithm B in a hierarchical, top-down fashion. In our example, CA Y will issue certificates using Algorithm Suite B only after CA X has started to do so (CA Y Ready Algorithm B Date > CA X Ready Algorithm B Date). This ordered transition avoids the issuance of "mixed" suite CA certificates, e.g., a CA certificate signed using Suite A that contains a key from Suite B. In the RPKI, a CA MUST NOT sign a CA certificate carrying a subject key that corresponds to an algorithm suite that differs from the one used to sign the certificate. (X.509 accommodates such mixed algorithm certificates, but this process avoids using that capability.) A non-top-down transition approach would require the use of such mixed-mode certificates and would lead to exponential growth of the RPKI repository. Also, because the RPKI CP mandates PoP for certificate requests, it is not possible for a CA to request a certificate for Algorithm Suite B until its parent CA supports that suite. (See Section 5 for more details.)

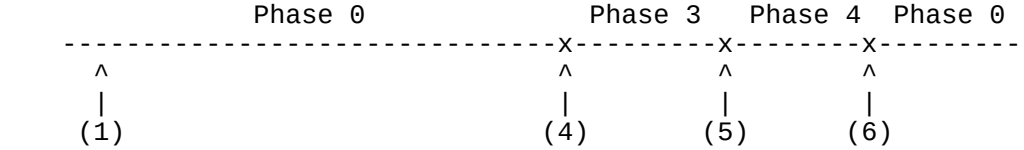
The algorithm agility model described here does not prohibit a CA from issuing an EE certificate with a subject public key from a different algorithm suite, if that certificate is not used to verify repository objects. This exception to the mixed algorithm suite certificate rule is allowed because an EE certificate that is not used to verify repository objects does not interfere with the ability of RPs to download and verify repository content. As noted above, every CA in the RPKI is required to perform a PoP check for the subject public key when issuing a certificate. In general, a subject cannot assume that a CA is capable of supporting a different algorithm. However, if the subject is closely affiliated with the CA, it is reasonable to assume that there are ways for the subject to know whether the CA can support a request to issue an EE certificate containing a specific, different public key algorithm. This document does not specify how a subject can determine whether a CA is capable of issuing a mixed suite EE certificate, because it anticipates that such certificates will be issued only in contexts where the subject and CA are sufficiently closely affiliated (for example, an ISP issuing certificates to devices that it manages).

The following figure gives an overview of the process:

Process for RPKI CAs:



Process for RPKI RPs:



(1) RPKI algorithm document is updated, and the algorithm transition timeline document is issued
(2) CA Ready Algorithm B Date
(3) CA Go Algorithm B Date
(4) RP Ready Algorithm B Date
(5) Twilight Date
(6) End-Of-Life (EOL) Date

Each of these milestones is discussed in the next section when each phase of the transition process is described.

Two situations have been identified that motivate pausing or rolling back the transition process. The first situation arises if the RPKI community is not ready to make the transition. For example, many CAs might not be prepared to issue signed products under Suite B, or many RPs might not be ready to process Suite B products. Under these circumstances, the timetable MUST be reissued, postponing the date for the phase in question and pushing back the dates for later phases. The other situation arises if, during the transition, serious concerns arise about the security of the Suite B algorithms. Such concerns would motivate terminating the transition and rolling back signed products, i.e., reverting to Suite A. In this case, the timetable MUST be republished, and the RPKI algorithm document MUST be superseded. The phase descriptions below allude to these two situations, as appropriate.

4.3. Phase 0

Phase 0 is the steady-state phase of the process; throughout this phase, Algorithm Suite A is the only supported algorithm suite in the RPKI. Phase 0 is also the steady state for the RPKI.

During Phase 0, CAs X, Y, and Z are required to generate signed product sets using only Algorithm Suite A. Also, RPs are required to validate signed product sets issued using only Algorithm Suite A.

The following figure shows an example of the structure of signed objects in the repository, indicating the algorithm suites in use and showing the relationships between three CAs (X, Y, and Z) that form a certification chain. Vertical alignment in the figure indicates objects signed by the same CA using the same private key. The differences in horizontal indentation also represent the use of different publication points for objects signed by different CAs. The characters "|->" are used for visualization purposes for both the signing relationship and the publication point change. For example, the objects CA-Y-Certificate-Algorithm-Suite-A, CA-X-CRL-Algorithm-Suite-A, and CA-X-Signed-Objects-Algorithm-Suite-A are all signed using the private key corresponding to CA-X-Certificate-Algorithm-Suite-A and published at CA X's corresponding publication point.

```
CA-X-Certificate-Algorithm-Suite-A (Cert-XA)
        |-> CA-Y-Certificate-Algorithm-Suite-A (Cert-YA)
                |-> CA-Z-Certificate-Algorithm-Suite-A (Cert-ZA)
                        |-> CA-Z-CRL-Algorithm-Suite-A (CRL-ZA)
                        |-> CA-Z-Signed-Objects-Algorithm-Suite-A
                |-> CA-Y-CRL-Algorithm-Suite-A (CRL-YA)
                |-> CA-Y-Signed-Objects-Algorithm-Suite-A
        |-> CA-X-CRL-Algorithm-Suite-A (CRL-XA)
        |-> CA-X-Signed-Objects-Algorithm-Suite-A
```

Note: Cert-XA represents the certificate for CA X, which is signed using Algorithm Suite A.

4.3.1. Milestone 1

The first milestone initiates the migration process. It updates [RFC6485] with the following definitions for the RPKI:

- Algorithm Suite A
- Algorithm Suite B

Additionally, the new algorithm transition timeline document MUST be published with the following information:

- CA Ready Algorithm B Date
- CA Go Algorithm B Date
- RP Ready Algorithm B Date
- Twilight Date
- EOL Date
- Readiness metrics for CAs and RPs in each phase

Each date specified here is assumed to be at one minute after midnight, UTC. No finer granularity time specification is required or supported.

4.4. Phase 1

Phase 1 starts at the CA Ready Algorithm B Date. During Phase 1, all non-leaf CAs MUST be ready to process a request from a child CA to issue or revoke a certificate using Algorithm Suite B. If it is determined that a substantial number of CAs are not ready, the algorithm transition timeline document MUST be reissued, as noted in Section 4.2. However, CAs that are capable of issuing Suite B certificates may continue to do so, if requested by their child CAs. As this phase does not require any RPs to process signed objects under Suite B, and since Suite B product sets SHOULD be stored at independent publication points, there is no adverse impact on RPs. If the Suite B algorithm is deemed unsuitable, the algorithm transition timeline and the algorithm specification documents MUST be replaced, and Algorithm Suite B MUST be deprecated using the process described in Section 10.

Because the transition will happen using a hierarchical, top-down model, a child CA will be able to issue certificates using Algorithm Suite B only after its parent CA has issued its own. The RPKI provisioning protocol can identify if a parent CA is capable of issuing certificates using Algorithm Suite B and can identify the corresponding algorithm suite in each Certificate Signing Request (CSR; see Section 5). During much of this phase, the Suite B product tree will be incomplete, i.e., not all CAs will have issued products under Suite B. Thus, for production purposes, RPs MUST fetch and validate only Suite A products. Suite B products should be fetched and processed only for testing purposes.

The following figure shows the status of repository entries for the three example CAs during this phase. Two distinct certificate chains are maintained, and CA Z has not yet requested any material using Algorithm Suite B.

```
CA-X-Certificate-Algorithm-Suite-A (Cert-XA)
        |-> CA-Y-Certificate-Algorithm-Suite-A (Cert-YA)
                |-> CA-Z-Certificate-Algorithm-Suite-A (Cert-ZA)
                        |-> CA-Z-CRL-Algorithm-Suite-A (CRL-ZA)
                        |-> CA-Z-Signed-Objects-Algorithm-Suite-A
                |-> CA-Y-CRL-Algorithm-Suite-A (CRL-YA)
                |-> CA-Y-Signed-Objects-Algorithm-Suite-A
        |-> CA-X-CRL-Algorithm-Suite-A (CRL-XA)
        |-> CA-X-Signed-Objects-Algorithm-Suite-A

CA-X-Certificate-Algorithm-Suite-B (Cert-XB)
        |-> CA-Y-Certificate-Algorithm-Suite-B (Cert-YB)
                |-> CA-Y-CRL-Algorithm-Suite-B (CRL-YB)
                |-> CA-Y-Signed-Objects-Algorithm-Suite-B
        |-> CA-X-CRL-Algorithm-Suite-B (CRL-XB)
        |-> CA-X-Signed-Objects-Algorithm-Suite-B
```

4.5. Phase 2

Phase 2 starts at the CA Go Algorithm B Date. At the start of this phase, each signed product set MUST be available using both Algorithm Suite A and Algorithm Suite B. Thus, prior to the start of this phase, every CA MUST ensure that there is a Suite B product corresponding to each Suite A product that the CA has issued. Throughout this phase, each CA MUST maintain this correspondence. During this phase, RPs MUST be prepared to validate sets issued using Algorithm Suite A and MAY be prepared to validate sets issued using the Algorithm Suite B.

If it is determined that a substantial number of CAs are not ready, the algorithm transition timeline document MUST be reissued, as noted in Section 4.2. (Since the processing requirement for RPs here is a MAY, if RPs have problems with Suite B products, this does not require pushing back the Phase 2 milestone, but it does motivate delaying the start of Phase 3.) CAs that are capable of publishing products under Suite B MAY continue to do so. Phase 2, like Phase 1, does not require any RPs to process signed objects under Suite B. Also, Suite B products SHOULD be stored at independent publication points so that there is no adverse impact on RPs that are not prepared to process Suite B products. (See Section 9 for additional details.) If the Suite B algorithm is deemed unsuitable, the

algorithm transition timeline and the algorithm specification documents MUST be replaced, and Algorithm Suite B MUST be deprecated using the process described in Section 10.

It is RECOMMENDED that RPs that can process Algorithm Suite B fetch and validate Suite B products. RPs that are not ready to process Suite B products MUST continue to make use of Suite A products. An RP that elects to validate signed product sets using both Algorithm Suite A and Algorithm Suite B should expect the same results. If there are discrepancies when evaluating corresponding signed product sets, successful validation of either product set is acceptable. A detailed analysis of the validation of multiple instances of signed objects is included in Section 6.

The following figure shows the status of the repository entries for the three example CAs throughout this phase, where all signed objects are available using both algorithm suites.

```
CA-X-Certificate-Algorithm-Suite-A (Cert-XA)
        |-> CA-Y-Certificate-Algorithm-Suite-A (Cert-YA)
                |-> CA-Z-Certificate-Algorithm-Suite-A (Cert-ZA)
                        |-> CA-Z-CRL-Algorithm-Suite-A (CRL-ZA)
                        |-> CA-Z-Signed-Objects-Algorithm-Suite-A
                |-> CA-Y-CRL-Algorithm-Suite-A (CRL-YA)
                |-> CA-Y-Signed-Objects-Algorithm-Suite-A
        |-> CA-X-CRL-Algorithm-Suite-A (CRL-XA)
        |-> CA-X-Signed-Objects-Algorithm-Suite-A

CA-X-Certificate-Algorithm-Suite-B (Cert-XB)
        |-> CA-Y-Certificate-Algorithm-Suite-B (Cert-YB)
                |-> CA-Z-Certificate-Algorithm-Suite-B (Cert-ZB)
                        |-> CA-Z-CRL-Algorithm-Suite-B (CRL-ZB)
                        |-> CA-Z-Signed-Objects-Algorithm-Suite-B
                |-> CA-Y-CRL-Algorithm-Suite-B (CRL-YB)
                |-> CA-Y-Signed-Objects-Algorithm-Suite-B
        |-> CA-X-CRL-Algorithm-Suite-B (CRL-XB)
        |-> CA-X-Signed-Objects-Algorithm-Suite-B
```

4.6. Phase 3

Phase 3 starts at the RP Ready Algorithm B Date. During this phase, all signed product sets are available using both algorithm suites, and all RPs MUST be able to validate them. (The correspondence between Suite A and Suite B products was required for Phase 2 and was maintained throughout that phase. The same requirements apply throughout this phase.) It is RECOMMENDED that, in preparation for Phase 4, RPs retrieve and process Suite B product sets first and

treat them as the preferred product sets for validation throughout this phase. Thus, an RP SHOULD try to validate the sets of signed products retrieved from the Algorithm Suite B repository first.

If a substantial number of RPs are unable to process product sets signed with Suite B, the algorithm transition timeline document MUST be reissued, pushing back the date for this and later milestones, as discussed in Section 4.2. Since the Suite B products SHOULD be published at distinct publication points, RPs that cannot process Suite B products can be expected to revert to the Suite A products that still exist. If the Suite B algorithm is deemed unsuitable, the algorithm transition timeline and the algorithm specification documents MUST be replaced and Algorithm Suite B MUST be deprecated using the process described in Section 10.

There are no changes to the CA behavior throughout this phase.

4.7. Phase 4

Phase 4 starts at the Twilight Date. At that date, Algorithm A is labeled as "old" and the Algorithm B is labeled as "current".

During this phase, all signed product sets MUST be issued using Algorithm Suite B and MAY be issued using Algorithm Suite A. All signed products sets issued using Suite B MUST be published at their corresponding publication points. Signed products sets issued using Suite A might not be available at their corresponding publication points. Every RP MUST validate signed product sets using Suite B. RPs MAY validate signed product sets using Suite A. However, RPs SHOULD NOT assume that the collection of Suite A product sets is complete. Thus, RPs SHOULD make use of only Suite B products sets. (See Section 6 for further details.)

If it is determined that many RPs are not capable of processing the new algorithm suite, the algorithm transition timeline document MUST be reissued, pushing back the date for this and the next milestone. The document MUST require the CA not to remove Suite A product sets if this phase is delayed. If Algorithm Suite B is deemed unsuitable, the algorithm transition timeline and the algorithm specification documents MUST be replaced, Algorithm Suite B MUST be deprecated using the process described in Section 10, and CAs MUST NOT remove Suite A product sets. At this stage, RPs are still capable of processing Suite A signed products, so the RPKI is still viable.

The following figure describes a possible status for the repositories of the example CAs.

```
CA-X-Certificate-Algorithm-Suite-A (Cert-XA)
        |-> CA-Y-Certificate-Algorithm-Suite-A (Cert-YA)
                |-> CA-Y-CRL-Algorithm-Suite-A (CRL-YA)
                |-> CA-Y-Signed-Objects-Algorithm-Suite-A
        |-> CA-X-CRL-Algorithm-Suite-A (CRL-XA)
        |-> CA-X-Signed-Objects-Algorithm-Suite-A

CA-X-Certificate-Algorithm-Suite-B (Cert-XB)
        |-> CA-Y-Certificate-Algorithm-Suite-B (Cert-YB)
                |-> CA-Z-Certificate-Algorithm-Suite-B (Cert-ZB)
                        |-> CA-Z-CRL-Algorithm-Suite-A (CRL-ZB)
                        |-> CA-Z-Signed-Objects-Algorithm-Suite-B
                |-> CA-Y-CRL-Algorithm-Suite-A (CRL-YB)
                |-> CA-Y-Signed-Objects-Algorithm-Suite-B
        |-> CA-X-CRL-Algorithm-Suite-A (CRL-XB)
        |-> CA-X-Signed-Objects-Algorithm-Suite-B
```

4.8. Return to Phase 0

The EOL Date triggers the return to Phase 0 (steady state). At this point, the old algorithm suite, Algorithm Suite A, MUST be deprecated using the process described in Section 10.

This phase closes the loop, as the new algorithm suite (Algorithm Suite B) is now the only required algorithm suite in RPKI. From this point forward, this suite is referred to as Algorithm Suite A.

If it is determined that many RPs are not capable of processing the new algorithm suite, the algorithm transition timeline document MUST be reissued, pushing back the date for this milestone.

5. Support for Multiple Algorithms in the RPKI Provisioning Protocol

The migration described in this document is a top-down process where two synchronization issues need to be solved between child and parent CAs:

- A child CA needs to identify which algorithm suites are supported by its parent CA.
- A child CA needs to signal which algorithm suite should be used by its parent CA to sign a CSR.

The RPKI provisioning protocol [RFC6492] supports multiple algorithms suites by implementing different resource classes for each suite. Several different resource classes also may use the same algorithm suite for different resource sets.

A child CA that wants to identify which algorithm suites are supported by its parent CA MUST perform the following tasks:

1. Establish a provisioning protocol session with its parent CA.

2. Perform a "list" command as described in Section 3.3.1 of [RFC6492].

3. From the Payload in the "list response" resource class, extract the "issuer's certificate" for each class. The algorithm suite for each class will match the algorithm suite used to issue the corresponding "issuer's certificate" (as specified in the SubjectPublicKeyInfo field of that certificate).

A child CA that wants to specify an algorithm suite to its parent CA (e.g., in a certificate request) MUST perform the following tasks:

1. Perform the tasks described above to identify the algorithm suites supported by its parent CA and the resource class corresponding to each suite.

2. Identify the corresponding resource class in the appropriate provisioning protocol command (e.g., "issue" or "revoke").

Upon receipt of a certificate request from a child CA, a parent CA will verify the PoP of the private key. If a child CA requests the issuing of a certificate using an algorithm suite that does not match a resource class, the PoP validation will fail and the request will not be performed.

6. Validation of Multiple Instances of Signed Products

During Phases 1, 2, 3, and 4, two algorithm suites will be valid simultaneously in RPKI. In this section, we describe the RP behavior when validating corresponding signed products using different algorithm suites.

During Phase 1, two corresponding instances MAY be available for each signed product, one signed under Algorithm Suite A and one under Algorithm Suite B. As noted in Section 4.4, in this phase there is a preference for Suite A product sets. All products are available under Suite A, while only some products may be available under Suite B. For production purposes, an RP MAY fetch and validate only Suite A products. Suite B products SHOULD be fetched and validated only for test purposes. When product sets exist under both suites, they should yield equivalent results, to facilitate testing. (It is not possible to directly compare Suite A and Suite B product sets, because certificates, CRLs, and manifests will appear syntactically

different. However, the output of the process, i.e., the ROA payloads -- Autonomous System number and address prefix data -- SHOULD match, modulo timing issues.)

During Phases 2 and 3 of this process, two corresponding instances of all signed products MUST be available to RPs. As noted in Section 4.5, it is RECOMMENDED that Suite B capable RPs fetch and validate Suite B products sets during Phase 2. If an RP encounters validation problems with the Suite B products, it SHOULD revert to using Suite A products. RPs that are Suite B capable MAY fetch both product sets and compare the results (e.g., ROA outputs) for testing.

In Phase 3, all RPs MUST be Suite B capable and MUST fetch Suite B product sets. If an RP encounters problems with Suite B product sets, it can revert to Suite A products. RPs encountering such problems SHOULD contact the relevant repository maintainers (e.g., using the mechanism defined in [RFC6493] to report problems.)

During Phase 4, only Suite B product sets are required to be present for all RPKI entities, as per Section 4.7. Thus, RPs SHOULD retrieve and validate only these product sets. Retrieval of Suite A products sets may yield an incomplete set of signed products and is NOT RECOMMENDED.

7. Revocation

The algorithm migration process mandates the maintenance of two parallel but equivalent certification hierarchies during Phases 2 and 3 of the process. During these phases, a CA MUST revoke and request revocation of certificates consistently under both algorithm suites. When not performing a key rollover operation (as described in Section 8), a CA requesting the revocation of its certificate during these two phases MUST perform that request for both algorithm suites (A and B). A non-leaf CA SHOULD NOT verify that its child CAs comply with this requirement. Note that a CA MUST request revocation of its certificate relative to a specific algorithm suite using the mechanism described in Section 5

During Phase 1, a CA that revokes a certificate under Suite A SHOULD revoke the corresponding certificate under Suite B if that certificate exists. During Phase 4, a CA that revokes a certificate under Suite B SHOULD revoke the corresponding certificate under Suite A if that certificate exists.

During Phase 1, a CA may revoke certificates under Suite B without revoking them under Suite A, since the Suite B products are for test purposes. During Phase 4, a CA may revoke certificates issued under Suite A without revoking them under Suite B, since Suite A products are being deprecated.

8. Key Rollover

Key rollover (without algorithm changes) is effected independently for each algorithm suite and MUST follow the process described in [RFC6489].

9. Repository Structure

The two parallel hierarchies that will exist during the transition process SHOULD have independent publications points. The repository structures for each algorithm suite are described in [RFC6481].

10. Deprecating an Algorithm Suite

To deprecate an algorithm suite, the following process MUST be executed by every CA in the RPKI:

1. Each CA MUST cease issuing certificates under the suite. This means that any request for a CA certificate from a child will be rejected, e.g., sending an "error_response" message with error code "request - no such resource class", as defined in [RFC6492].

2. Each CA MUST cease generating signed products, except the CRL and manifest, under the deprecated algorithm suite.

3. Each CA MUST revoke the EE certificates for all signed products that it has issued under the deprecated algorithm suite. The CA SHOULD delete these products from its publication point to avoid burdening RPs with the need to download and process these products.

4. Each CA MUST revoke all CA certificates that it has issued under the deprecated algorithm suite.

5. Each CA SHOULD remove all CA certificates that it has issued under the deprecated algorithm suite.

6. Each CA that publishes a TAL under the deprecated algorithm suite MUST removed it from the TAL’s publication point.

7. Each CA SHOULD continue to maintain the publication point for the deprecated algorithm suite at least until the CRL nextUpdate. This publication point MUST contain only the CRL and a manifest for that publication point. This behavior provides a window in which RPs may be able to become aware of the revoked status of the signed products that have been deleted.

8. Each RP MUST remove any TALs that is has published under the deprecated algorithm suite.

CAs in the RPKI hierarchy may become aware of the deprecation of the algorithm suite at different times and may execute the procedure above asynchronously relative to one another. Thus, for example, a CA may request revocation of its CA certificate, only to learn that the certificate has already been revoked by the issuing CA. The revocation of a CA certificate makes the CRL and manifest issued under it incapable of validation. The asynchronous execution of this procedure likely will result in transient "inconsistencies" among the publication points associated with the deprecated algorithm suite. However, these inconsistencies should yield "fail-safe" results, i.e., the objects signed under the deprecated suite should be rejected by RPs.

11. Security Considerations

An algorithm transition in RPKI should be a very infrequent event, and it requires wide community consensus. The events that may lead to an algorithm transition may be related to a weakness of the cryptographic strength of the algorithm suite in use by RPKI, which is normal to happen over time. The procedures described in this document mean that it will take years to complete an algorithm transition. During that time, the RPKI system will be vulnerable to any cryptographic weakness that may have triggered this procedure (e.g., a downgrade attack).

This document does not describe an emergency mechanism for algorithm migration. Due to the distributed nature of RPKI and the very large number of CAs and RPs, the authors do not believe it is feasible to effect an emergency algorithm migration procedure.

If a CA does not complete its migration to the new algorithm suite as described in this document (after the EOL of the "old" algorithm suite), its signed product set will no longer be valid. Consequently, the RPKI may, at the end of Phase 4, have a smaller number of valid signed products than before starting the process. Conversely, an RP that does not follow this process will lose the ability to validate signed products issued under the new algorithm

suite. The resulting incomplete view of routing information from the RPKI (as a result of a failure by CAs or RPs to complete the transition) could degrade routing in the public Internet.

12. Acknowledgements

The authors would like to acknowledge the work of the SIDR working group co-chairs (Sandra Murphy, Chris Morrow, and Alexey Melnikov) as well as the contributions given by Geoff Huston, Arturo Servin, Brian Weis, Terry Manderson, Brian Dickson, David Black, and Danny McPherson.

13. Normative References

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

[RFC3779] Lynn, C., Kent, S., and K. Seo, "X.509 Extensions for IP Addresses and AS Identifiers", RFC 3779, June 2004.

[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, May 2008.

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

[RFC6482] Lepinski, M., Kent, S., and D. Kong, "A Profile for Route Origin Authorizations (ROAs)", RFC 6482, February 2012.

[RFC6484] Kent, S., Kong, D., Seo, K., and R. Watro, "Certificate Policy (CP) for the Resource Public Key Infrastructure (RPKI)", BCP 173, RFC 6484, February 2012.

[RFC6485] Huston, G., "The Profile for Algorithms and Key Sizes for Use in the Resource Public Key Infrastructure (RPKI)", RFC 6485, February 2012.

[RFC6489] Huston, G., Michaelson, G., and S. Kent, "Certification Authority (CA) Key Rollover in the Resource Public Key Infrastructure (RPKI)", BCP 174, RFC 6489, February 2012.

[RFC6490] Huston, G., Weiler, S., Michaelson, G., and S. Kent, "Resource Public Key Infrastructure (RPKI) Trust Anchor Locator", RFC 6490, February 2012.

[RFC6492] Huston, G., Loomans, R., Ellacott, B., and R. Austein, "A Protocol for Provisioning Resource Certificates", RFC 6492, February 2012.

[RFC6493] Bush, R., "The Resource Public Key Infrastructure (RPKI) Ghostbusters Record", RFC 6493, February 2012.

Authors' Addresses

Roque Gagliano
Cisco Systems
Avenue des Uttins 5
Rolle 1180
Switzerland

EMail: rogaglia@cisco.com

Stephen Kent
BBN Technologies
10 Moulton St.
Cambridge, MA 02138
USA

EMail: kent@bbn.com

Sean Turner
IECA, Inc.
3057 Nutley Street, Suite 106
Fairfax, VA 22031
USA

EMail: turners@ieca.com