Network Working Group	R.G.M. Gagliano
Internet-Draft	Cisco Systems
Intended status: Standards Track	S.K. Kent
Expires: February 03, 2012	BBN Technologies
	S.T. Turner
	IECA, Inc.
	August 02, 2011

Algorithm Agility Procedure for RPKI.
draft-ietf-sidr-algorithm-agility-03

Abstract

This document specifies the process that Certification Authorities (CAs) and Relying Parties (RP) 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 in a time scale 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).

Status of this Memo

Copyright Notice

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

The RPKI must accommodate transitions between the public keys used by CAs. Transitions of this sort are usually termed “key rollover”. Planned key rollover will occur at regular intervals 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 [I-D.ietf-sidr-keyroll].
This document describes the mechanism to perform a key rollover in RPKI due to the migration to a new signature algorithm suite. 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, 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 migration from one algorithm suite to another. Algorithm migration is expected to be very infrequent, but it also will require support of a “current” and “next” suite for a prolonged interval, probably several years.
This document defines how entities in the RPKI execute (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.

This document does not specify any algorithm suite per se. The RPKI Certificate Policy (CP) [I-D.ietf-sidr-cp] mandates the use of the algorithms defined in [I-D.ietf-sidr-rpki-algs] by CAs and RPs. When an algorithm transition is initiated, [I-D.ietf-sidr-rpki-algs] will be updated (as defined in Section 4.1 of this document) redefining the required algorithm(s) for compliant RPKI CAs and RPs under the CP.

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” [I-D.ietf-sidr-repos-struct]. Additional terms and conventions use din 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, in examples in this document

Algorithm Suite B The “next” algorithm suite used for hashing and signing, used in examples in this document

Algorithm Suite C The “old” 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 this document.

CA Y The CA that is changing keys and/or algorithm suites, used in examples this document

CA Z A CA that is a ”child” of CA Y, used in examples this document

Certificate re-issuance (unilateral) A CA MAY reissue a certificate to a subordinate Subject without the involvement of the Subject. The public key, resource extensions, and most other fields are copied from the current Subject certificate into the next Subject certificate. The Issuer name MAY change, if necessary to reflect the Subject name in the CA certificate under which the reissued certificate will be validated. The validity interval also MAY be changed. This action is defined as a unilateral certificate re-issuance.

Non-Leaf CA A CA that issues certificates to other CAs is a non-leaf CA.

Leaf CA
A leaf CA is a CA that issues only EE certs.

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 submitted by the subject.

Signed Product Set (or 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’s CP document, by reference to [I-D.ietf-sidr-rpki-algs]. When a migration of the RPKI algorithm suite is needed, the first step MUST be an update of the [I-D.ietf-sidr-rpki-algs] document that will include all the information described in Section 4.3.

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 B suite.

CA Go Algorithm B Date - After this date, all (non-leaf) CAs MUST have re-issued all of its signed product set under the Algorithm B suite.

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

Twilight Algorithm B - After this date, a CA MAY cease issuing signed products under the Algorithm A suite. Also, after this date, a RP MAY cease to validate signed materials issued under the Algorithm A suite.

End Of Life (EOL) Algorithm A - After this date every CA MUST NOT generate certificates, CRLs, or other RPKI signed objects under the Algorithm A suite. Also, after this date, no RP SHOULD accept as valid any certificate, CRL or signed object using the Algorithm A suite.

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 moment is the "EOL Algorithm A" date. Due to the decentralized nature of the

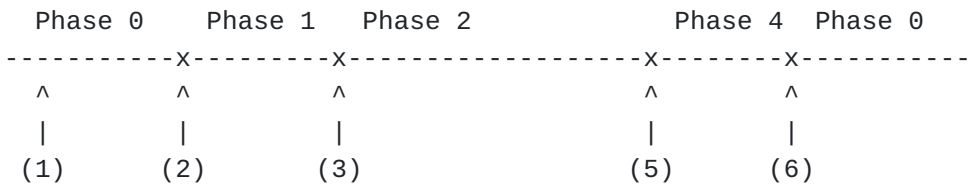
RPKI infrastructure, it is expected that the process will take several months or even years.

In order to facilitate the transition, CAs will start issuing certificates using the Algorithm B in a hierarchical top-down order. In our example, CA Y will issue certificates using the Algorithm B suite 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 issuance of "mixed" suite certificates, e.g., a CA certificate signed using Suite A, containing 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.

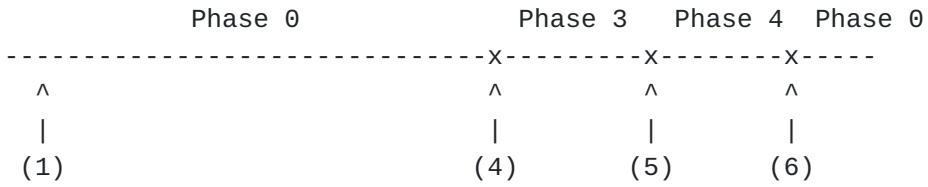
The algorithm agility model described here does not prohibit a CA issuing an EE certificate with a subject public key from a different algorithm suite, if that certificate is not used to verify repository objects. This exception to the mixed algorithm suite certificate rule is allowed because an EE certificate that is not used to verify repository objects does not interfere with the ability of RPs to download and verify repository content. Nonetheless, every CA in the RPKI is required to perform a Proof of Possession (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's algorithm document updated.
(2) CA Ready Algorithm B Date
(3) CA Go Algorithm B Date
(4) RP Ready Algorithm B Date
(5) Twilight Date
(6) End Of Live (EOL) Date

4.3. Phase 0

Phase 0 is the initial phase of the process, throughout this phase the algorithm suite A is the only supported algorithm suite in RPKI.
The first milestone, which will initiate the migration process, is updating the [I-D.ietf-sidr-rpki-algs] document with the following definitions for the RPKI:

- *Algorithm Suite A
- *Algorithm Suite B
- *CA Ready Algorithm B Date
- *CA Go Algorithm B Date
- *RP Ready Algorithm B Date
- *Twilight Date
- *EOL Date

All Dates MUST be represented using the local UTC date-time format specified in [RFC3339].
As an example, during Phase 0, CAs X, Y and Z are required to generate signed product sets using only the Algorithm Suite A. Also, RPs are

required to validate signed product sets issued using only Algorithm Suite A.

```
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 represent the certificate for CA X, that is signed using the algorithm suite A.

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 the Algorithm B suite.
As the transition will happen using a (hierarchic) top-down model, a child CA will be able to issue certificates using the Algorithm B suite 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 the Algorithm Suite B, and can identify the corresponding algorithm suite in each Certificate Signing Request (see Section 5).
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 the Algorithm B suite.

```
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, all signed product sets MUST be available using both Algorithm Suite A and Algorithm Suite B. 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.

An RP that validates all signed product sets using both Algorithm Suite A or Algorithm Suite B, SHOULD expect the same results. However, an object that validates using either Algorithm Suite A or Algorithm Suite B MUST be considered valid. A detailed analysis on the validation of multiple instance 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 using either suite. An object that validates using either Algorithm Suite A or Algorithm Suite B MUST be considered as valid. It is RECOMMENDED that RPs utilize only Suite B for validation throughout this phase, in preparation for Phase 4. There are no changes to the CA behavior throughout this phase.

4.7. Phase 4

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

```
Before Twilight                 -->     After Twilight

Algorithm Suite A ("current")   -->     Algorithm Suite C ("old")
Algorithm Suite B ("new")       -->     Algorithm Suite A ("current")
```

During this phase, all signed product sets MUST be issued using Algorithm Suite A (formerly B) and MAY be issued using Algorithm Suite C (formerly A). All signed products sets issued using Suite A MUST be published at their corresponding publication points, but signed products sets issued using Suite C MAY be published at their corresponding publication points. Also, every RP MUST validate signed

product sets using Suite A but also MAY validate signed product sets using Suite C.
The following figure describe a possible status for the repositories of the example CAs. In this case, CA Z no longer issues signed products using the Algorithm Suite C.

```
CA X-Certificate-Algorithm-Suite-C (Cert-XC)
        |
        |-> CA-Y-Certificate-Algorithm-Suite-C (Cert-YC)
                |-> CA-Y-CRL-Algorithm-Suite-C (CRL-YC)
                |-> CA-Y-Signed-Objects-Algorithm-Suite-C
        |-> CA-X-CRL-Algorithm-Suite-C (CRL-XC)
        |-> CA-X-Signed-Objects-Algorithm-Suite-C

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
```

4.8. Return to Phase 0

Phase 0 starts at the EOL Algorithm Date. At this phase, ALL signed product sets using Algorithm Suite C MUST be considered invalid. CAs MUST neither issue nor publish signed products using Algorithm Suite C. This phase closes the loop as Algorithm Suite A is the only required algorithm suite in RPKI.

5. Multi Algorithm support 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 identify which algorithm suite should be used to sign a Certificate Signing Request (CSR)

The RPKI provisioning protocol [I-D.ietf-sidr-rescerts-provisioning] supports multiple algorithms suites by implementing a 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 [I-D.ietf-sidr-rescerts-provisioning]

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".

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 to identify the resource class for each Algorithm Suite supported by its parent CA (as above).

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 issuing 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 instance 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 instances of the same signed product but signed with different algorithm suites. As a general rule, the validation of signed products using different algorithm suites are independent and the RP MUST NOT keep any relationship between the different hierarchies. During Phase 1 two (corresponding) files for an object MAY be available for each signed product, one signed under Algorithm Suite A and one under Algorithm Suite B. When an RP validates these signed products, if either instance of an object validates, the product is accepted. A failure to validate one instance of a product, under either algorithm Suite MUST NOT cause the RP to reject the other instance of the product. Because both instances of such products MUST contain the same resources, relying on either instance will yield the same outcome. During Phases 2 and 3 of this process, two (corresponding) instances of all signed products MUST be available to RPs. As in Phase 1, when an RP validates these signed products, if either instance validates, the product is accepted. A failure to validate one instance of a product, under either algorithm Suite MUST NOT cause the RP to reject the other instance of the product. Also, as above, if only one instance of a

signed product can be validated, subordinate products issued under the other (non-validated) algorithm suite cannot be used, and thus SHOULD NOT be processed (or even retrieved).
During Phase 4 two (corresponding) files for an object MAY be available for each signed product, one signed under Algorithm Suite A and one under Algorithm Suite C. When an RP validates these signed products, if either instance of an object validates, the product is accepted. A failure to validate one instance of a product, under either algorithm Suite MUST NOT cause the RP to reject the other instance of the product. Because both instances of such products MUST contain the same resources, relying on either instance will yield the same outcome.

7. Revocations

As the algorithm migration process mandates the maintenance of two parallel certificate hierarchies, revocations requests for each algorithm suite MUST be handled independently. A Child CA MUST request revocation of a certificate relative to a specific algorithm suite. During phase 2 and phase 3, the two parallel certificate hierarchies are designed to carry identical information. Consequently, a child CA requesting the revocation of a certificate during these two phases MUST perform that request for both algorithm suites (A and B). A non-leaf CA is NOT required to verify that its child CAs comply with this requirement.

8. Key rollover

Key rollover (without algorithm changes) is effected independently for each algorithm suite and MUST follow the process described in [I-D.ietf-sidr-keyroll].

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 [I-D.ietf-sidr-repos-struct].

10. IANA Considerations

No IANA requirements

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 procedure described in this document will take months or 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.
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 not 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, a 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 info 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 and Chris Morrow) as well as the contributions given by Geoff Huston, Arturo Servin and Brian Weis.

13. References

13.1. Normative References

[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
[RFC2560]	Myers, M., Ankney, R., Malpani, A., Galperin, S. and C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP", RFC 2560, June 1999.
[RFC3339]	Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, July 2002.
[RFC3779]	Lynn, C., Kent, S. and K. Seo, "X.509 Extensions for IP Addresses and AS Identifiers", RFC 3779, June 2004.
[RFC4193]	Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, October 2005.
[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.
[I-D.ietf-sidr-rpki-algs]	Huston, G, "A Profile for Algorithms and Key Sizes for use in the Resource Public Key

	Infrastructure", Internet-Draft draft-ietf-sidr-rpki-algs-04, November 2010.
[I-D.ietf-sidr-keyroll]	Huston, G, Michaelson, G and S Kent, "CA Key Rollover in the RPKI", Internet-Draft draft-ietf-sidr-keyroll-05, December 2010.
[I-D.ietf-sidr-rescerts-provisioning]	Huston, G, Loomans, R, Ellacott, B and R Austein, "A Protocol for Provisioning Resource Certificates", Internet-Draft draft-ietf-sidr-rescerts-provisioning-10, June 2011.
[I-D.ietf-sidr-res-certs]	Huston, G, Michaelson, G and R Loomans, "A Profile for X.509 PKIX Resource Certificates", Internet-Draft draft-ietf-sidr-res-certs-21, December 2010.
[I-D.ietf-sidr-cp]	Kent, S, Kong, D, Seo, K and R Watro, "Certificate Policy (CP) for the Resource PKI (RPKI", Internet-Draft draft-ietf-sidr-cp-17, April 2011.
[I-D.ietf-sidr-repos-struct]	Huston, G, Loomans, R and G Michaelson, "A Profile for Resource Certificate Repository Structure", Internet-Draft draft-ietf-sidr-repos-struct-06, November 2010.

13.2. Informative References

[RFC5781]	Weiler, S., Ward, D. and R. Housley, "The rsync URI Scheme", RFC 5781, February 2010.

Appendix A. Change Log

From 02 to 03:

1. Explicitely named than "mixed" certificates are not allowed for CA certs but may be possible for EE certs that are not used to validate repository objects.

From 01 to 02:

1. Add reference to Multi-Objects validation
2. EOL Data is the only milestone that RP and CA take actions "at the same time".
3. Updated references
4. Editorial

From 00 to 01:

1. Include text to clarify former Suites

2. Include text that documents that an RP that validates an object signed with either suites in Phase 2 MUST consider it as valid

From individual submission to WG item:

1. Change form "laisez faire" to "top-down"

2. Included Multi Algorithm support in the RPKI provisioning protocol

3. Included Validation of multiple instance of signed products

4. Included Revocations

5. Included Key rollover

6. Included Repository structure

7. Included Security Considerations

8. Included Acknowledgements

Authors' Addresses

Roque Gagliano Gagliano Cisco Systems Avenue des Uttins 5 Rolle, 1180 Switzerland EMail: rogaglia@cisco.com

Stephen Kent Kent BBN Technologies 10 Moulton St. Cambridge, MA 02138 USA EMail: kent@bbn.com

Sean Turner Turner IECA, Inc. 3057 Nutley Street, Suite 106 Fairfax, VA 22031 USA EMail: turners@ieca.com