Workgroup: Network Working Group
Internet-Draft:
draft-ietf-sidrops-signed-tal-08
Published: 17 December 2021
Intended Status: Standards Track
Expires: 20 June 2022
Authors: C. Martinez   G. Michaelson   T. Harrison
LACNIC   APNIC   APNIC
T. Bruijnzeels   R. Austein
NLnet Labs   Dragon Research Labs

RPKI Signed Object for Trust Anchor Key

Abstract

A Trust Anchor Locator (TAL) is used by Relying Parties (RPs) in the Resource Public Key Infrastructure (RPKI) to locate and validate a Trust Anchor (TA) Certification Authority (CA) certificate used in RPKI validation. This document defines an RPKI signed object for a Trust Anchor Key (TAK), that can be used by a TA to signal the location(s) of the accompanying CA certificate for the current key to RPs, as well as the successor key and the location(s) of its CA certificate. This object helps to support both planned and unplanned key rolls without impacting RPKI validation.

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", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Overview

A TAL [RFC8630] is used by an RP in the RPKI to locate and validate TA CA certificates used in RPKI validation. However, until now there has been no in-band way of notifying RPs of updates to a TAL. In-band notification means that TAs can be more confident of RPs being aware of key roll operations.

This document defines a new RPKI signed object that can be used to document the location(s) of the TA CA certificate for the current TA key, as well as the value of the successor key and the location(s) of its TA CA certificate. This allows RPs to be notified automatically of such changes, and enables TAs to pre-stage a successor key so that planned and unplanned key rolls can be performed without risking the invalidation of the RPKI tree under the TA. We call this object the Trust Anchor Key (TAK) object.

When RPs are first bootstrapped, they use a TAL to discover the key and location(s) of the CA certificate for a TA. The RP can then retrieve and validate the CA certificate, and subsequently validate the manifest [RFC6486] and CRL published by that TA (section 5 of [RFC6487]). However, before processing any other objects it will first validate the TAK object, if present. If the TAK object indicates that the current key is still valid, then the RP updates its local state to reflect any changes to the certificate location(s) for that current key, and then continues processing as per normal. If the TAK object indicates that the current key has been revoked, then the RP will fetch the successor key, update its local state with that key and its associated certificate location(s), and continue processing using that key.

In a situation where an RP is very outdated, it may have to process a large number of TAK objects in order to reach the current TA key. This is a one-time cost only, though, since once the current TA key is recorded as such by the RP, future operations will start at the certificate location(s) for that TA key, and the previous TAK objects will not need to be retrieved again.

3. TAK Object Definition

The TAK object makes use of the template for RPKI digitally signed objects [RFC6488], which defines a Cryptographic Message Syntax (CMS) [RFC5652] wrapper for the content as well as a generic validation procedure for RPKI signed objects. Therefore, to complete the specification of the TAK object (see Section 4 of [RFC6488]), this document defines:

* The OID (in Section 3.1) that identifies the signed object as being a TAK. (This OID appears within the eContentType in the

encapContentInfo object, as well as the content-type signed attribute in the signerInfo object.)

* The ASN.1 syntax for the TAK eContent, in Section 3.2.

* The additional steps required to validate a TAK, in Section 3.3.

3.1. The TAK Object Content Type

This document requests an OID for the TAK object as follows:

```
id-ct-signed-Tal OBJECT IDENTIFIER ::=
   { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs9(9)
     smime(16) id-smime ct(1) TBD }
```

This OID MUST appear both within the eContentType in the encapContentInfo object, as well as the content-type signed attribute in the signerInfo object (see [RFC6488]).

3.2. The TAK Object eContent

The content of a TAK object is ASN.1 encoded using the Distinguished Encoding Rules (DER) [X.690], and is defined per the module in Appendix A.

3.2.1. TAKey

This structure defines a TA key, similarly to [RFC8630]. It contains a sequence of one or more URIs and a SubjectPublicKeyInfo.

3.2.1.1. certificateURIs

This field is equivalent to the URI section defined in section 2.2 of [RFC8630]. It MUST contain at least one CertificateURI element. Each CertificateURI element contains the IA5String representation of either an rsync URI [RFC5781], or an HTTPS URI [RFC7230].

3.2.1.2. subjectPublicKeyInfo

This field contains a SubjectPublicKeyInfo (section 4.1.2.7 of [RFC5280]) in DER format [X.690].

3.2.2. TAK

3.2.2.1. version

The version number of the TAK object MUST be 0.

3.2.2.2. current

This field contains the TA key of the repository in which the TAK object is published.

3.2.2.3. successor

This field contains the TA key to be used once the current key is revoked.

3.2.2.4. revoked

This field contains a boolean which (if true) indicates that the current TA key should be considered as revoked, in which case RPs should continue processing using the successor TA key.

3.3. TAK Object Validation

To determine whether a TAK object is valid, the RP MUST perform the following checks in addition to those specified in [RFC6488]:

* The eContentType OID matches the OID described in Section 3.1.

* The TAK object appears as the product of a TA CA certificate.

* The TA CA has published only one TAK object in its repository for this key, and this object appears on the manifest as the only entry using the ".tak" extension (see [RFC6481]).

* The EE certificate of this TAK object describes its Internet Number Resources (INRs) using the "inherit" attribute.

* The decoded TAK content conforms to the format defined in Section 3.2.

* The SubjectPublicKeyInfo value of the current TA key in the TAK object matches that of the TA CA certificate used to issue the EE certificate of the TAK object.

If any of these checks does not succeed, the RP MUST ignore the TAK object, and proceed as though it were not listed on the manifest.

4. TAK Object Generation and Publication

A TA MAY choose to use TAK objects to communicate its current and successor keys. If a TA chooses to use TAK objects, then it SHOULD generate and publish TAK objects under each of its keys.

A non-normative guideline for naming this object is that the filename chosen for the TAK object in the publication repository be a value derived from the public key part of the entity's key pair, using the algorithm described for CRLs in section 2.2 of [RFC6481] for generation of filenames. The filename extension of ".tak" MUST be used to denote the object as a TAK.

In order to generate a TAK object, the TA MUST perform the following actions:

* The TA MUST generate a key pair for a "one-time-use" EE certificate to use for the TAK.

* The TA MUST generate a one-time-use EE certificate for the TAK.

* This EE certificate MUST have an SIA extension access description field with an accessMethod OID value of id-ad-signedObject, where the associated accessLocation references the publication point of the TAK as an object URL.

* As described in [RFC6487], an [RFC3779] extension is required in the EE certificate used for this object. However, because the resource set is irrelevant to this object type, this certificate MUST describe its Internet Number Resources (INRs) using the "inherit" attribute, rather than explicit description of a resource set.

* This EE certificate MUST have a "notBefore" time that matches or predates the moment that the TAK will be published.

* This EE certificate MUST have a "notAfter" time that reflects the intended duration for which this TAK will be published. If the EE certificate for a TAK object is expired, it MUST no longer be published, but it MAY be replaced by a newly generated TAK object with equivalent content and an updated "notAfter" time.

* The current TA key for the TAK MUST match that of the TA CA certificate under which the TAK was issued.

5. Relying Party Use

Relying Parties MUST keep a record of the current key for each configured TA, as well as the URI(s) where the CA certificate for this key may be retrieved. This record MAY be bootstrapped by the use of a pre-configured (and unsigned) TAL file [RFC8630], but it MUST be updated with authoritative signed information found in valid TAK objects from subsequent validation runs.

When performing top-down validation, RPs MUST first validate and process the TAK object for its current known key, by performing the following steps:

*A CA certificate is retrieved and validated from the known URIs as described in sections 3 and 4 of [RFC8630].

*The manifest and CRL for this certificate are then validated as described in [RFC6487] and [RFC6486].

*The TAK object, if present, is validated as described in Section 3.3.

If the TAK object indicates that the current key has not been revoked, then the RP updates its local state with the URIs for that key from the TAK object and continues standard top-down validation as described in [RFC6487] using that key.

If the TAK object indicates that the current key has been revoked, then the RP updates its current known key details to be those of the successor key, and then begins top-down validation again using the successor key. The RP repeats this process until it reaches a TA key that either does not publish a TAK object, or publishes a TAK object that indicates that the corresponding current key is not revoked. At that point, it continues standard top-down validation using that key.

If the RP reaches a TAK object that does not include a successor key, but is marked as revoked, then the TA has accidentally revoked its current key. Similarly, if the RP processes the same TAK object twice while attempting to validate a TA, then the TA has accidentally set up a loop among its TAK objects. If the RP encounters either of these scenarios, the RP MUST ignore the TAK objects that it has processed, and proceed as though they were not listed on any of the relevant manifests, so as to allow the TA time to fix the problem. For discussion of how a TA can resolve these problems, see Section 8.

An RP MUST NOT use a successor key for top-down validation when the current key is not revoked, except for the purposes of testing that the new key is working correctly. This allows a TA to publish a successor key for a period of time, allowing RPs to test it, while still being able to rely on RPs using the current key for their production RPKI operations. Once any RP problem reports have been resolved, a TA can then revoke the current key to advance the transition.

6. Maintaining Multiple TA Keys

Although an RP that can process TAK objects will only ever use one key (either the current key, or the successor key if the current key is revoked), an RP that cannot process TAK objects will continue to use the current key even when it is marked as revoked in the TAK. As a result, even when a TA has revoked a key, the TA may have to maintain that key for a period of time alongside the new key in order to ensure continuity of service for older clients.

If a TA has multiple TA keys, then the signed material for these keys MUST be published under different directories in the context of the 'id-ad-caRepository' and 'id-ad-rpkiManifest' Subject Information Access descriptions contained on the CA certificates [RFC6487]. Publishing objects under the same directory is potentially confusing for RPs, and could lead to object invalidity in the event of file name collisions.

However, the CA certificates for each key, and the contents published by each key, MUST be equivalent (except for the TAK object). In other words, for the purposes of RPKI validation, it MUST NOT make a difference which of the keys is used as a starting point.

This means that the IP and AS resources contained on all current CA certificates for the current TA keys MUST be the same. Furthermore, for any delegation of IP and AS resources to a child, the TA MUST have an equivalent CA certificate published under each of its keys. Any updates in delegations MUST be reflected under each of its keys. A TA SHOULD NOT publish any other objects besides a CRL, a Manifest, a single TAK object, and any number of CA certificates for delegation to child CAs.

If a TA uses a single remote publication server for its keys, per [RFC8181], then it MUST include all <publish/> and <withdraw/> PDUs for the products of each of its keys in a single query, in order to ensure that they will reflect the same content at all times.

If a TA uses multiple publication servers, then it is by definition inevitable that the content of different keys will be out of sync at times. In such cases, the TA SHOULD ensure that the duration of these moments are limited to the shortest possible time. Furthermore, the following should be observed:

* In cases where a CA certificate is revoked completely, or replaced by a certificate with a reduced set of resources, these changes will not take effect fully until all the relevant repository publication points have been updated. Given that TA key operations are normally performed infrequently, this is

unlikely to be a problem: if the revocation or shrinking of an issued CA certificate is staged for days/weeks, then experiencing a delay of several minutes for the repository publication points to be updated is fairly insignificant.

* In cases where a CA certificate is replaced by a certificate with an extended set of resources, the TA MUST inform the receiving CA only after all of its repository publication points have been updated. This ensures that the receiving CA will not issue any products that could be invalid if an RP uses a TA key just before the CA certificate was due to be updated.

Finally, note that the publication locations of CA certificates for delegations to child CAs under each key will be different, and therefore the Authority Information Access 'id-ad-caIssuers' values (section 4.8.7 of [RFC6487]) on certificates issued by the child CAs may not be as expected when performing top-down validation, depending on the TA key that is used. However, these values are not critical to top-down validation, so RPs performing such validation MUST NOT reject a certificate simply because this value is not as expected.

7. Performing TA Key Rolls

In this section we will describe how present day RPKI TAs that use only one key pair, and that do not use TAK objects, can change to having a successor key, allowing them to perform both planned and unplanned key rolls.

7.1. Phase 1: Add a TAK for Key 'A'

Before adding a successor key, a Trust Anchor may want to build up operational experience in maintaining a TAK object that describes its current key only. We will refer to this key as key 'A' throughout this section.

The TA will have a TAL file [RFC8630] that contains one or more URIs where the (equivalent) CA certificates for this key 'A' can be retrieved. The TA can now generate a TAK object that sets key 'A' as its current key.

The TA SHOULD publish the CA certificate for key 'A' at one or more new locations not used in the TAL file, and use these new URIs in the TAK object. The TA is free to choose any naming strategy for these locations. As a non-normative suggestion, the TA could use the date that this phase was started as part of the file name or directory where the CA certificate is published.

The TA can now monitor the retrieval of its CA certificates from the URI(s) in the newly published TAK object, relative to the retrieval from the URI(s) listed in its TAL file, to learn the proportion of RPs that can successfully validate and use the TAK object.

7.2. Phase 2: Add a Key 'B'

The TA can now generate a new key pair, key 'B'. This key MUST now be used to create a new CA certificate for this key, and issue equivalent CA certificates for delegations to child CAs, as described in Section 6.

At this point, the TA can also construct a new TAL file [RFC8630] for key 'B', and test locally that the validation outcome for the new key is indeed equivalent to the other current key(s).

When the TA is certain that both keys are equivalent, it MUST issue a new TAK object under key 'A', setting key 'B' as the successor key for key 'A' without revoking key 'A'. It MUST also issue a TAK object under key 'B', with key 'B' as the current key for that object.

7.3. Phase 3: Activate Key 'B'

To roll to key 'B', the TA issues a new TAK object under key 'A' with the revoked field set to true. RPs that process TAK objects will start validating from key 'B' at that point.

The TA must also release a new TAL file for key 'B', as the intended key to be used by RP software. As described above, it SHOULD use a different set of URIs in the TAL compared to the TAK file, so that the TA can learn the proportion of RPs that can successfully validate and use the updated TAK objects.

To support RPs that do not take account of TAK objects, the TA should continue operating key 'A' for a period of time after it has been marked as revoked. The length of that period of time is a local policy matter for that TA: it might operate the key until no clients are attempting to validate using it, for example.

7.4. Phase 4: Remove Key 'A'

The TA SHOULD now publish a long-lived TAK object, CRL and manifest under key 'A', remove all other content, and destroy key 'A'. This way, RP software that uses a TAL for key 'A' can still successfully find keys 'B' and 'C', and in future 'D', 'E', etc.

If access to key 'A' was lost, then there is no way to indicate to clients still using key 'A' that key 'B' should be used from that point onwards. In this instance, the TA will rely on clients updating their local state manually, by way of the new TAL file.

7.5. Unplanned Direction Roll

If key 'B' is compromised, then the TA replaces it as the successor key for key 'A' with another new key ('C'). Since RPs cannot move to key 'B' until key 'A' is marked as revoked, they will begin validation using key 'A', and note that key 'B' has since been replaced as the successor key.

8. Deployment Considerations

Including TAK objects while RPs do not support this standard will result in those RPs rejecting these objects. It is not expected that this will result in the invalidation of any other object under a Trust Anchor.

The mechanism introduced here can only be relied on once a majority of RPs support it. Defining when that moment arrives is something that cannot be established at the time of writing this document. The use of unique URIs for keys in TAK objects, different from those used for the corresponding TAL files, should help TAs understand the proportion of RPs that support this mechanism.

A TA that accidentally revokes a current key via a TAK object without listing a successor key can set the 'revoked' field on that TAK object to false and republish that TAK object. The same is true of a TA that accidentally sets up a loop among its TAK objects, such that a client cannot access a TAK with an unrevoked key and proceed with validation. These are the only instances in which it would be sensible for a TA to publish a TAK object that marked a TA key as not revoked after having already published a TAK object that marked that TA key as revoked. In all other instances, publishing such a TAK object could lead to a split in the RP population for the TA: RPs that had already processed the TAK object with the 'revoked' field set to true would have moved on to the successor key, while RPs that had not would continue to use the current key.

9. Security Considerations

Because a TA key can mark itself as revoked and require clients to start using a new key, an adversary that gains access to a TA's current key and its associated publication servers can essentially take over the TA.

This risk can be mitigated by the use of Hardware Security Modules (HSMs) by TAs, which will guard against theft of a private key, as well as operational processes to guard against (accidental) misuse of the keys in an HSM by operators.

An example where planned rolls are useful is when a TA is using an HSM from vendor X, and they want to migrate to an HSM from vendor Y.

Alternate models of TAL update exist and can be complementary to this mechanism. For example, TAs can liaise directly with validation software developers to include updated and reissued TAL files in new code releases, and use existing code update mechanisms in the RP community to distribute the changes.

10. IANA Considerations

10.1. OID

IANA is asked to add the following to the "RPKI Signed Objects" registry:

Decimal	Description	References
TBD	Trust Anchor Key	[section 3.1]

10.2. File Extension

IANA is asked to add an item for the Signed TAL file extension to the "RPKI Repository Name Scheme" created by [RFC6481] as follows:

Extension	RPKI Object	References
.tak	Trust Anchor Key	[this document]

10.3. Module Identifier

IANA is asked to register an object identifier for one module identifier in the "SMI Security for S/MIME Module Identifier (1.2.840.113549.1.9.16.0)" registry as follows:

Decimal	Description	References
TBD	Trust Anchor Key	[section 3.1]

*Description: RPKISignedTrustAnchorList-2021

*OID: 1.2.840.113549.1.9.16.0.TBD

*Specification: [this document]

11. Implementation Status

NOTE: Please remove this section and the reference to RFC 7942 prior to publication as an RFC.

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in

[RFC7942]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to RFC 7942, "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

11.1. APNIC

*Responsible Organization: Asia-Pacific Network Information Centre

*Location: https://github.com/APNIC-net/rpki-signed-tal-demo

*Description: A proof-of-concept for relying party TAK usage.

*Level of Maturity: This is a proof-of-concept implementation.

*Coverage: This implementation includes all of the features described in this specification. The repository includes a link to various test TALs that can be used for testing TAK scenarios, too.

*Contact Information: Tom Harrison, tomh@apnic.net

12. Revision History

03 - Last draft under Tim's authorship.

04 - First draft with George's authorship. No substantive revisions.

05 - First draft with Tom's authorship. No substantive revisions.

06 - Rob Kisteleki's critique.

07 - Switch to two-key model.

08 - Keepalive.

13. Acknowledgments

The authors wish to thank Martin Hoffmann for a thorough review of this document, and Russ Housley for reviewing the ASN.1 definitions and providing a new module for the TAK object.

14. References

14.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3779] Lynn, C., Kent, S., and K. Seo, "X.509 Extensions for IP Addresses and AS Identifiers", RFC 3779, DOI 10.17487/RFC3779, June 2004, <https://www.rfc-editor.org/info/rfc3779>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/info/rfc5280>.

[RFC5781] Weiler, S., Ward, D., and R. Housley, "The rsync URI Scheme", RFC 5781, DOI 10.17487/RFC5781, February 2010, <https://www.rfc-editor.org/info/rfc5781>.

[RFC6481] Huston, G., Loomans, R., and G. Michaelson, "A Profile for Resource Certificate Repository Structure", RFC 6481, DOI 10.17487/RFC6481, February 2012, <https://www.rfc-editor.org/info/rfc6481>.

[RFC6486] Austein, R., Huston, G., Kent, S., and M. Lepinski, "Manifests for the Resource Public Key Infrastructure (RPKI)", RFC 6486, DOI 10.17487/RFC6486, February 2012, <https://www.rfc-editor.org/info/rfc6486>.

[RFC6487] Huston, G., Michaelson, G., and R. Loomans, "A Profile for X.509 PKIX Resource Certificates", RFC 6487, DOI 10.17487/RFC6487, February 2012, <https://www.rfc-editor.org/info/rfc6487>.

[RFC6488] Lepinski, M., Chi, A., and S. Kent, "Signed Object Template for the Resource Public Key Infrastructure (RPKI)", RFC 6488, DOI 10.17487/RFC6488, February 2012, <https://www.rfc-editor.org/info/rfc6488>.

[RFC7230]
Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014, <https://www.rfc-editor.org/info/rfc7230>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8181] Weiler, S., Sonalker, A., and R. Austein, "A Publication Protocol for the Resource Public Key Infrastructure (RPKI)", RFC 8181, DOI 10.17487/RFC8181, July 2017, <https://www.rfc-editor.org/info/rfc8181>.

[RFC8630] Huston, G., Weiler, S., Michaelson, G., Kent, S., and T. Bruijnzeels, "Resource Public Key Infrastructure (RPKI) Trust Anchor Locator", RFC 8630, DOI 10.17487/RFC8630, August 2019, <https://www.rfc-editor.org/info/rfc8630>.

[X.690] ITU-T Recommendation X.690 (2002) | ISO/IEC 8825-1:2002, "Information technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)", 2002.

14.2. Informative References

[RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, DOI 10.17487/RFC5652, September 2009, <https://www.rfc-editor.org/info/rfc5652>.

[RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <https://www.rfc-editor.org/info/rfc7942>.

Appendix A. ASN.1 Module

This appendix includes the ASN.1 module for the TAK object.

```
<CODE BEGINS>


RPKISignedTrustAnchorList-2021
    { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1)
      pkcs9(9) smime(16) mod(0) TBD }

DEFINITIONS EXPLICIT TAGS ::=
BEGIN

IMPORTS

CONTENT-TYPE
    FROM CryptographicMessageSyntax-2009 -- in [RFC5911]
    { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1)
      pkcs-9(9) smime(16) modules(0) id-mod-cms-2004-02(41) }

SubjectPublicKeyInfo
    FROM PKIX1Explicit-2009 -- in [RFC5912]
    { iso(1) identified-organization(3) dod(6) internet(1)
      security(5) mechanisms(5) pkix(7) id-mod(0)
      id-mod-pkix1-explicit-02(51) } ;

ct-signedTAL CONTENT-TYPE ::=
    { TYPE TAK IDENTIFIED BY
      id-ct-signedTAL }

id-ct-signedTAL OBJECT IDENTIFIER ::= { iso(1) member-body(2)
    us(840) rsadsi(113549) pkcs(1) pkcs9(9) smime(16) ct(1) TBD }

CertificateURI ::= IA5String

TAKey ::= SEQUENCE {
    certificateURIs  SEQUENCE SIZE (1..MAX) OF CertificateURI,
    subjectPublicKeyInfo  SubjectPublicKeyInfo
}

TAK ::= SEQUENCE {
    version     INTEGER DEFAULT 0,
    current     TAKey,
    successor   TAKey OPTIONAL,
    revoked     BOOLEAN
}

END


<CODE ENDS>
```

Authors' Addresses

Carlos Martinez
LACNIC

Email: carlos@lacnic.net
URI: https://www.lacnic.net/

George G. Michaelson
Asia Pacific Network Information Centre
6 Cordelia St
South Brisbane QLD 4101
Australia

Email: ggm@apnic.net

Tom Harrison
Asia Pacific Network Information Centre
6 Cordelia St
South Brisbane QLD 4101
Australia

Email: tomh@apnic.net

Tim Bruijnzeels
NLnet Labs

Email: tim@nlnetlabs.nl
URI: https://www.nlnetlabs.nl/

Rob Austein
Dragon Research Labs

Email: sra@hactrn.net