Network Working Group
Internet-Draft
Intended status: Informational
Expires: 10 March 2026

C. de Kater
N. Rustignoli
SCION Association
S. Hitz
Anapaya Systems
6 September 2025

SCION Control Plane PKI
draft-dekater-scion-pki-10

Abstract

This document presents the trust concept and design of the SCION _Control Plane Public Key Infrastructure (CP-PKI)_. SCION (Scalability, Control, and Isolation On Next-generation networks) is a path-aware, inter-domain network architecture where the Control Plane PKI handles cryptographic material and lays the foundation for the authentication procedures in SCION. It is used by SCION's Control Plane to authenticate and verify path information, and provisions SCION's trust model based on Isolation Domains.

This document describes the trust model behind the SCION Control Plane PKI, including specifications of the different types of certificates and the Trust Root Configuration. It also specifies how to deploy the Control Plane PKI infrastructure.

This document contains new approaches to secure path aware networking. It is not an Internet Standard, has not received any formal review of the IETF, nor was the work developed through the rough consensus process. The approaches offered in this work are offered to the community for its consideration in the further evolution of the Internet.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://scionassociation.github.io/scion-cppki_I-D/draft-dekater-scion-pki.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-dekater-scion-pki/.

Source for this draft and an issue tracker can be found at https://github.com/scionassociation/scion-cppki_I-D.

Status of This Memo

This Internet-Draft will expire on 10 March 2026.

Copyright Notice

Table of Contents

1. Introduction 4
  1.1. Terminology 5
  1.2. Conventions and Definitions 7
  1.3. Trust Model 7
  1.4. Trust Relations within an Isolation Domain 9
    1.4.1. Updates and Trust Resets 9
    1.4.2. Substitutes to Certificate Revocation 9
  1.5. Overview of Certificates, Keys, and Roles 10
  1.6. Trust as a Function 12
    1.6.1. Input 12
    1.6.2. Output 13
2. Certificate Specification 13
  2.1. SCION Control Plane PKI Keys and Certificates - Overview 13
    2.1.1. Trust Hierarchy 13
    2.1.2. Control Plane Root Certificate 14
    2.1.3. Control Plane CA Certificate 14
    2.1.4. Control Plane AS Certificate 14

2.1.5. Voting Certificates 15
2.1.6. Certificates - Formal Overview 16
2.2. Certificate Specification 19
2.2.1. Basic Fields: SCION-Specific Constraints and Conditions 19
2.2.2. Extensions 22
3. Trust Root Configuration Specification 29
3.1. TRC Specification 30
3.1.1. TRC Types and States 30
3.1.2. TRC Format 31
3.1.3. TRC Signature Syntax 39
3.1.4. Certification Path - Trust Anchor Pool 41
3.1.5. TRC Updates 43
3.2. Initial TRC Signing Ceremony 48
4. Deploying the CP PKI - Specifications 49
4.1. Deploying a TRC 49
4.1.1. Base TRC 49
4.1.2. TRC Update 49
4.2. Signing and Verifying Control Plane Messages 50
4.2.1. Signing a Control Plane Message 50
4.2.2. Verifying a Control Plane Message 51
4.3. Creating a New Control Plane AS Certificate 53
5. Security Considerations 53
5.1. Dependency on Certificates 53
5.1.1. Compromise of an ISD 54
5.1.2. Recovery from Compromise 54
5.2. Denial of Service Attacks 55
6. IANA Considerations 55
7. References 56
7.1. Normative References 56
7.2. Informative References 57
Acknowledgments 58
Deployment Testing: SCIONLab 59
Appendix A. Signing Ceremony Initial TRC 59
Ceremony Participants 59
Ceremony Preparations 60
Ceremony Process 61
Phase 1: Certificate Exchange 61
Phase 2: Generation of the TRC Payload 62
Phase 3: TRC Signing 62
Phase 4: TRC Validation 63
Change Log 63
draft-dekater-scion-pki-10 63
draft-dekater-scion-pki-09 63
draft-dekater-scion-pki-08 63
draft-dekater-scion-pki-07 64
draft-dekater-scion-pki-06 64
Authors' Addresses 64

1. Introduction

SCION is a path-aware internetworking routing architecture as described in [RFC9217]. It allows endpoints and applications to select paths across the network to use for traffic, based on trustworthy path properties. SCION is an inter-domain network architecture and is therefore not concerned with intra-domain forwarding.

SCION has been developed with the following goals:

Availability - to provide highly available communication that can send traffic over paths with optimal or required characteristics, quickly handle inter-domain link or router failures (both on the last hop or anywhere along the path) and provide continuity in the presence of adversaries.

Security - to introduce a new approach to inter-domain path security that leverages path awareness in combination with a unique trust model. The goal is to provide higher levels of trust in routing information to prevent traffic hijacking, and enable users to decide where their data travels based on routing information that can be unambiguously attributed to an AS, ensuring that packets are only forwarded along authorized path segments. A particular use case is to enable geofencing.

Scalability - to improve the scalability of the inter-domain control plane and data plane, avoiding existing limitations related to convergence and forwarding table size. The advertising of path segments is separated into a beaconing process within each Isolation Domain (ISD) and between ISDs which incurs minimal overhead and resource requirements on routers.

SCION relies on three main components:

PKI - To achieve scalability and trust, SCION organizes existing ASes into logical groups of independent routing planes called _Isolation Domains (ISDs)_. All ASes in an ISD agree on a set of trust roots called the _Trust Root Configuration (TRC)_ which is a collection of signed root certificates in X.509 v3 format [RFC5280]. The ISD is governed by a set of _core ASes_ which typically manage the trust roots and provide connectivity to other ISDs. This is the basis of the public key infrastructure used for the authentication of messages used by the SCION Control Plane.

Control Plane - performs inter-domain routing by discovering and securely disseminating path information between ASes. The core ASes use Path-segment Construction Beacons (PCBs) to explore intra-ISD paths, or to explore paths across different ISDs.

Data Plane - carries out secure packet forwarding between SCION-enabled ASes over paths selected by endpoints. A SCION border router reuses existing intra-domain infrastructure to communicate to other SCION routers or SCION endpoints within its AS.

This document describes the SCION PKI component used by the Control Plane. It should be read in conjunction with the other components [I-D.dekater-scion-controlplane] and [I-D.dekater-scion-dataplane].

The SCION architecture was initially developed outside of the IETF by ETH Zurich with significant contributions from Anapaya Systems. It is deployed in the Swiss finance sector to provide resilient connectivity between financial institutions. The aim of this document is to document the existing protocol specification as deployed, to encourage interoperability among implementations, and to introduce new concepts that can potentially be further improved to address particular problems with the current Internet architecture.

Note (to be removed before publication): this document, together with the other components [I-D.dekater-scion-controlplane] and [I-D.dekater-scion-dataplane], deprecates [I-D.dekater-panrg-scion-overview].

1.1. Terminology

Control Plane PKI (CP-PKI): The control plane PKI is the public-key infrastructure upon which SCION’s Control Plane relies for the authentication of messages. It is a set of policies, roles, and procedures that are used to manage trust root configurations (TRCs) and certificates.

Autonomous System (AS): An autonomous system is a network under a common administrative control. For example, the network of an Internet service provider or organization can constitute an AS.

Isolation Domain (ISD): In SCION, Autonomous Systems (ASes) are organized into logical groups called isolation domains or ISDs. Each ISD consists of ASes that span an area with a uniform trust environment (i.e., a common jurisdiction). A possible model is for ISDs to be formed along national boundaries or federations of nations.

Core AS: Each isolation domain (ISD) is administered by a set of distinguished autonomous systems (ASes) called core ASes, which are responsible for initiating the path discovery and path construction process known as "beaconing".

Trust Root Configuration (TRC): A Trust Root Configuration or TRC is a signed collection of certificates pertaining to an isolation domain (ISD). TRCs also contain ISD-specific policies.

Authoritative AS: Authoritative ASes are those ASes in an ISD that always have the latest TRCs of the ISD. As a consequence, authoritative ASes also start the announcement of a TRC update.

Base TRC: A base TRC is a trust root configuration (TRC) that other parties trust axiomatically. In other words, trust for a base TRC is assumed, not derived from another cryptographic object. Each ISD MUST create and sign a base TRC when the ISD is established. A base TRC is either the first TRC of the ISD or the result of a trust reset.

TRC Signing Ceremony: The ceremony during which the very first base TRC of an ISD, called the initial TRC, is signed. The initial TRC is a special case of the base TRC where the number of the ISD is assigned.

TRC Update: A _regular_ TRC update is a periodic re-issuance of the TRC where the entities and policies listed in the TRC remain unchanged. A _sensitive_ TRC update is an update that modifies critical aspects of the TRC, such as the set of core ASes. In both cases, the base TRC remains unchanged.

Voting ASes: Those ASes within an ISD that may sign TRC updates. The process of appending a signature to a new TRC is called "casting a vote".

Voting Quorum: The voting quorum is a trust root configuration (TRC) field that indicates the number of votes (signatures) needed on a successor TRC for it to be verifiable. A voting quorum greater than one will thus prevent a single entity from creating a malicious TRC update.

Grace Period: The grace period is an interval during which the previous version of a trust root configuration (TRC) is still considered active after a new version has been published.

Trust Reset: A trust reset is the action of announcing a new base TRC for an existing ISD. A trust reset SHOULD only be triggered after a catastrophic event involving the loss or compromise of several important private keys.

1.2. Conventions and Definitions

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.

1.3. Trust Model

Given the diverse nature of the constituents in the current Internet, an important challenge is how to scale authentication of network elements (such as AS ownership, hop-by-hop routing information, name servers for DNS, and domains for TLS) to the global environment. The roots of trust of currently prevalent public key infrastructure (PKI) models do not scale well to a global environment because (1) mutually distrustful parties cannot agree on a single trust root (monopoly model), and because (2) the security of a plethora of roots of trust is only as strong as its weakest link (oligopoly model) - see also [BARRERA17].

The monopoly model suffers from two main drawbacks: First, all parties must agree on a single root of trust. Secondly, the single root of trust represents a single point of failure, the misuse of which enables the forging of certificates. Its revocation can also result in a kill switch for all the entities it certifies.

The oligopoly model relies on several roots of trust, all equally and completely trusted. However, this is not automatically better: whereas the monopoly model has a single point of failure, the oligopoly model has the drawback of exposing more than one point of failure.

Thus, there is a need for a trust architecture that supports meaningful trust roots in a global environment with inherently distrustful parties. This new trust architecture should provide the following properties:

* Trust agility (see further below);

* Resilience to single root of trust compromise;

* Multilateral governance; and

* Support for policy versioning and updates.

Ideally, the trust architecture allows parties that mutually trust each other to form their own trust "union" or "domain", and to freely decide whether to trust other trust unions (domains) outside their own trust bubble.

To fulfill the above requirements, which in fact apply well to inter-domain networking, SCION introduces the concept of *Isolation Domains*. An Isolation Domain (ISD) is a building block for achieving high availability, scalability, and support for heterogeneous trust. It consists of a logical grouping of ASes that share a uniform trust environment (i.e. a common jurisdiction).

An ISD is administered by one or multiple ASes, called the *voting ASes*. Furthermore, each ISD has a set of ASes that form the ISD core; these are the *core ASes*. The set of core and voting ASes can, but do not necessarily have to, overlap. It is governed by a policy called the *Trust Root Configuration* (TRC), which is negotiated by the ISD core, and which defines the locally scoped roots of trust used to validate bindings between names and public keys.

Authentication in SCION is based on digital certificates that bind identifiers to public keys and carry digital signatures that are verified by roots of trust. SCION allows each ISD to define its own set of trust roots, along with the policy governing their use. Such scoping of trust roots within an ISD improves security as compromise of a private key associated with a trust root cannot be used to forge a certificate outside the ISD. An ISD’s trust roots and policy are encoded in the TRC, which has a version number, a list of public keys that serves as root of trust for various purposes, and policies governing the number of signatures required for performing different types of actions. The TRC serves as a way to bootstrap all authentication within SCION. Additionally, TRC versioning is used to efficiently revoke compromised roots of trust.

The TRC also provides _trust agility_, that is it enables users to select the trust roots used to initiate certificate validation. This implies that users are free to choose an ISD they believe maintains a uncompromised set of trust roots. ISD members can also decide whether to trust other ISDs and thus transparently define trust relationships between parts of the network. The SCION trust model therefore, differs from the one provided by other PKI architectures.

The need for trust agility also means that SCION does not by design provide IP prefix origin validation as provided by RPKI [RFC8210]. RPKI's trust model is currently reliant on the trust roots provided by the five Regional Internet Registries, and therefore outside of the governance of an ISD.

1.4. Trust Relations within an Isolation Domain

As previously mentioned, the Control Plane PKI is organized at an ISD level whereby each ISD can independently specify its own Trust Root Configuration (TRC) and build its own verification chain. Each TRC consists of a collection of signed root certificates, which are used to sign CA certificates, which are in turn used to sign AS certificates. The TRC also includes ISD policies that specify, for example, the TRC's usage, validity, and future updates. The so-called *base TRC* constitutes the ISD's trust anchor which is signed during a signing ceremony by the voting ASes and then distributed throughout the ISD.

1.4.1. Updates and Trust Resets

There are two types of TRC updates: regular and sensitive. A *regular TRC update* is a periodic re-issuance of the TRC where the entities and policies listed in the TRC remain unchanged, whereas a *sensitive TRC update* is an update that modifies critical aspects of the TRC, such as the set of core ASes. In both cases the base TRC remains unchanged.

If the ISD's TRC has been compromised, it is necessary for an ISD to re-establish the trust root. This is possible with a process called *trust reset* (if permitted by the ISD's trust policy). In this case, a new base TRC is created.

1.4.2. Substitutes to Certificate Revocation

The Control Plane PKI does not explicitly support certificate revocation. Instead it relies on the two mechanisms described above and on short-lived certificates. This approach constitutes an attractive alternative to a revocation system for the following reasons:

* Both short-lived certificates and revocation lists must be signed by a CA. Instead of periodically signing a new revocation list, the CA can simply re-issue all the non-revoked certificates. Although the overhead of signing multiple certificates is greater than that of signing a single revocation list, the overall complexity of the system is reduced. In the Control Plane PKI the number of certificates that each CA must renew is manageable as it is limited to at most the number of ASes within an ISD.

* Even with a revocation system, a compromised key cannot be instantaneously revoked. Through their validity period, both short-lived certificates and revocation lists implicitly define an attack window (i.e. a period during which an attacker who managed to compromise a key could use it before it becomes invalid). In both cases, the CA must consider a tradeoff between efficiency and security when picking this validity period.

1.5. Overview of Certificates, Keys, and Roles

The base TRC constitutes the root of trust within an ISD. Figure 1 provides a view of the trust chain within an ISD, based on its TRC. For detailed descriptions, please refer to Section 2 and Section 3.

```
                +----------------------------------------+
                |                 TRC 2                  |
                | +------------------------------------+ |
                | |- Version       - Core ASes         | |
 +--------+     | |- ID            - Description       | |    +--------+
 | TRC 1  |     | |- Validity      - No Trust Reset    | |    | TRC 3  |
 | (Base  |---->| |- Grace Period  - Voting Quorum     | |--->|        |
 |  TRC)  |     | |- ...                               | |    |        |
 +--------+     | +------------------------------------+ |    +--------+
                |                                        |
                | +----------------+  +----------------+ |
                | | Regular Voting |  |Sensitive Voting| |
                | |  Certificate   |  |  Certificate   | |
                | +----------------+  +----------------+ |
                |                                        |
                | +----------------+  +----------------+ |
                | |     Votes      |  |   Signatures   | |
                | +----------------+  +----------------+ |
                |                                        |
                | +------------------------------------+ |
                | |        CP Root Certificates        | |
                | +------+---------------------+-------+ |
                |        |                     |         |
                +----------------------------------------+
                         |                     |
                         |                     |
                         v                     v
                   +-----------+         +-----------+
                   |   CP CA   |         |   CP CA   |
                   |Certificate|         |Certificate|
                   +-+-------+-+         +-----+-----+
                     |       |                 |
                     |       |                 |
                     v       v                 v
            +-----------+ +-----------+  +-----------+
            |   CP AS   | |   CP AS   |  |   CP AS   |
            |Certificate| |Certificate|  |Certificate|
            +-----------+ +-----------+  +-----------+
```

Figure 1: Chain of trust within an ISD

All certificates used in the Control plane PKI are in X.509 v3 format [RFC5280] and additionally the TRC contains self-signed certificates instead of plain public keys. Self-signed certificates have the following advantages over plain public keys: (1) They make the binding between name and public key explicit; and (2) the binding is signed to prove possession of the corresponding private key.

All ASes in SCION have the task to sign and verify control plane messages. However, certain ASes have additional roles:

* *Core ASes*: Core ASes are a distinct set of ASes in the SCION Control Plane. For each ISD, the core ASes are listed in the TRC. Each core AS in an ISD has links to other core ASes (in the same or in different ISDs).

* *Certification authorities (CAs)*: CAs are responsible for issuing AS certificates to other ASes and/or themselves.

* *Voting ASes*: Only certain ASes within an ISD may sign TRC updates. The process of appending a signature to a new TRC is called "casting a vote", and the designated ASes that hold the private keys to sign a TRC update are "voting ASes".

* *Authoritative ASes*: Authoritative ASes are those ASes in an ISD that always have the latest TRCs of the ISD. They start the announcement of a TRC update.

1.6. Trust as a Function

The Control Plane PKI can be seen as a function that transforms potential distrust among different parties into a mutually accepted trust contract including a trust update and reset policy as well as certificates used for authentication procedures in SCION’s Control Plane.

For the function to work, it is not necessary that the ASes of the ISD all trust each other. However, all ASes MUST trust the ISD’s core ASes, authoritative ASes, voting ASes, as well as its CA(s). These trusted parties negotiate the ISD trust contract in a "bootstrapping of trust" ceremony, where cryptographic material is exchanged and the ISD’s trust anchor (the initial Trust Root Configuration) is created and signed.

1.6.1. Input

Prior to the ceremony, the trusted parties MUST decide about the validity period of the TRC as well as the number of votes required to update a TRC. They MUST also bring the required keys and certificates, the so-called root and voting keys/certificates.

The trusted parties require an ISD number, whose numbering scheme is described in [I-D.dekater-scion-controlplane] section ISD Numbers, and allocation in [ISD-AS-assignments].

1.6.2. Output

The output of the bootstrapping of trust ceremony, or the trust "function", are the ISD’s initial Trust Root Configuration as well as mutually trusted and accepted CA and AS certificates - the latter are used to verify control plane messages. Together with the ISD’s control plane root certificates, the CA and AS certificates build the ISD’s trust and verification chain.

2. Certificate Specification

This section provides a detailed specification of all certificates used by the Control Plane PKI.

2.1. SCION Control Plane PKI Keys and Certificates - Overview

There are three types of Control Plane (CP) certificates: root certificates, CA certificates, and AS certificates. Together, they build a chain of trust that is anchored in the Trust Root Configuration (TRC) file of the respective Isolation Domain (ISD). Additionally, there are regular and sensitive voting certificates which define the keys to cast votes in a regular or sensitive TRC update.

All certificates in the Control Plane PKI are in X.509 v3 format [RFC5280].

2.1.1. Trust Hierarchy

The trust is anchored in the TRC for each ISD. The trust root is axiomatic: All trust derived from this anchor relies on all parties transitively trusting the TRC.

The trust hierarchy looks like this:

```
TRC
── Regular Voting Certificates
     └─── TRC (next version, regular update)
── Sensitive Voting Certificates
     └─── TRC (next version, sensitive update)
── CP Root Certificates
     └─── CP CA Certificates
          └─── CP AS Certificates
```

2.1.2. Control Plane Root Certificate

The private key of the Control Plane root certificate is used to sign Control Plane CA certificates. Consequently, the public key of the Control Plane Root certificate is used to verify Control Plane CA certificates, i.e. root certificates determine which ASes act as a CA in an ISD.

In X.509 terms, Control Plane root certificates are _self-signed_ CA certificates. That is, issuer and subject of the certificate are the same entity, and the public key in the root certificate can be used to verify the root certificate’s signature. The public key of the Control Plane root certificate and proof of ownership of the private key are embedded in the TRC of an ISD, via the self-signed Control Plane root certificate. This facilitates the bootstrapping of trust within an ISD, and marks the Control Plane root certificates as the starting point of an ISD’s certificate verification path.

The RECOMMENDED *maximum validity period* of a Control Plane root certificate is 1 year.

Note: The TRC of each ISD contains a trusted set of Control Plane root certificates, and this set builds the root of each ISD’s verification path. For more information on the selection of this trusted set of root certificates, see Section 3.

2.1.3. Control Plane CA Certificate

The private key of the Control Plane CA is used to sign Control Plane AS certificates. Consequently, Control Plane CA certificates holding the public key of the Control Plane CA are used to verify Control Plane AS certificates.

The public key needed to verify the CA certificate is in a Control Plane root certificate. CA certificates do not bundle the root certificate needed to verify them. In order to verify a CA certificate, a pool of root certificates must first be extracted from one or more active TRCs (as described in Section 4.2).

The RECOMMENDED *maximum validity period* of a Control Plane CA certificate is 11 days.

2.1.4. Control Plane AS Certificate

SCION ASes sign control plane messages, such as Path Construction Beacons, with their AS private key. Consequently, Control Plane AS certificates holding the corresponding AS public key are required to verify control plane messages.

In X.509 terms, Control Plane AS certificates are end-entity certificates. That is, they cannot be used to verify other certificates.

The RECOMMENDED *maximum validity period* of a CP AS certificate is 3 days.

2.1.5. Voting Certificates

There are two types of voting certificates: the (1) regular voting certificates and the (2) sensitive voting certificates. They contain the public keys associated with the private keys that MAY cast votes in the TRC update process. Voting certificates are X.509-style certificates.

Regular and sensitive voting certificates are used to verify regular and sensitive TRC updates, respectively, and are embedded in the TRC.

2.1.5.1. Regular Voting Certificate

Regular voting certificates state which keys MAY cast votes in a regular update. In X.509 terms, regular voting certificates are self-signed end-entity certificates. This means that the issuer and subject of a regular voting certificate are the same entity, and the public key within the certificate can be used to verify the certificate’s signature. However, a regular voting certificate cannot be used to verify other certificates.

The RECOMMENDED *maximum validity period* of a regular voting certificate is 1 year.

2.1.5.2. Sensitive Voting Certificate

Sensitive voting certificates specify which keys MAY cast votes in a sensitive update. In X.509 terms, sensitive voting certificates are self-signed end-entity certificates. This means that the issuer and subject of a sensitive voting certificate are the same entity, and the public key within the certificate can be used to verify the certificate’s signature. However, a sensitive voting certificate cannot be used to verify other certificates.

The RECOMMENDED *maximum validity period* of a sensitive voting certificate is 5 years.

Note:

Both SCION Control Plane root certificates and Control Plane CA certificates are in fact CA certificates. That is, they can both be used to verify other certificates.

One important difference between both certificate types lies in their validity period: A SCION Control Plane root certificate has a RECOMMENDED maximum validity period of one year, whereas the RECOMMENDED maximum validity period of a SCION Control Plane CA certificate is 11 days. This is because a root certificate is part of the TRC of an ISD, which itself also has a RECOMMENDED maximum validity period of one year (see Table 1 below). This ensures that the TRC need not be updated all the time and is thus relatively stable.

The SCION root private key and public key/certificate are used to sign and verify the Control Plane CA certificates, respectively. The control plane CA certificates are explicitly NOT part of the TRC, for reasons of security. The Control Plane CA certificates are used to verify the Control Plane AS certificates, which in turn are used to verify control plane messages. Routing is made more secure if both the SCION Control Plane CA and AS certificates can be renewed on a very regular basis. If the control plane CA and AS certificates were part of the TRC, then the TRC would have to be updated constantly, which is undesirable.

2.1.6. Certificates - Formal Overview

Table 1 and Table 2 below provide an overview of the different types of key pairs and certificates in the control plane PKI.

Name	Notation (1)	Used to verify/sign
Sensitive voting key	K_sens	TRC updates (sensitive)
Regular voting key	K_reg	TRC updates (regular)
CP root key	K_root	CP CA certificates
CP CA key	K_CA	CP AS certificates
CP AS key	K_AS	CP messages, path segments

Table 1: Key chain

(1) K_x = PK_x + SK_x, where x = certificate type, PK_x = public key, and SK_x = private key

Name	Notation	Signed with	Contains	Validity (2)
TRC (trust root conf.)	TRC	SK_sens, SK_reg (1)	C_root, C_sens, C_reg (1)	1 year
Sensitive voting cert.	C_sens	SK_sens	PK_sens	5 years
Regular voting cert.	C_reg	SK_reg	PK_reg	1 year
CP root certificate	C_root	SK_root	PK_root	1 year
CP CA certificate	C_CA	SK_root	PK_CA	11 days (3)
CP AS certificate	C_AS	SK_CA	PK_AS	3 days

Table 2: Certificates

(1) Multiple signatures and certificates of each type MAY be included in a TRC.
(2) Recommended maximum validity period.
(3) A validity of 11 days with 4 days overlap between two CA certificates is RECOMMENDED to enable the best possible operational procedures when performing a CA certificate rollover.

Figure 2 shows the content of a base/initial TRC, and the relationship between a TRC and the five types of certificates. The initial signatures are replaced by those of the Regular Voting Certificates with the first regular update to the base TRC.

```
+-----------------------------------------------+
|                    TRC 1                      |
|                (base/initial)                 |
| +-------------------------------------------+ |
| | - Version          - Core ASes            | |
| | - ID               - Description          | |
| | - Validity         - No Trust Reset       | |
| | - Grace Period     - Voting Quorum        | |
| | - ...                                     | |
```

```
   | +------------------------------------------+ |
   |                                              |
   | +-------------------+ +--------------------+ |
   | |        Votes      | |   Regular Voting   | |
   | |  (cert. indices)  | |    Certificates    | |
   | |                   | |  +-----+ +-----+   | |
   | |       (empty)     | |  | (1) | | (2) |   | |
   | |                   | |  |  C  | |  C  |   | |
   | |                   | |  | reg | | reg |   | |
   | |                   | |  +-----+ +-----+   | |
   | +-------------------+ +--------------------+ |
   |                                              |
   | +--------------------+ +-------------------+ |
   | |     Signatures     | | Sensitive Voting  | |
   | | +----------------+ | |    Certificates   | |
   | | | 73 A9 4E A0 ...| | |                   | |
   | | +----------------+ | | +-----+ +-----+   | |
   | |         ...        | | | (3) | | (4) |   | |
   | | +----------------+ | | |  C  | |  C  |   | |
   | | | 53 B7 7C 98 ...| | | | sens| | sens|   | |
   | | +----------------+ | | +-----+ +-----+   | |
   | +--------------------+ +-------------------+ |
   |                                              |
   | +------------------------------------------+ |
   | |          CP Root Certificates            | |
   | |                                          | |
   | | +-----+ +-----+ +-----+ +-----+          | |
   | | | (5) | | (6) | | (7) | | (8) |          | |
   | | |  C  | |  C  | |  C  | |  C  |          | |
   | | | root| | root| | root| | root| ...      | |
   | | +-----+ +--+--+ +-----+ +--+--+          | |
   | +------------+---------------+-------------+ |
   +--------------+---------------+---------------+
                  |               |
                  v               v
         +-----------+         +-----------+
         |   CP CA   |         |   CP CA   |
         |Certificate|         |Certificate|
         +-----+-----+         +-----+-----+
               |                     |
               v                     v
         +-----------+         +-----------+
         |   CP AS   |         |   CP AS   |
         |Certificate|         |Certificate|
         +-----------+         +-----------+
```

Figure 2: TRC update chain and the different types of associated certificates. Arrows show how signatures are verified; in other words, they indicate that a public key contained in a certificate or TRC can be used to verify the authenticity of another item.

2.2. Certificate Specification

Whilst the certificates used in the Control Plane PKI are X.509 v3 certificates, the SCION specification is more restrictive. This section defines these additional constraints and conditions in comparison to [RFC5280].

2.2.1. Basic Fields: SCION-Specific Constraints and Conditions

The described fields of the Control Plane PKI certificates are relevant for each certificate regardless of the certificate type. For detailed descriptions of the full generic format of X.509 v3 certificates, see [RFC5280] and [X.509] clause 7.2.

TBSCertificate sequence: Contains information associated with the subject of the certificate and the CA that issued it. It includes the following fields:

* version field: Describes the version of the encoded certificate. It MUST be set to "v3" (as extensions are REQUIRED in SCION).

* serialNumber field: A positive integer assigned by the CA to each certificate. It MUST be unique for each certificate issued by a given CA.

* signature field: Contains the identifier for the algorithm used by the CA to sign the certificate.

  - *SCION constraints*: Currently, SCION only supports the ECDSA signature algorithm. The details can be found at: Section 2.2.1.1.

  - *Additional conditions and remarks*: As a consequence, the parameters field in the AlgorithmIdentifier sequence MUST NOT be used.

* issuer field: Contains the distinguished name (DN) of the entity that has issued and signed the certificate (usually a CA).

  - *SCION constraints*:

    o This field MUST be non-empty.

o SCION implementations MUST ONLY use the “UTF8String” value type for all attributes (including the SCION-specific attribute ISD-AS number).

- *Additional conditions and remarks*: All SCION implementations MUST support the additional SCION-specific attribute ISD-AS number. For details, see Section 2.2.1.2 and Section 2.2.1.2.1.

* validity field: Defines the validity period of the certificate.

  - *SCION constraints*: All certificates MUST have a well-defined expiration date. Certificates with a generalized time value are not valid and MUST be rejected.

  - *Additional conditions and remarks*: SCION recommends a specific maximum validity period for each type of certificate. For details, see Section 2.1.6. SCION implementations SHOULD adopt these values.

* subject field: Defines the entity that owns the certificate.

  - *SCION constraints*:

    o This field MUST be non-empty.

    o SCION implementations MUST ONLY use the “UTF8String” value type for all attributes (including the SCION-specific attribute ISD-AS number).

  - *Additional conditions and remarks*: The subject field is specified in the same way as the issuer field. For details, see Section 2.2.1.2 and Section 2.2.1.2.1.

* subjectPublicKeyInfo field: Carries the public key of the certificate’s subject (the entity that owns the certificate, as defined in the subject field). The subjectPublicKeyInfo field also identifies which algorithm to use with the key.

  - *SCION constraints*: For constraints regarding the algorithm, see the signature field.

* issuerUniqueID field: it MUST NOT be used in SCION.

* subjectUniqueID field: it MUST NOT be used in SCION.

* extensions sequence: Defines the extensions of the certificate. For a description of all extensions used in SCION, see Section 2.2.2.

2.2.1.1. signature Field - Additional Information

The signature field contains information about the signature algorithm. Current implementations use the ECDSA signature algorithm defined in [X9.62].

The Object Identifiers (OIDs) for ECDSA are defined as ecdsa-with-SHA256, ecdsa-with-SHA384, and ecdsa-with-SHA512 in [RFC5758].

SCION implementations MUST include support for the ECDSA curves below. Other algorithms or curves MAY be used in the future.

* NIST P-256 (NISTFIPS186-4, section D.1.2.3) (named secp256r1 in [RFC5480])

* NIST P-384 (NISTFIPS186-4, section D.1.2.4) (named secp384r1 in [RFC5480])

* NIST P-521 (NISTFIPS186-4, section D.1.2.5) (named secp521r1 in [RFC5480])

The OIDs for the above curves are specified in section 2.1.1.1 of [RFC5480].

The appropriate hash size to use when producing a signature with an ECDSA key is:

* ECDSA with SHA-256, for a P-256 signing key

* ECDSA with SHA-384, for a P-384 signing key

* ECDSA with SHA-512, for a P-521 signing key

2.2.1.2. issuer Field - Additional Information

The issuer field contains the distinguished name (DN) of the CA that created the certificate. [RFC5280], section 4.1.2.4, describes the field’s syntax and attributes. In addition to these attributes, SCION implementations MUST also support the SCION-specific attribute ISD-AS number. See Section 2.2.1.2.1.

2.2.1.2.1. ISD-AS number Attribute

The ISD-AS number attribute identifies the SCION ISD and AS. In the SCION open source implementation, the attribute type is id-at-ia, defined as:
id-at-ia AttributeType ::= {id-scion id-cppki(1) id-at(2) 1}

where id-scion specifies the root SCION object identifier (OID).

Note: The root SCION object identifier (OID) for the SCION open-source implementation is the IANA Private Enterprise Number '55324':
id-scion ::= OBJECT IDENTIFIER {1 3 6 1 4 1 55324}

The string representation of the ISD-AS number attribute MUST follow the text representation defined in [I-D.dekater-scion-controlplane], section "Text Representation" where AS numbers in the lower 32-bit range are represented in decimal notation, and others in hexadecimal notation.

The ISD-AS number attribute MUST be present exactly once in the distinguished name of the certificate issuer or owner, specified in the issuer or subject field respectively. Implementations MUST NOT create nor successfully verify certificates whose issuer and subject fields do not include the ISD-AS number at all, or include it more than once.

CA certificates MUST include an ISD-AS number in their distinguished name so the control plane knows from which AS to retrieve the certificate, thereby avoiding circular dependencies.

Note: Voting certificates are not required to include the ISD-AS number attribute in their distinguished name.

2.2.2. Extensions

[RFC5280], section 4.2.1, defines the syntax of the Extensions sequence in a X.509 certificate. Descriptions of each standard certificate extension can be found in [RFC5280], section 4.2.1. The corresponding clauses in [X.509] are clause 7.2 and clause 9, respectively.

Currently, the following extensions are relevant for SCION:

* authorityKeyIdentifier

* subjectKeyIdentifier

* keyUsage

* extKeyUsage

* basicConstraints

The following sections describe the SCION-specifics in regard to these extensions.

2.2.2.1. authorityKeyIdentifier Extension

The authorityKeyIdentifier extension identifies the public key corresponding to the private key used to sign a certificate.

For the syntax and definition of the authorityKeyIdentifier extension, see [RFC5280], section 4.2.1.1, and [X.509], clause 9.2.2.1.

The authorityKeyIdentifier extension provides three attributes to specify the public key:

* keyIdentifier

* authorityCertIssuer

* authorityCertSerialNumber

In SCION, using the keyIdentifier attribute is the preferred way to specify the authorityKeyIdentifier extension.

Important: SCION implementations MAY also support the use of the authorityCertIssuer and authorityCertSerialNumber attributes. However, if these attributes are set and support for them is missing, implementations SHOULD error out.

This extension MUST always be non-critical. However, SCION implementations MUST error out if the extension is not present AND the certificate is not self-signed.

2.2.2.2. subjectKeyIdentifier Extension

The subjectKeyIdentifier extension identifies certificates that contain a particular public key. It can be used, for example, by control plane messages to identify which certificate to use for verification. The extension allows for overlapping control plane CA keys, for example during updates.

For the syntax and definition of the subjectKeyIdentifier extension, see [RFC5280], section 4.2.1.2, and [X.509], clause 9.2.2.2.

This extension MUST always be non-critical. However, SCION implementations MUST error out if the extension is not present.

2.2.2.3. keyUsage Extension

The keyUsage extension identifies the intended usage of the public key in the corresponding certificate. For the syntax and definition of the keyUsage extension, see [RFC5280], section 4.2.1.3, and [X.509], clause 9.2.2.3.

The attributes of the keyUsage extension define possible ways of using the public key. The attributes have the following meaning in SCION:

* digitalSignature: The public key can be used to verify the digital signature of a control plane payload.

* contentCommitment: Not used.

* keyEncipherment: Not used.

* dataEncipherment: Not used.

* keyAgreement: Not used.

* keyCertSign: The public key can be used to verify the CA signature on a control plane certificate.

* cRLSign: Not used.

* encipherOnly: Not used.

* decipherOnly: Not used.

Important: If a certificate’s public key is used to verify the signature of a control plane payload (digitalSignature attribute), it MUST be possible to trace back the private key used to sign the certificate. This is done by referencing the ISD-AS and the subject key identifier (via the subjectKeyIdentifier extension). For more information about the subjectKeyIdentifier extension, see Section 2.2.2.2.

If present, the keyUsage extension SHOULD be marked as "critical". That is, the critical Boolean attribute of this extension MUST be set to TRUE (the default is FALSE).

Note: If a certificate extension is marked "critical", the public key in the certificate SHOULD only be used for the purpose set in the critical extension.

Each Control Plane PKI certificate type uses the public key differently, and consequently also specifies the attributes of the keyUsage extension differently. The next table shows the specifications per certificate type.

Certificate Type	Root	CA	AS	Voting (regular and sensitive)
Attribute:				
keyUsage extension itself	MUST be present	MUST be present	MUST be present	MAY be present (but is not required)
digitalSignature	MUST NOT be set (1)	MUST NOT be set (2)	MUST be set	If the extension is present, the digitalSignature attribute MUST NOT be set
keyCertSign	MUST be set	MUST be set	MUST NOT be set	If the extension is present, the keyCertSign attribute MUST NOT be set

Table 3: keyUsage extension - Specifications per certificate type

(1) The root certificate SHOULD NOT be used to verify control plane messages.
(2) The CA certificate SHOULD NOT be used to verify control plane messages.

2.2.2.4. extKeyUsage Extension

The extKeyUsage extension specifies additional usages of the public key in the certificate. For the syntax and definition of the extKeyUsage extension, see [X.509], clause 9.2.2.4.

SCION uses the following attributes of the Extended Key Usage extension, as defined in Section 4.2.1.12 of [RFC5280]:

* id-kp-serverAuth: If set, the public key can be used for SCION Control Plane server authentication.

* id-kp-clientAuth: If set, the public key can be used for SCION Control Plane client authentication.

* id-kp-timeStamping: If set, the public key can be used for the verification of timestamps.

Additionally, the Extended Key Usage extension sequence MAY include the SCION-specific attributes id-kp-root, id-kp-regular, and id-kp-sensitive. These attributes are used in the TRC setup, to distinguish root certificates, regular voting certificates, and sensitive voting certificates from each other. For more information, see Section 3.1.2.2.11.

The specifications of the extKeyUsage extension differ per SCION Control Plane PKI certificate type. The next table provides an overview of the specifications per certificate type.

Certificate Type	Root	CA	AS	Voting (regular and sensitive)
Attribute:				
extKeyUsage extension itself	MUST be present	MAY be present (not required)	MUST be present	MUST be present
id-kp-serverAuth	MUST NOT be included	MUST NOT be included	MUST be included, if the certificate is used on the server-side of a control plane TLS session.	MUST NOT be included
id-kp-clientAuth	MUST NOT be included	MUST NOT be included	MUST be included, if the certificate is used on the client-	MUST NOT be included

			side of a control plane TLS session.	
id-kp-timeStamping	MUST be included		MUST be included	MUST be included
SCION-specific	id-kp-root MUST be included. For details, see Section 2.2.2.4.1			Regular voting cert: id-kp-regular MUST be included. For details, see Section 2.2.2.4.1 Sensitive voting cert: id-kp-sensitive MUST be included. For details, see Section 2.2.2.4.1

Table 4: extKeyUsage extension - Specifications per certificate type

2.2.2.4.1. SCION-Specific Attributes

The id-kp-root, id-kp-regular, and id-kp-sensitive attributes MUST be specified as follows:

* Root certificate:
  id-kp-root AttributeType ::= {id-scion id-cppki(1) id-kp(3) 3}

* Regular voting certificate:
  id-kp-regular AttributeType ::= {id-scion id-cppki(1) id-kp(3) 2}

* Sensitive voting certificate:
  id-kp-sensitive AttributeType ::= {id-scion id-cppki(1) id-kp(3) 1}

where id-scion specifies the root SCION object identifier (OID).

Note: The root SCION object identifier (OID) for the SCION open-source implementation is the IANA Private Enterprise Number ’55324’: id-scion ::= OBJECT IDENTIFIER {1 3 6 1 4 1 55324}

2.2.2.5. basicConstraints Extension

The basicConstraints extension specifies whether the certificate subject may act as a CA. For the syntax and definition of the basicConstraints extension, see [X.509], clause 9.4.2.1.

The basicConstraints extension includes the following attributes relevant for SCION:

* cA attribute: Specifies whether the certificate subject may act as a CA. If yes, this attribute MUST be set to TRUE.

* pathLenConstraint attribute: This attribute is only relevant if the cA attribute is set to TRUE. It specifies the maximum number of CA certificates that may follow this CA certificate in the certification chain. Value "0" means that this CA may only issue end-entity certificates, but no CA certificates. If the attribute is not set, there is no limit to the maximum length of the certification path.

The settings of the basicConstraints extension differ for each SCION Control Plane PKI certificate type. The next table shows the specifications per certificate type.

Certificate Type	Root	CA	AS	Voting (regular and sensitive)
Attribute:				
basicConstraints extension itself	MUST be present	MUST be present	SHOULD NOT be present	SHOULD NOT be present
cA	MUST be set to TRUE	MUST be set to TRUE	If the extension is present, this attribute MUST be set to FALSE	If the extension is present, this attribute MUST be set to FALSE
pathLenConstraint	SHOULD be set to "1", MUST be marked as "critical"	SHOULD be set to "0" (1), MUST be marked as "critical"	If the extension is present, this attribute MUST be absent.	If the extension is present, this attribute MUST be absent.

Table 5: basicConstraints extension - Specifications per certificate type

(1) Control Plane CAs can only issue end-entity certificates.

3. Trust Root Configuration Specification

This section provides an in-depth specification of the Trust Root Configuration (TRC) file (see Section 3.1). The TRC contains policy information about an ISD and acts as a distribution mechanism for the trust anchors of that ISD. It enables the securing of control plane interactions and is thus an integral part of the SCION infrastructure.

The initial TRC of an ISD is signed during a signing ceremony and then distributed throughout the ISD. This signing ceremony follows specific rules; Section 3.2 describes these rules.

3.1. TRC Specification

The TRC is a signed collection of [X.509] v3 certificates. Additionally, the TRC contains ISD-specific policies encoded in a Cryptographic Message Syntax (CMS) [RFC5652] envelope.

The TRC’s certificates collection consists of a set of control plane root certificates which build the root of the certification chain for the AS certificates in an ISD. The other certificates in the TRC are solely used for signing the next TRC, a process called "voting". The verification of a new TRC thus depends on the policies and voting certificates defined in the previous TRC.

This section specifies the TRC including format definitions and dpayload fields. The section uses the ITU-T [X.680] syntax.

3.1.1. TRC Types and States

The following types of TRCs exist:

* Initial: The very first TRC of an ISD is the initial TRC of that ISD. It is a special case of the base TRC, where the number of the ISD is specified.

* Base: A base TRC is either the initial TRC, or the first TRC after a trust reset (see Section 1.4.1). Trust for a base TRC cannot be inferred by verifying a TRC update; base TRCs are trusted axiomatically, similarly to how root CA certificates are trusted by clients in the Web PKI.

* Update: All non-base TRCs are updated TRCs. They are the product of either a regular or a sensitive update.

A TRC can have the following states:

* Valid: The validity period of a TRC is defined in the TRC itself, in the validity field (see Section 3.1.2.2.3). A TRC is considered valid if the current time falls within its validity period.

* Active: An active TRC is a valid TRC that can be used for verifying certificate signatures. This is either the latest TRC or the predecessor TRC, if it is still in its grace period (as defined in the gracePeriod field of the new TRC, see Section 3.1.2.2.4). No more than two TRCs can be active at the same time for any ISD.

Figure 2 shows the content of both a base/initial TRC, the changes made with the first regular update to the base TRC. All elements of the TRC is detailed in the following subsections.

3.1.2. TRC Format

The TRC defines the roots of trust of an ISD and is the basis of the ISD's Control Plane PKI. It holds the root and voting certificates of the ISD and defines the ISD's trust policy.

3.1.2.1. TRC Schema

The following code block shows the format of a TRC specification file (the payload schema):

```
TRCPayload  ::=  SEQUENCE {
    version   TRCFormatVersion,
    iD        TRCID,
    validity  Validity,

    gracePeriod   INTEGER,
    noTrustReset  BOOLEAN DEFAULT FALSE,
    votes         SEQUENCE OF INTEGER (SIZE (1..255)),

    votingQuorum  INTEGER (1..255),

    coreASes           SEQUENCE OF ASN,
    authoritativeASes  SEQUENCE OF ASN,
    description        UTF8String (SIZE (0..1024)),

    certificates       SEQUENCE OF Certificate }

TRCFormatVersion  ::=  INTEGER { v1(0) }

TRCID  ::=  SEQUENCE {
    iSD           ISD,
    serialNumber  INTEGER (1..MAX),
    baseNumber    INTEGER (1..MAX) }

ISD  ::=  INTEGER (1..65535)

Validity  ::=  SEQUENCE {
    notBefore  Time,
    notAfter   Time }

ASN  ::=  INTEGER (1..281474976710655)
```

The TRCPayload sequence contains the identifying information of a TRC as well as policy information for TRC updates. Furthermore, it defines the list of certificates that build the trust anchor of the ISD.

For signature calculation, the data that is to be signed is encoded using ASN.1 distinguished encoding rules (DER) [X.690]. For more details, see Section 3.1.3.

3.1.2.2. TRC Fields

This section describes the syntax and semantics of all TRC payload fields.

3.1.2.2.1. version Field

The version field describes the version of the TRC format specification.

Currently, the version MUST always be "v1".

3.1.2.2.2. iD Field

The iD field specifies the unique identifier of the TRC.

The identifier is a unique sequence of

* ISD number (iSD attribute),
* base number (baseNumber attribute), and
* TRC serial number (serialNumber attribute).

All numbers MUST be positive integers.

* The *ISD number* MUST be an integer in the inclusive range from 64 to 4094 (i.e., the numbering range for public ISDs, see Section 1.6.1).
* The *base number* indicates the starting point of the current TRC update chain. This starting point is either the ISD’s initial TRC or the currently valid base TRC, if the valid base TRC differs from the initial TRC. The latter MUST be the case after a trust reset.
* The *serial number* represents the current update cycle, counting from the initial TRC of a specific ISD.

A TRC where the base number is equal to the serial number is a base TRC. The initial TRC is a special case of a base TRC and MUST have a serial number of 1 and a base number of 1. With every TRC update, the serial number MUST be incremented by one. This facilitates uniquely identifying the predecessor and successor TRC in an update chain.

If a trust reset is necessary, a new base TRC is announced in order to start a new and clean TRC update chain. The base number of this new TRC update chain SHOULD be the number following the serial number of the latest TRC that was produced by a non-compromised TRC update for this ISD.

Example
The following simple example illustrates how to specify the ID of the TRCs in an TRC update chain for _ISD 15_. The IDs are given in a human-readable notation, where Bxx is the base number, and Sxx the serial number.

Update	TRC ID	Remarks
Initial	ISD15-B01-S01	
Regular	ISD15-B01-S02	Only the serial number is incremented.
Regular	ISD15-B01-S03	Only the serial number is incremented.
Sensitive	ISD15-B01-S04	Only the serial number is incremented.
Trust reset	ISD15-*B05*-S05	A trust reset includes the creation of a new base TRC. The new base number follows the serial number "04" of the latest TRC resulting from a non-compromised TRC update for this ISD.
Regular	ISD15-B05-S06	Only the serial number is incremented.
Regular	ISD15-B05-S07	Only the serial number is incremented.
And so on		

Table 6: ID of TRCs in TRC update chain

3.1.2.2.3. validity Field

The validity field defines the validity period of the TRC. This is the period of time during which the TRC is in the "valid" state. The notBefore and notAfter attributes of the validity field specify the lower and upper bound of the time interval during which a TRC can be active.

Note: An active TRC is a valid TRC that can be used for verifying certificate signatures. The time period during which a TRC is active can be shorter than the time period during which the TRC is valid. For more information, see Section 3.1.1.

The validity field consists of a sequence of two dates, as defined in section 7.2. of [X.509].

In addition to this standard definition, the following constraint applies to the validity field of the TRC:

* All TRCs MUST have a well-defined expiration date. SCION implementations MUST NOT create TRCs that use the "99991231235959Z" generalized time value, and verifiers MUST error out when encountering such a TRC.

3.1.2.2.4. gracePeriod Field

The gracePeriod field of a TRC specifies the period of time during which the predecessor TRC can still be considered active (the "grace period"). The grace period starts at the beginning of the validity period of the new TRC.

The validity period of the predecessor TRC ends when:

* the grace period has passed;

* the predecessor’ s expiration time is reached; or

* the successor TRC of the new TRC has been announced.

Note: The event that happens first marks the end of the predecessor’s validity period.

The gracePeriod field defines the grace period as a number of seconds (positive integer).

The value of the gracePeriod field in a base TRC MUST be zero. The value of the gracePeriod field in a non-base TRC SHOULD be non-zero. It SHOULD be long enough to provide sufficient overlap between the TRCs in order to facilitate interruption-free operations in the ISD. If the grace period is too short, some Control Plane AS certificates might expire before the corresponding AS can fetch an updated version from its CA.

3.1.2.2.5. noTrustReset Boolean

The noTrustReset Boolean specifies whether a trust reset is forbidden by the ISD. Within a TRC update chain, this value MUST NOT be changed by a regular or sensitive update. However, it is possible to change the noTrustReset value in the event of a trust reset, where a new base TRC is created.

The noTrustReset field is OPTIONAL and defaults to FALSE.

Important: Note that once the noTrustReset Boolean is set to TRUE and a trust reset is disallowed, this cannot be reversed. Therefore, ISDs SHOULD always set this value to FALSE, unless they have sufficiently assessed the risks and implications of making a trust reset impossible.

Note: A trust reset represents a special use case where a new base TRC is created. It therefore differs from a TRC update (regular or sensitive), as the signatures in the new base TRC cannot be verified with the certificates contained in the predecessor TRC. Instead, a trust reset base TRC must be axiomatically trusted, similarly to how the initial TRC is trusted.

3.1.2.2.6. votes Field

The votes field contains a sequence of indices that refer to the voting certificates in the predecessor TRC. If index i is part of the votes field, then the voting certificate at position i in the certificates sequence of the predecessor TRC casted a vote on the successor TRC. For more information on the certificates sequence, see Section 3.1.2.2.11.

Note: In a base TRC, the votes sequence is empty.

Every entry in the votes sequence MUST be unique.
Further restrictions on votes are discussed in Section 3.1.5.

Note: The votes sequence of indices is mandatory in order to prevent stripping voting signatures from the TRC. Absence of the votes sequence makes it possible to transform a TRC with more voting signatures than the voting quorum into multiple verifiable TRCs with the same payload, but different voting signature sets. This would violate the requirement of uniqueness of a TRC.

3.1.2.2.7. votingQuorum Field

The votingQuorum field defines the number of necessary votes on a successor TRC to make it verifiable.

A voting quorum greater than one will prevent a single entity from creating a malicious TRC update.

3.1.2.2.8. coreASes Field

The coreASes field contains the AS numbers of the core ASes in this ISD.

Each core AS number MUST be unique in the sequence of core AS numbers. That is, each AS number MUST appear only once in the coreASes field.

3.1.2.2.8.1. Revoking or Assigning Core Status

* To revoke the core status of a given AS, remove the respective AS number from the sequence of AS numbers in the coreASes field.

* To assign the core status to a given AS, add the respective AS number to the sequence of AS numbers in the coreASes field.

Important: Revoking or assigning the core status of/to an AS always requires a (sensitive) TRC update.

3.1.2.2.9. authoritativeASes Field

The authoritativeASes field contains the AS numbers of the authoritative ASes in this ISD.

Authoritative ASes are those ASes in an ISD that always have the latest TRCs of the ISD. As a consequence, authoritative ASes also start the announcement of a TRC update.

* Every authoritative AS MUST be a core AS and be listed in the coreASes field.

* Each authoritative AS number MUST be unique in the sequence of authoritative AS numbers. That is, each AS number MUST NOT appear more than once in the authoritativeASes field.

3.1.2.2.9.1. Revoking or Assigning Authoritative Status

* To revoke the authoritative status of a given AS, remove the respective AS number from the sequence of AS numbers in the authoritativeASes field.

* To assign the authoritative status to a given AS, add the respective AS number to the sequence of AS numbers in the authoritativeASes field.

Important: Revoking or assigning the authoritative status of/to an AS always requires a (sensitive) TRC update.

3.1.2.2.10. description Field

The description field contains a UTF-8 encoded string that describes the ISD.

* The description field SHOULD NOT be empty.

* The description of the ISD MUST be in English. Additionally, the description field MAY contain information in other languages.

3.1.2.2.11. certificates Field

The voting ASes and the certification authorities (CAs) of an ISD are not specified explicitly in the ISD’s TRC. Instead, this information is defined by the list of voting and root certificates in the certificates field of the TRC payload.

The certificates field is a sequence of self-signed X.509 certificates. Each certificate in the certificate sequence MUST be one of the following types:

* a sensitive voting certificate,

* a regular voting certificate, or

* a CP root certificate.

A certificate that is no control plane root or voting certificate MUST NOT be included in the sequence of certificates in the certificates field.

Note: A certificate’s type (voting or root) is specified in the extKeyUsage extension of the certificate, by means of the SCION-specific attributes id-kp-regular, id-kp-sensitive, and id-kp-root, respectively. For more information, see Section 2.2.2.4.

The following constraints MUST hold for each certificate in the certificates field of the TRC payload:

* Each certificate MUST be unique in the sequence of certificates. That is, each certificate MUST NOT appear more than once in the certificates field.

* The issuer / serialNumber pair for each certificate MUST be unique.

* If an ISD-AS number is present in the distinguished name of the certificate, this ISD number MUST be equal to the ISD number of the TRC (which is defined in the iD field (see Section 3.1.2.2.2).

* Every certificate MUST have a validity period that fully contains the validity period of this TRC. That is, the notBefore date of this TRC's validity period MUST be equal to or later than the certificate's notBefore date, and the notAfter date of this TRC's validity period MUST be before or equal to the certificate's notAfter date.

* Per certificate type, every certificate distinguished name MUST be unique.

The following must hold for the entire sequence of certificates in the certificates field:

* votingQuorum <= count (sensitive voting certificates)
  That is, the quorum defined in the TRC's votingQuorum field (Section 3.1.2.2.7) MUST be smaller than or equal to the number of _sensitive_ voting certificates specified in the TRC's certificates field.

* votingQuorum <= count (regular voting certificates)
  That is, the quorum defined in the TRC's votingQuorum field (Section 3.1.2.2.7) MUST be smaller than or equal to the number of _regular_ voting certificates specified in the TRC's certificates field.

3.1.3. TRC Signature Syntax

A TRC contains policy information about an ISD and acts as a distribution mechanism for the trust anchors of that ISD.

Each TRC is digitally signed and the syntax used to sign and encapsulate the TRC payload is the Cryptographic Message Syntax (CMS) as defined in [RFC5652]. The signed TRC payload is of the CMS signed-data content type, as defined in Section 5 of [RFC5652], and encapsulated in a CMS ContentInfo element, as defined in Section 3 of [RFC5652].

For detailed information on the general syntax definitions of the Cryptographic Message Syntax, see sections 3 and 5 of [RFC5652].

3.1.3.1. SCION-specific rules

SCION implementations MUST fulfil the following additional rules, as well as the general syntax rules specified in [RFC5652]:

* EncapsulatedContentInfo sequence:

  - The eContentType field MUST be set to "id-data".

  - The content of the eContent field MUST be the DER-encoded TRC payload. This has the benefit that the format is backwards compatible with PKCS #7, as described in Section 5.2.1 of [RFC5652].

* SignedData sequence:

  - The certificates field MUST be left empty. The certificate pool used to verify a TRC update is already specified in the certificates field of the predecessor TRC's payload (see also Section 3.1.2.2.11).

  - The version field MUST be set to "1". This is because SCION uses the "id-data" content type to encapsulate content info, and does not specify any certificate in the SignedData sequence (see also Section 5.1 of [RFC5652]).

* SignerIdentifier choice:

  - The type of signer identifier chosen here MUST be IssuerAndSerialNumber.

* SignerInfo sequence:

  - The version field MUST be set to "1". This is because SCION uses the IssuerAndSerialNumber type of signer identifier (see also Section 5.3 of [RFC5652]).

  - The algorithm specified in the signatureAlgorithm field MUST be one of the algorithms supported by SCION (for details, see signature Field - Additional Information (Section 2.2.1.1)).

  - The digestAlgorithm is determined by the algorithm specified in the signatureAlgorithm field.

3.1.3.2. TRC Equality

The signer information in the signed TRC is part of an unordered set, as per [RFC5652]. This implies that the signer information can be reordered without affecting verification, although certain operations require TRCs to be equal im accordance with the following definition:

Two TRCs are equal, if and only if their payloads are byte equal.

Two TRCs with byte equal payloads can be considered as equal because the TRC payload exactly defines which signatures must be attached in the signed TRC:

* The REQUIRED signatures from voting certificates are explicitly mentioned in the votes field of the payload: If index "i" is part of the votes field, then the voting certificate at position i in the certificates sequence of the predecessor TRC casted a vote on the successor TRC. See also Section 3.1.2.2.6.

* The REQUIRED signatures for new certificates are implied by the currently valid TRC payload, and, in case of a TRC update, the predecessor payload.

3.1.4. Certification Path - Trust Anchor Pool

The certification path of a Control PlaneAS certificate starts in a Control Plane root certificate. The Control Plane root certificate for a given ISD is distributed via the TRC.

However, AS certificates and the corresponding signing CA certificates are *not* part of the TRC, but bundled into certificate chains and distributed separately from the corresponding TRC. This separation makes it possible to extend the validity period of the root certificate, and to update the corresponding TRC without having to modify the certificate chain. To be able to validate a certification path, each AS builds a collection of root certificates from the latest TRC of the relevant ISD.

The following section explains how to build a trust anchor pool.

Note: Any entity sending information that is secured by the Control Plane PKI, such as control plane messages, MUST be able to provide all the necessary trust material, such as certificates, to verify said information. If any cryptographic material is missing in the process, the relying party MUST query the originator of the message for the missing material. If it cannot be resolved, the verification process fails. For more details, see 4.2 "Signing and Verifying Control Plane Messages" Section 4.2.

3.1.4.1. TRC Selection For Trust Anchor Pool

The selection of the right set of TRCs to build the trust anchor pool depends on the time of verification. The trust anchor pool is usually used to verify control plane messages and this case, the time of verification is the current time. However, if the trust anchor pool will be used for auditing, the time of verification is the point in time to check whether a given signature was verifiable.

The selection algorithm for building the trust anchor pool is described in pseudo-python code below.

```
def select_trust_anchors(trcs: Dict[(int,int), TRC], \
verification_time: int) -> Set[RootCert]:
    """
    Args:
        trcs: The dictionary mapping (serial number, \
        base number) to the TRC for a given ISD.
        verification_time: The time of verification.

    Returns:
        The set of CP Root certificates acting as trust anchors.
    """
    # Find highest base number that has a TRC with validity
    # period starting before verification time.
    base_nr = 1
    for trc in trcs.values()
        if trc.id.base_nr > base_nr and trc.validity.not_before \
        <= verification_time:
            base_nr = trc.id.base_nr

    # Find TRC with highest serial number with given base number
    # and a validity period starting before verification time.
    serial_nr = 1
    for trc in trcs[isd].values():
        if trc.id.base_nr != base_nr:
            continue
        if trc.id.serial_nr > serial_nr and \
        trc.validity.not_before <= verification_time:
            serial_nr = trc.id.serial_nr

    candidate = trcs[(serial_nr, base_nr)]

    # If the verification time is not inside the validity period,
    # there is no valid set of trust anchors.
    if not candidate.validity.contains(verification_time):
        return set()

    # If the grace period has passed, only the certificates in
    # that TRC may be used as trust anchors.
    if candidate.validity.not_before + candidate.grace_period \
    < verification_time:
        return collect_trust_anchors(candidate)

    predecessor = trcs.get((serial_nr-1, base_nr))
    if not predecessor or predecessor.validity.not_after < \
    verification_time:
```

```
            return collect_trust_anchors(candidate)

        return collect_trust_anchors(candidate) | \
        collect_trust_anchors(predecessor)


    def collect_trust_anchors(trc: TRC) -> Set[RootCert]:
        """
        Args:
            trc: A TRC from which the CP Root Certificates shall \
            be extracted.

        Returns:
            The set of CP Root certificates acting as trust anchors.
        """
        roots = set()
        for cert in trc.certificates:
            if not cert.basic_constraints.ca:
                continue
            roots.add(cert)
        return roots
```

3.1.5. TRC Updates

All non-base TRCs of an ISD are updates of the ISD’s base TRC(s). The TRC update chain consists of regular and sensitive TRC updates, and the type of update determines the (payload) information that changes in the updated TRC.

This section describes the rules that apply to updating a TRC in regard to the payload information contained in the TRC. Some rules are valid for both update types whilst some only apply to a regular or a sensitive TRC update. Based on the type of update, different sets of voters are needed to create a verifiable TRC update and the corresponding voting (signing) process is also described. To verify a TRC update, a relying party MUST perform a couple of checks which are also listed.

3.1.5.1. Changed or New Certificates

In the context of a TRC update,

* A certificate is _changing_, if the certificate is part of the certificates sequence in the predecessor TRC, but no longer part of the certificates sequence in the updated TRC. Instead, the certificates sequence of the updated TRC holds another certificate of the _same type_ and with the _same distinguished name_.

* A certificate is _new_, if there is *no* certificate of the same type and distinguished name at all in the certificates sequence of the predecessor TRC.

Note: Every new sensitive or regular voting certificate in a TRC attaches a signature to the TRC. This is done to ensure that the freshly included voting entity agrees with the contents of the TRC it is now part of.

3.1.5.2. Update Rules - Overview

The following table gives an overview of the types of TRC update as well as the rules that must apply in regard to the updated TRC's payload information.

The sections that follow provide more detailed descriptions of each rule.

Type of Update	Payload Updated TRC - Unchanged Elements	Payload Updated TRC - Required Changes	Payload Updated TRC: Other Rules to Hold
Both Regular AND Sensitive Updates	- iD field: iSD and baseNumber - noTrustReset field	iD field: serialNumber MUST be incremented by 1	votes field: Number of votes (indices) => number set in the votingQuorum field of the predecessor TRC
Regular TRC Update	- Quorum in the votingQuorum field - Core ASes in the coreASes field - ASes in the authoritativeASes field - Nr. and distinguished names of root & voting certificates in the certificates field - Set of sensitive voting certificates in the certificates field		votes field: - All votes MUST only refer to _regular_ voting certificates in the predecessor TRC - Must include votes of each changed regular voting certificate from the predecessor TRC signatures field: - Must include signatures of each changed root certificate from the predecessor TRC
Sensitive TRC Update	If the update does not qualify as a regular update, it is a sensitive update		votes field: - All votes MUST only refer to _sensitive_ voting certificates in the predecessor TRC

Table 7: Overview of the update types and corresponding rules

3.1.5.3. General Update Rules

The following rules MUST hold for each updated TRC, independent of the update type:

* The iSD and baseNumber in the iD field MUST NOT change (see also Section 3.1.2.2.2).

* The serialNumber in the iD field MUST be incremented by one.

* The noTrustReset field MUST NOT change (see also Section 3.1.2.2.5).

* The votes sequence of the updated TRC MUST only contain indices that refer to sensitive or regular voting certificates in the predecessor TRC. This guarantees that the updated TRC only contains valid votes authenticated by sensitive or regular voting certificates in the predecessor TRC. For more information, see Section 3.1.2.2.6 and Section 3.1.2.2.11.

* The number of votes in the updated TRC MUST be greater than or equal to the number set in the votingQuorum field of the predecessor TRC (see Section 3.1.2.2.7). The number of votes corresponds to the number of indices in the votes field of the updated TRC.

3.1.5.4. Regular TRC Update

A regular TRC update is a periodic re-issuance of the TRC where the entities and policies listed in the TRC remain unchanged.

A TRC update qualifies as a regular update, if the following rules apply in regard to the TRC’s payload information.

* The settings of the following fields in the updated TRC MUST remain the same compared to the predecessor TRC:

  - The voting quorum set in the votingQuorum field.

  - The core ASes specified in the coreASes field.

  - The authoritative ASes specified in the authoritativeASes field.

  - The number of sensitive and regular voting certificates as well as Control Plane root certificates included in the certificates field, and their distinguished names.

  - The set of sensitive voting certificates specified in the certificates field.

* For every regular voting certificate that changes, the regular voting certificate in the predecessor TRC is part of the voters on the updated TRC. That is, for each changed regular voting certificate, an index in the votes field of the updated TRC MUST refer to the changed regular voting certificate in the predecessor TRC.

* For every Control Plane root certificate that changes, the updated TRC MUST include a signature created with the private key belonging to the changed Control Plane root certificate (which is part of the predecessor TRC).

* In order for a regular TRC update to be verifiable, all votes MUST be cast by _regular_ voting certificates. That is, each index in the votes field of the regularly updated TRC MUST refer to a _regular_ voting certificate in the certificates field of the predecessor TRC.

3.1.5.5. Sensitive TRC Update

If a TRC update does not qualify as a regular update, it is considered a sensitive update.

* In order for a sensitive update to be verifiable, all votes MUST be cast by _sensitive_ voting certificates. That is, each index in the votes field of the sensitively updated TRC MUST refer to a _sensitive_ voting certificate in the certificates field of the predecessor TRC.

3.1.5.6. Signing a TRC Update

As described above, a set of voters MUST cast votes on the updated TRC to make it verifiable. The votingQuorum field of the predecessor TRC (see Section 3.1.2.2.7) defines the required number of voters, which will represent regular or sensitive voting certificates, respectively.

Furthermore, if one or more _new_ certificates are added to the updated TRC, the corresponding voting representatives MUST also sign the updated TRC in order to show that they have access to the private keys listed in these fresh certificates. This is called "showing proof-of-possession", and done by signing the TRC with the respective private key. For the distinction between changed and new certificates in a TRC update, see Section 3.1.5.1.

It is up to the ISD members to decide how the "casting a vote" procedure for updated TRCs will take place. Some ISDs will make a distinction between regular and sensitive updates. These ISDs divide the regular and sensitive signing keys in different security classes and act accordingly, e.g. they keep the regular key in an online vault while the sensitive key would be stored offline. This way, the regular TRC update would lend itself to being automated (since the keys are accessible online) whereas the sensitive one would require manual actions to access the offline key. However, other ISDs keep both regular and sensitive keys online and perform both updates automatically.

3.1.5.7. TRC Update Verification

To verify a TRC update, the relying party MUST perform the following checks:

* Check that the specified update rules as described above are respected.

* Check whether the update is regular or sensitive.

  - In case of a regular update,

    o check that the signatures for the changing certificates are present and verifiable, and

    o check that all votes are cast by a regular voting certificate.

  - In case of a sensitive update, check that all votes are cast by a sensitive voting certificate.

* In both cases, check that all signatures are verifiable, and no superfluous signatures are attached.

If one or more of the above checks gives a negative result, the updated TRC SHOULD be rejected.

3.2. Initial TRC Signing Ceremony

The very first base TRC of an ISD - called the initial TRC - is a special case of the base TRC where the number of the ISD is chosen. The initial TRC MUST be signed during a special signing ceremony - all voting representatives of the initial TRC need to take part in this signing ceremony to sign this and exchange their public keys. Following this, all entities within an ISD can obtain the TRC by means of a secure offline or online mechanism.

Appendix "Appendix A. Signing Ceremony Initial TRC" describes a possible procedure for the signing ceremony of an ISD's initial TRC. It is in principle up to the initial members of an ISD how to organize the signing ceremony. However, it is recommended having a process in line with the ceremony described in the Appendix.

4. Deploying the CP PKI - Specifications

This section provides several specifications regarding the deployment of the control plane PKI.

4.1. Deploying a TRC

4.1.1. Base TRC

Base TRCs are trust anchors and thus axiomatically trusted. All ASes within an ISD MUST be pre-loaded with the currently valid base-version TRC of their own ISD. For all specifications regarding the creation and distribution of initial/base TRCs, see Section 3.2.

4.1.2. TRC Update

All non-base TRCs of an ISD are updates of the ISD's base TRC(s). The TRC update chain consists of regular and sensitive TRC updates. The specifications and rules that apply to updating a TRC are described in Section 3.1.5.

4.1.2.1. TRC Update Discovery

Relying parties MUST have at least one valid TRC available. Relying parties MUST discover TRC updates within the grace period defined in the updated TRC. They SHOULD discover TRC updates in a matter of minutes to hours. Additionally, the following requirement MUST be satisfied:

Requirement
Any entity sending information that is secured by the Control Plane PKI MUST be able to provide all the necessary trust material to verify said information.

SCION provides the following mechanisms for discovering TRC updates and fulfilling the above requirement:

* _Beaconing Process_
  The TRC version is announced in the beaconing process. Each AS MUST announce what it considers to be the latest TRC, and MUST include the hash value of the TRC contents to facilitate the discovery of discrepancies. Therefore, relying parties that are

part of the beaconing process discover TRC updates passively, i.e. a core AS notices TRC updates for remote ISDs that are on the beaconing path. A non-core AS only notices TRC updates for the local ISD through the beaconing process. The creation of a new TRC SHOULD trigger the generation of new control plane messages, as the propagation of control plane messages will help other ASes rapidly discover the new TRC.

* _Path Lookup_
  In every path segment, all ASes MUST reference the latest TRC of their ISD. Therefore, when resolving paths, every relying party will notice TRC updates, even remote ones.

* _Active Discovery_
  Any TRC can be obtained at any time from the sender of the information it secures; either in a specific version or in its latest available version. The necessary query and response is described in [I-D.dekater-scion-controlplane], section "Control Service gRPC API - Trust Material".

Note: The first two mechanisms above only work when there is active communication between the relying party and the ISD in question.

4.2. Signing and Verifying Control Plane Messages

SCION requires that control plane messages are signed. The main purpose of the Control Plane PKI is providing a mechanism to distribute and authenticate public keys that are used to verify control plane messages and information, e.g. each hop information in a path segment is signed by the respective AS. Consequently, all relying parties MUST be able to verify signatures with the help of the Control Plane PKI.

The following sections specify the requirements that apply to the signing and verification of control plane messages.

4.2.1. Signing a Control Plane Message

An AS signs control plane messages with the private key that corresponds to the (valid) AS’ certificate.

The AS MUST attach the following information as signature metadata. It is the minimum information a relying party requires to identify which certificate to use to verify the signed message.

* ISD-AS number: The ISD-AS number of the signing entity. For specification details, see Section 2.2.1.2.1.

* Subject key identifier: The identifier of the public key that MUST be used to verify the message. For specification details, see Section 2.2.2.2.

Additionally, the signer SHOULD include the following information:

* Serial and base number of the latest TRC: Including this information allows relying parties to discover TRC updates and trust resets. For specification details, see Section 3.1.2.2.2.

* Timestamp: For many messages, the time at which it was signed is useful information to ensure freshness.

4.2.2. Verifying a Control Plane Message

When the relying party receives a control plane message they want to verify, the relying party first needs to identify the certificate needed to validate the corresponding signature on the message.

AS certificates are bundled together with the corresponding signing CA certificate into certificate chains. For efficiency, SCION distributes these certificate chains separately from the signed messages.

A certificate chain is verified against the Control Plane root certificate, although the the root certificate is bunded with the TRC and *not* in the chain. This makes it possible to extend the validity period of the root certificate and update the corresponding TRC without having to modify the certificate chain.

To verify a control plane message, the relying party MUST perform the following steps:

1. Build a collection of root certificates from the latest TRC of the relevant ISD (that is, the ISD referenced in the signature metadata of the message). If the grace period (see Section 3.1.2.2.4) introduced by the latest TRC is still on-going, the root certificates in the second-to-latest TRC MUST also be included. For a description on how to build the correct collection of certificates, see Section 3.1.4.1.

2. If the signature metadata of the message contains the serial and base number of the latest TRC, the relying party MUST check that they have this latest TRC. If not, the relying party MUST request the latest TRC.

3. After constructing the pool of root certificates, the relying party MUST select the certificate chain used to verify the message. The AS certificate included in this certificate chain MUST have the following properties:

   * The ISD-AS number in the subject of the AS certificate MUST match the ISD-AS number in the signature metadata. See also Section 2.2.1.2.1.

   * The subject key identifier of the AS certificate MUST match the subject key identifier in the signature metadata. See also Section 2.2.2.2.

   * The AS certificate MUST be valid at verification time. Normally, this will be the current time. In special cases, e.g., auditing, the time can be set to the past to check if the message was verifiable at the given time.

4. After selecting a certificate chain to verify the control plane messages, the relying party MUST verify the certificate chain, by:

   * Executing the regular X.509 verification procedure. For details, see [X.509].

   * Checking that

     - all subjects of the certificates in the chain carry the same ISD number (see also Section 2.2.1.2.1,

     - each certificate is of the correct type (see also Section 2.1), and

     - the CA certificate validity period covers the AS certificate validity period.

5. If the verification of the certificate chain was successful, the relying party can now verify the control plane messages, with the root certificates from the certificate chain.

If any cryptographic material is missing in the process, the relying party MUST query the originator of the message for the missing material. If it cannot be resolved, the verification process fails.

Important: An implication of the above procedure is that path segments SHOULD be verifiable at time of use. It is not enough to rely on path segments being verified on insert since TRC updates that change the root key can invalidate a certificate chain.

4.3. Creating a New Control Plane AS Certificate

The steps REQUIRED to create a new AS certificate are the following:

1. The AS creates a new key pair and a certificate signing request (CSR) using that key pair.

2. The AS sends the certificate signing request to the relevant CA within the ISD.

3. The CA uses its CA key and the CSR to create the new AS certificate.

4. The CA sends the AS certificate back to the AS.

When an AS joins an ISD, the first CSR is sent out of band to one of the CAs as part of the formalities to join the ISD. Subsequent certificate renewals MAY be automated and can leverage the control plane communication infrastructure.

5. Security Considerations

The goal of SCION is to provide a secure inter-domain network architecture, therefore this section focuses on _inter_-AS security considerations. All _intra_-AS trust- and security aspects are out of scope.

5.1. Dependency on Certificates

In PKIs, CAs have both the responsibility and power to issue and revoke certificates. A compromised or misbehaving CA could refuse to issue certificates to legitimate entities and/or issue illegitimate certificates to allow impersonation of another entity. In the context of the Control Plane PKI, refusing to issue or renew a certificate to an AS will ultimately cut that AS off from the network, turning the Control Plane PKI into a potential network kill switch, so within each ISD there are usually multiple independent CAs.

SCION fundamentally differs from a global monopolistic trust model as each ISD manages its own trust roots instead of a single global entity providing those roots. This structure gives each ISD autonomy in terms of key management and in terms of trust, and prevents the occurrence of a global kill switch affecting all ISDs at once. However, each ISD is still susceptible to compromises that could affect or halt other components (control plane and forwarding).

5.1.1. Compromise of an ISD

In SCION there is no central authority that could "switch off" an ISD as each relies on its own independent trust roots. Each AS within an ISD is therefore dependant on its ISD's PKI for its functioning, although the following compromises are potentially possible:

* At TRC level: The private root keys of the root certificates contained in an TRC are used to sign CA certificates. If one of these private root keys is compromised, the adversary could issue illegitimate CA certificates which may be used in further attacks. To maliciously perform a TRC update, an attacker would need to compromise multiple voting keys, the number of which is dependent on the voting quorum set in the TRC - the higher the quorum, the more unlikely a malicious update will be.

* At CA level: The private keys of an ISD's CA certificates are used to sign the AS certificates and all ASes within an ISD obtain certificates directly from the CAs. If one of the CA' s keys is compromised, an adversary could issue illegitimate AS certificates, which may be used in further attacks.

* At AS level: Each AS within an ISD signs control plane messages with their AS private key. If the keys of an AS are compromised by an adversary, this adversary can illegitimately sign control plane messages including Path Construction Beacons (PCBs). This means that the adversary can manipulate the PCBs and propagate them to neighboring ASes or register/store them as path segments.

5.1.2. Recovery from Compromise

This section deals with possible recovery from the compromises discussed in the previous paragraph. As described in Section 1.4.2, there is no revocation in the Control Plane PKI.

* At TRC level: If any of the root keys or voting keys contained in the TRC are compromised, the TRC MUST be updated as described in Section 3.1.5. A trust reset is only required in the case the number of compromised keys at the same time is greater or equal than the TRC's quorum (see Section 3.1.2.2.7) and a invalid update has been produced and distributed in the network.

* At CA level: If the private key related to a CA certificate is compromised, the impacted CA AS MUST obtain a new CA certificate from the corresponding root AS. CA certificates are generally short lived to limit the impact of compromise. Alternatively, with a TRC update, a new root keys can also be forced, invalidating the compromised CA.

* At AS level: In the event of a key compromise of a (non-core) AS, the impacted AS needs to obtain a new certificate from its CA. This process will vary depending on internal issuance protocols.

5.2. Denial of Service Attacks

The Control Plane PKI lays the foundation for the authentication procedures in SCION by providing each AS within a specific ISD with a certified key pair. These keys enable the authentication of all control plane messages - every AS and endpoint can verify all control plane messages by following the certificate chain.

The relying party MUST be able to discover and obtain new or updated cryptographic material. For the control plane messages, this is simplified by the observation that the sender of a message (e.g. of a path construction beacon during path exploration or a path segment during a path lookup) always has all the cryptographic material to verify it. Thus, the receiver can always immediately obtain all the cryptographic material from the message originator. As the corresponding PKI messaging thus only occurs when the control plane is already communicating, these requests to obtain cryptographic material are not prone to additional denial of service attacks. We therefore refer to [I-D.dekater-scion-controlplane] for a more detailed description of DoS vulnerabilities of control-plane messages.

For certificate renewal, on the other hand, this does not apply. Denial of Service on the CA infrastructure or on the communication links from the individual ASes to the CA could be used by an attacker to prevent victim ASes from renewing their certificates, halting the path discovery process. This risk can be mitigated in multiple ways:

* CAs only need to be accessible from ASes within the ISD, reducing the potential DoS attack surface

* ISDs usually rely on multiple CAs

* ISDs could create policies and processes to renew certificates out-of-band

6. IANA Considerations

This document has no IANA actions.

The ISD and SCION AS number are SCION-specific numbers. They are currently allocated by Anapaya Systems, a provider of SCION-based networking software and solutions (see [ISD-AS-assignments-Anapaya]). This task is being transitioned from Anapaya to the SCION Association (see [ISD-AS-assignments]).

7. References

7.1. Normative References

[I-D.dekater-scion-controlplane]
de Kater, C., Rustignoli, N., and S. Hitz, "SCION Control Plane", Work in Progress, Internet-Draft, draft-dekater-scion-controlplane-09, 7 July 2025, <https://datatracker.ietf.org/doc/html/draft-dekater-scion-controlplane-09>.

[I-D.dekater-scion-dataplane]
de Kater, C., Rustignoli, N., Hugly, J., and S. Hitz, "SCION Data Plane", Work in Progress, Internet-Draft, draft-dekater-scion-dataplane-06, 7 July 2025, <https://datatracker.ietf.org/doc/html/draft-dekater-scion-dataplane-06>.

[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/rfc/rfc2119>.

[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/rfc/rfc5280>.

[RFC5480] Turner, S., Brown, D., Yiu, K., Housley, R., and T. Polk, "Elliptic Curve Cryptography Subject Public Key Information", RFC 5480, DOI 10.17487/RFC5480, March 2009, <https://www.rfc-editor.org/rfc/rfc5480>.

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

[RFC5758] Dang, Q., Santesson, S., Moriarty, K., Brown, D., and T. Polk, "Internet X.509 Public Key Infrastructure: Additional Algorithms and Identifiers for DSA and ECDSA", RFC 5758, DOI 10.17487/RFC5758, January 2010, <https://www.rfc-editor.org/rfc/rfc5758>.

[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/rfc/rfc8174>.

[RFC9217] Trammell, B., "Current Open Questions in Path-Aware Networking", RFC 9217, DOI 10.17487/RFC9217, March 2022, <https://www.rfc-editor.org/rfc/rfc9217>.

[X.509] "ITU-T X.509 (10/2016) | Information technology Open Systems Interconnection The Directory: Public-key and attribute certificate frameworks", January 2016, <https://handle.itu.int/11.1002/1000/13031>.

[X.680] "ITU-T X.680 (02/2021) | Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation", January 2021, <https://handle.itu.int/11.1002/1000/14468>.

[X.690] "ITU-T X.690 (02/2021) | Information technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)", January 2021, <https://handle.itu.int/11.1002/1000/14472>.

[X9.62] "ANSI X9.62-1998 | Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm", 1998.

7.2. Informative References

[BARRERA17] Barrera, D., Chuat, L., Perrig, A., Reischuk, R., and P. Szalachowski, "The SCION internet architecture", Association for Computing Machinery (ACM), Communications of the ACM vol. 60, no. 6, pp. 56-65, DOI 10.1145/3085591, May 2017, <https://doi.org/10.1145/3085591>.

[CHUAT22] Chuat, L., Legner, M., Basin, D., Hausheer, D., Hitz, S., Mueller, P., and A. Perrig, "The Complete Guide to SCION", ISBN 978-3-031-05287-3, 2022, <https://doi.org/10.1007/978-3-031-05288-0>.

[I-D.dekater-panrg-scion-overview]
de Kater, C., Rustignoli, N., and A. Perrig, "SCION Overview", Work in Progress, Internet-Draft, draft-dekater-panrg-scion-overview-06, 7 May 2024, <https://datatracker.ietf.org/doc/html/draft-dekater-panrg-scion-overview-06>.

[ISD-AS-assignments]
"SCION Registry", 2025, <http://scion.org/registry/>.

[ISD-AS-assignments-Anapaya]
"SCION ISD and AS Assignments", 2025, <https://docs.anapaya.net/en/latest/resources/isd-as-assignments/>.

[RFC5398] Huston, G., "Autonomous System (AS) Number Reservation for Documentation Use", RFC 5398, DOI 10.17487/RFC5398, December 2008, <https://www.rfc-editor.org/rfc/rfc5398>.

[RFC6996] Mitchell, J., "Autonomous System (AS) Reservation for Private Use", BCP 6, RFC 6996, DOI 10.17487/RFC6996, July 2013, <https://www.rfc-editor.org/rfc/rfc6996>.

[RFC8210] Bush, R. and R. Austein, "The Resource Public Key Infrastructure (RPKI) to Router Protocol, Version 1", RFC 8210, DOI 10.17487/RFC8210, September 2017, <https://www.rfc-editor.org/rfc/rfc8210>.

[SCIONLAB] Kown, J., Garca-Pardo, J., Legner, M., Wirz, F., Frei, M., Hausheer, D., and A. Perrig, "SCIONLAB - A Next-Generation Internet Testbed", 2020, <https://ieeexplore.ieee.org/abstract/document/9259355>.

Acknowledgments

Many thanks go to Fritz Steinmann (SIX Group AG), Juan A. Garcia Prado (ETH Zurich) and Russ Housley (IETF) for reviewing this document. We are also very grateful to Adrian Perrig (ETH Zurich), for providing guidance and feedback about each aspect of SCION. Finally, we are indebted to the SCION development teams of Anapaya and ETH Zurich, for their practical knowledge and for the documentation about the CP PKI, as well as to the authors of [CHUAT22] - the book is an important source of input and inspiration for this draft.

Deployment Testing: SCIONLab

SCIONLab is a global research network that is available to test the SCION architecture. You can create and use your ASes using your own computation resources which allows you to gain real-world experience of deploying and managing a SCION network.

More information can be found on the SCIONLab website and in the [SCIONLAB] paper.

Appendix A. Signing Ceremony Initial TRC

A Signing Ceremony is used to create the initial (first) Trust Root Configuration of an ISD. Each ISD may decide how to conduct this ceremony, but it is RECOMMENDED to establish a procedure similar to the one described below:

Ceremony Participants

The Signing Ceremony SHOULD include the following participants:

* *Ceremony Administrator* - an individual in charge of moderating the signing process, guiding the participants through the steps, and acting as an intermediary for sharing information. The Ceremony Administrator is typically appointed by the ISD Manager or by resolution of the Voting ASes.

* *Voting AS Representatives* - individuals representing each Voting AS who are able to create voting signatures on the TRC. They are in possession of a device with the private keys of their respective certificates in the TRC.

* *Witness(es)* - individual(s) who have no active role in the Signing Ceremony but may stop the process and request more information if they feel its integrity may have been compromised. The Witness(es) are typically appointed by resolution of the Voting ASes.

Note: The ISD members MUST decide on the roles of the Signing Ceremony participants in advance of Signing Ceremony, and MUST have reached agreement about the Certificate Authority (CA) ASes (that will also issue the root certificates). It is assumed that all parties are trustworthy and issues encountered during the Signing Ceremony may be assumed to be caused by honest mistakes and not by malicious intent. Hash comparison checks are included to counter mistakes and so that every participant can ensure they are operating on the same data, and the private keys of each participant never leave their machine. The Ceremony Administrator does not have to be entrusted with private keys.

Ceremony Preparations

The participants MUST decide in advance on the physical location of the Signing Ceremony, the devices that will be used, and the ISD policy as follows:

* ISD number - usually obtained from the SCION registry, see Section 3.1.2.2.2;

* The description of the TRC, see Section 3.1.2.2.10;

* Validity period of the TRC, see Section 3.1.2.2.3;

* Voting quorum for the TRC, see Section 3.1.2.2.7;

* AS numbers of the Core ASes, see Section 3.1.2.2.8;

* AS numbers of the Authoritative ASes, see Section 3.1.2.2.9;

* The list of Control Plane Root Certificates.

Each representative of a Voting AS MUST also create the following before the ceremony:

* A sensitive voting private key and a certificate holding the corresponding public key.

* A regular voting private key and a certificate holding the corresponding public key.

In addition, each Certificate Authority MUST create a control plane root private key and a certificate holding the corresponding public key. A representative of the Certificate Authority need not be present at the ceremony as they do not need to sign the TRC, but they MUST provide their root certificate to be shared at the ceremony. The validity period of the certificates generated in advance MUST cover the full TRC validity period.

The location MUST provide electricity and power sockets for each participant, and should provide a monitor or projector that allows the Ceremony Administrator to display proceedings.

The Ceremony Administrator and Voting ASes MUST each bring to the Signing Ceremony a secure machine capable of signing and verifying TRCs and computing the SHA-512 digest of the files. For voting ASes, the machine requires access to their own sensitive and regular voting private keys.

The Ceremony Administrator MUST provide or be provided with a device to exchange data between the ceremony participants.

The Signing Ceremony SHOULD include a procedure to verify that all devices are secure.

Ceremony Process

The number of Voting ASes present at the Signing Ceremony must be equal to or larger than the specified voting quorum.

The signing process has four phases of data sharing, led by the Ceremony Administrator who provides instructions to the other participants:

Phase 1: Certificate Exchange

All parties share the certificates that must be part of the TRC with the Ceremony Administrator. For the Voting ASes, these are the sensitive and the regular voting certificates, and for the Certificate Authority these are the Control Plane root certificates.

Each representative copies the requested certificates from their machine onto a data exchange device provided by the Ceremony Administrator that is passed between all representatives, before being returned to the Ceremony Administrator. Representatives MUST NOT copy the corresponding private keys onto the data exchange device as this invalidates the security of the ceremony.

The Ceremony Administrator then checks that the validity period of each provided certificate covers the previously agreed upon TRC validity, that the signature algorithms are correct, and that the certificate type is valid (root, sensitive voting or regular voting certificate). If these parameters are correct, the Ceremony Administrator computes the SHA256 hash value for each certificate, aggregates and bundles all the provided certificates, and finally calculates the SHA512 hash value for the entire bundle. All hash values must be displayed to the participants.

The Ceremony Administrator MUST then share the bundle with the representatives of the voting ASes who MUST validate on their machine that the hash value of their certificates and that of the bundled certificates is the same as displayed by the Ceremony Administrator.

Phase 1 concludes when every representative has confirmed the SHA256 sums are correct. If there is any mismatch then this phase MUST be repeated.

Phase 2: Generation of the TRC Payload

The Ceremony Administrator generates the TRC payload based on the bundled certificates and the Section 3.1.2.2 completed in accordance with ISD policy, see Appendix "Ceremony Preparations".

Once the voting representatives have verified the TRC data, the Ceremony Administrator computes the DER encoding of the data according to Section 3.1.2.1 and the SHA256 hash value of the TRC payload file. The TRC payload file is then shared with the voting representatives via the data exchange device who verify the TRC payload hash value by computing this on their machine and checking it matches the one displayed by the Ceremony Administrator.

Phase 2 concludes when all voting representatives confirm that the contents of the TRC payload are correct.

Phase 3: TRC Signing

Each voting representative attaches a signature created with their own private voting key to the TRC (payload file), using their own machine. This serves to prove possession of the private keys.

Phase 3 concludes when all voting representatives have attached their signatures to the TRC.

Phase 4: TRC Validation

All voting representatives copy the TRC payload signed with their private voting keys to the data exchange device and return this to the Ceremony Administrator. The Ceremony Administrator assembles the final TRC by aggregating the payload data and verifying the signatures based on the certificates exchanged during Phase 1. The Ceremony Administrator then shares the assembled TRC with all participants who MUST again inspect the signatures and verify them based on the certificates exchanged in Phase 1.

The Signing Ceremony is completed once when every voting representative confirms that the signatures match. All participants can then use the TRC to distribute trust anchors for the ISD.

Change Log

Changes made to drafts since ISE submission. This section is to be removed before publication.

draft-dekater-scion-pki-10

* removed ISD assignment table and replaced to reference in control-plane draft

* Updated number assignment reference

* Signatures: mention that other algorithms that ECDSA may be used in the future

* Figures: add SVG version

draft-dekater-scion-pki-09

* Signing ceremony and introduction - improved text

* Clarified why a CA must have an ISD-AS number assigned

* Mention Active Discovery as a TRC discovery mechanism

* Abstract: mention goal and that document is not an Internet Standard

draft-dekater-scion-pki-08

* Fix some oversized diagrams

* Introduction text rewording

draft-dekater-scion-pki-07

Minor changes:

* Clarified relationship with RPKI.

* Added this changelog

* General text editing

* References: fixed ITU, ANSI, Assigned ISD-AS, fixed cross-reference to text formatting in the CP draft

draft-dekater-scion-pki-06

Major changes:

* Added overview of SCION components to Introduction section.

Minor changes:

* General edits to make terminology consistent, remove duplication and rationalize text.

* Removed forward references.

* Added RFC2119 compliant terminology.

Authors' Addresses

Corine de Kater
SCION Association
Email: c_de_kater@gmx.ch

Nicola Rustignoli
SCION Association
Email: nic@scion.org

Samuel Hitz
Anapaya Systems
Email: hitz@anapaya.net