TPKI

A Transparent by Design Public Key Infrastructure Based on Short Lived Certificates

1. Abstract

This standard is a work in progress. Although portions of the standard have been fully defined and implemented, the document does not cover all possible operations for all objects. Until such time as this document does reflect all operations, portions of this document may be subject to change. At the time at which this document does reflect all operations, this note shall be removed.

This document describes a novel PKI based on short lived certificates.

2. Introduction

2.1. An Overview of Design Requirements

This section introduces the design concepts and motivations for the formulation of our solution. This section is not a formal treatment of the concepts. This section is a view into the authors' rationale while attempting to find a solution that works in practice.
The general requirements may be summarized as follows:

1. There must exist some form of centralized trust, but this centralized trust must not be consolidated in a single entity.
2. Revocation does not work in practice as implemented today. An alternative mechanism to revocation must be employed.
3. There must be an auditable system that must contain all valid certificates.
4. The system that stores all certificates must be capable of handling the volume of requests required to serve all requests by all users of the PKI OR the system that stores all certificates must be capable of allowing untrusted third parties to construct proofs about the contents of this system in a trustless manner.
5. A certificate that is not contained in the auditable system should not be treated as valid unless the object allows for an MTU.
6. The audit of issued certificates should be integral to the operation of the system itself.
7. The system must not depend on third parties that perform oversight without prompt and enforceable repercussion for failure.
8. A certificate must have a strongly collision resistant identifier that is inseparable from the issuer, recipient, and cryptographic key being used.
9. A certificate must not be a single polymorphic object that is intended for universal application.
10. The system should use an encoding that is easily parsed and human readable.

The issue of building fully decentralized identity and trust in the enterprise setting has no known solutions. Thus, there is no mechanism by which some form of centralized trust may be removed from a discussion of PKI in the enterprise setting. Given this constraint, a system must be designed to limit damage that may occur through abuse, mismanagement, or compromise of any such centralized entity. Further, the ability to revoke trust in a centralized entity must be an isolable event that does not cause catastrophic failure of the system as a whole. The hierarchy of trust must be well defined. This implies that cross signing of root level certificates should never be performed. In short, the graph of all certificates issued should be a set of independent directed acyclic graphs. This allows the consequence of expiration, revocation, and other actions to be well defined.

To address the problem of revocation, one possible solution is short lived certificates. Although this increases complexity, it also minimizes damage that results from a key compromise. Given the seeming inability of current PKI systems to address the problems around revocation, the authors of this paper believe something analogous to short lived certificates is the only rational solution.

Numbers two and three address the fact that centralized trust without oversight is the basis for abuse. Today, a root certificate may print a proof of identity for almost any other party without any reasonable means of prompt detection. This fraudulent proof will be nearly universally accepted and, as history has taught us, is highly unlikely to carry any consequence for the issuing authority. The only circumstances in which consequences are likely is under gross negligence of a catastrophic nature. Alternatively, history would also appear to indicate that the issuer may be held accountable if the certificates are fraudulently created in the name of entities with sufficient power to force a response.

The irony of numbers three, four, and five are reflected by the fact that the original x.509 standard depended on the existence of a global hierarchical database. This dependency assumption is apparent in the distinguished name field of x.509, but the x.509 standard never defined how such a database could be built. In limited defense of those authors, we are only now becoming aware of how to build such systems.

To address the problem of auditability, we may look to the concepts surrounding Certificate Transparency. Although certificate transparency does address some forms of fraudulent issuance, certificate transparency does not solve the problems of revocation. In order to address this fundamental requirement of a fault tolerant system, a replicated database may be used. By forcing the inclusion of all valid certificates into an auditable distributed database, fraud becomes substantially more complex. This forces a certificate authority to operate under public scrutiny and may be performed with lower operational penalty than revocation lists. How this is possible will be fully discussed later in this paper.

Short of being perfectly trustless in replication, the system must be designed such that it is both difficult and expensive to attempt a split view during replication. These statements may seem

weaker than many often cited descriptions of an ideally secure system, but the reality is there are no perfect solutions to this problem. The only rational response is to design the system such that there is limited benefit that may result from such action. So long as the probability of an undetectable split view attack is sufficiently small, and the consequences of detection are sufficiently large, we may at least limit how often such attempts are made. This may be of little comfort to the individual against which such an attack is leveraged, but it is the inversion of what we have accepted today. In the scenario we have described above, a mis-issuance attack could in theory be carried out against a limited subset of individuals for a short duration. As x.509 operates today, with the exception of systems that enforce certificate transparency or certificate pinning, a fraudulent certificate may be widely used while also existing nearly indefinitely without detection. A more formal handling of these concepts will be covered in subsequent sections of this paper.

Without the ability to replicate the database of all certificates in a secure manner, there are no known solutions that simultaneously address revocation, detection of mis-issuance, and still allow for functional operation without failure in practice. In one implementation, this failure may arise out of the same failure modes as OCSP, where a central authority is forced to handle a nearly unbounded number of requests. The mandate to perform this amount of work either forces an exorbitant cost for a single certificate, or forces a provider of such services to perform poorly at the assigned task. Alternatively, revocation lists must be propagated with no hope of ever being a functional system at scale.

In the setting of a distributed, replicated database that contains all valid certificates, the difficulties of revocation may be substantially reduced. The same properties that allow secure replication may also be leveraged to allow proofs of non-revocation to be constructed by many entities in parallel. There is a known solution for such a construction, namely blockchains. Notice the use of the plural form, because the reality is no single blockchain could handle the volume of writes required by a ubiquitous global system. This will require many scoped systems. Any solution that cites blockchain as a solution but does not handle the inherent limitations of write volumes will fail. Recent work around hierarchical blockchain systems allow these scoped blockchains to be anchored in a single blockchain. This is the solution we have implemented.

The requirement that there not be any single polymorphic object is based on the best practices of how to build extensible protocols that require substantial complexity for proper operation. For instance, a rational software engineer would never build a REST API that only contained a single route and performed all other actions using query parameters. This is simply a requirement to observe the well established concept that global variables are not good engineering practice.

In order to prevent the complex parsing logic of a single polymorphic certificate type, application specific objects should be used. Although this introduces many application specific objects, the strategy has two benefits. First, the extension of functionality in one object type does not have consequence on any other type that is not lower in the type hierarchy of the modified object.

Second, not all applications require all functionality. Rather than force a developer to implement a complex intertwined system of validation logic in order to enable the use of a small subset of the system, implementations may be limited in scope to only validate specific object types. Such systems must treat any object that is not fully implemented as invalid.

The capability of an object should be limited in scope and explicitly stated. Any capability not explicitly stated must never be granted. Although this may seem like an obvious choice to the uninitiated reader, x.509 has in previous iterations failed this fundamental design principle. Only a decade before the publication of this paper, it was trivial to perform privilege escalation in x.509 that allowed any user who possessed a valid end entity certificate to create a valid end entity certificate for any other user. This attack was not universal, but it was common enough that it caused significant problems until patched. The shocking basis for the attack was the omission of a field within an x.509 certificate and not due to the inadvertent inclusion of a field.

The canonical encoding of certificate objects should be based on a well defined standard that is not ambiguous and has many existing implementations. Given this requirement, we have selected JOSE as the basis for communicating cryptographically signed objects in a secure manner. Although other mechanisms exist, such as CBOR, these are not as well established at this time. Further, JOSE depends on an encoding scheme that is human readable after a single base 64 decoding.

3. Object Definitions

3.1. Root CA Certificate Objects

Certificate objects are JWS objects that may be used to establish identity within a TPKI. Root CA Certificates are self-signed objects that are assumed to be pre-loaded by end software that utilize this system, as is already implemented under x.509.

3.1.1. Root CA Polymorphic Certificate

3.1.1.1. Description

- This is the CA root certificate type that all intermediate certificates are issued under except those intermediate certificates that are used to issue non-repudiation certificates to end subjects.

3.1.1.2. Allowed Issuance Types

- Root CA Intermediate Polymorphic Certificate

3.1.1.3. Location in Blockchain

3.1.1.3.1. Account

- Keccak256(Claims.ACT.X)

3.1.1.3.2. Index

- FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

3.1.1.4. Fields

3.1.1.4.1. Header

3.1.1.4.1.1. HTP

- Type
  - String
- Requirements
  - MUST be present
  - MUST be equal to “mnx”
- Description
  - This field is the indication flag that the object MUST conform to this specification

3.1.1.4.1.2. PIN

- Type
  - Int64
- Requirements
  - MUST be present
  - MUST be a valid member of protocol enum
- Description
  - Protocol Identifier Number.

3.1.1.4.1.3. VER

- Type
  - String
- Requirements
  - MUST be present
  - MUST be a semver encoded object of the form <major>.<minor>.<revnum>
  - All numbers MUST be a nonnegative integer of size no greater than int64
- Description

- Defines the version number of the protocol

3.1.1.4.1.4. TYP

- Type
  - String
- Requirements
  - MUST be present
  - MUST be equal to string jws
- Description
  - Identifying object is JSON Web Signature

3.1.1.4.1.5. CTY

- Type
  - Int64
- Requirements
  - MUST be present
  - MUST be a valid member of object enum for protocol
- Description
  - Describes the object type for parsing logic. This is protocol specific.

3.1.1.4.1.6. PUB

- Type
  - Object: JWK
- Requirements
  - MUST be present
- Description
  - The JWK of the public key used to sign this object. This MUST be treated as the asymmetric cryptographic key that is used to sign subordinate certificates of this root certificate.

3.1.1.4.2. Claims

3.1.1.4.2.1. IAT

- Type
  - Int64
- Requirements
  - MUST be present
- Description
  - The issued at time encoded as a UTC0 Unix timestamp

3.1.1.4.2.2. RVK

- Type
  - Object: JWK
- Requirements
  - MUST be present
- Description
  - Hash of public key used for revocation issuance.

3.1.1.4.2.3. ACT

- Type
  - Object: JWK
- Requirements
  - MUST be present
  - At this time ACT MUST have KTY as EC, ALG as RecoverableECDSA, CRV as Secp256k1
- Description
  - The blockchain account where subordinate certificates will be written.

3.1.1.4.2.4. JTI

- Type
  - String
- Requirements
  - MUST be present
  - MUST be 32 bytes of url safe base 64
  - MUST not be all zeros when converted to byte format
- Description
  - Uniquely identifies this object

3.1.1.4.2.5. SUB

- Type
  - Object: Subject Description Object
- Requirements
  - MUST be present
  - MUST uniquely identify the subject
- Description
  - Uniquely defines the subject

3.1.1.4.3. Footer

- Signature

3.1.2. Root CA Non-Repudiation Root Certificate

3.1.2.1. Allowed Issuance Types

- Root CA Intermediate Non-Repudiation Certificate

3.1.2.2. Location in Blockchain

3.1.2.2.1. Account

- Keccak256(Claims.ACT.X)

3.1.2.2.2. Index

- FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

3.1.2.3. Fields

3.1.2.3.1. Header

3.1.2.3.1.1. HTP

- Type
  - String
- Requirements
  - MUST be present
  - MUST be equal to “mnx”
- Description
  - This field is the indication flag that the object MUST conform to this specification

3.1.2.3.1.2. PIN

- Type
  - Int64
- Requirements
  - MUST be present
  - MUST be a valid member of protocol enum
- Description
  - Protocol Identifier Number.

3.1.2.3.1.3. VER

- Type
  - String
- Requirements
  - MUST be present

    - MUST be a semver encoded object of the form <major>.<minor>.<revnum>
    - All numbers MUST be a nonnegative integer of size no greater than int64
- Description
    - Defines the version number of the protocol

3.1.2.3.1.4. TYP

- Type
    - String
- Requirements
    - MUST be present
    - MUST be equal to string jws
- Description
    - Identifying object is JSON Web Signature

3.1.2.3.1.5. CTY

- Type
    - Int64
- Requirements
    - MUST be present
    - MUST be a valid member of object enum for protocol
- Description
    - Describes the object type for parsing logic. This is protocol specific.

3.1.2.3.1.6. PUB

- Type
    - Object: JWK
- Requirements
    - MUST be present
- Description
    - The JWK of the public key used to sign this object. This MUST be treated as the asymmetric cryptographic key that is used to sign subordinate certificates of this root certificate.

3.1.2.3.2. Claims

3.1.2.3.2.1. IAT

- Type
    - Int64
- Requirements

    - MUST be present
- Description
    - The issued at time encoded as a UTC0 Unix timestamp

3.1.2.3.2.2. RVK

- Type
    - Object: JWK
- Requirements
    - MUST be present
- Description
    - Hash of public key used for revocation issuance.

3.1.2.3.2.3. ACT

- Type
    - Object: JWK
- Requirements
    - MUST be present
    - At this time ACT MUST have KTY as EC, ALG as RecoverableECDSA, CRV as Secp256k1
- Description
    - The blockchain account where subordinate certificates will be written.

3.1.2.3.2.4. JTI

- Type
    - String
- Requirements
    - MUST be present
    - MUST be 32 bytes of url safe base 64
    - MUST not be all zeros when converted to byte format
- Description
    - Uniquely identifies this object

3.1.2.3.2.5. SUB

- Type
    - Object: Subject Description Object
- Requirements
    - MUST be present
    - MUST uniquely identify the subject
- Description
    - Uniquely defines the subject

3.1.2.3.3. Footer

- Signature

3.2. Root CA Intermediate Certificate Objects

3.2.1. Root CA Intermediate Polymorphic Certificate

3.2.1.1. Description

- This is the CA intermediate root certificate that allows end entity certificates to be issued. This certificate type allows the issuance of all entity certificates except for non-repudiation certificates.

3.2.1.2. Allowed Issuance Types

- Entity-Level CA Polymorphic Certificate
- Entity-Level Key Negotiation Certificate
- Entity-Level Key Encipherment Certificate
- Entity-Level Data Encipherment Certificate
- Entity-Level Digital Signature Certificate

3.2.1.3. Location in Blockchain

3.2.1.3.1. Account

- Keccak256(Claims.ACT.X)

3.2.1.3.2. Index

- Claims.JTI

3.2.1.4. Fields

3.2.1.4.1. Header

3.2.1.4.1.1. HTP

- Type
  - String
- Requirements
  - MUST be present
  - MUST be equal to “mnx”
- Description

    - This field is the indication flag that the object MUST conform to this specification

3.2.1.4.1.2. PIN

- Type
    - Int64
- Requirements
    - MUST be a valid member of protocol enum
- Description
    - Protocol Identifier Number.

3.2.1.4.1.3. VER

- Type
    - String
- Requirements
    - MUST be present
    - MUST be a semver encoded object of the form <major>.<minor>.<revnum>
    - All numbers MUST be a nonnegative integer of size no greater than int64
- Description
    - Defines the version number of the protocol

3.2.1.4.1.4. TYP

- Type
    - String
- Requirements
    - MUST be equal to string jws
- Description
    - Identifying object is JSON Web Signature

3.2.1.4.1.5. CTY

- Type
    - Int64
- Requirements
    - MUST be a valid member of object enum for protocol
- Description
    - Describes the object type for parsing logic. This is protocol specific.

3.2.1.4.1.6. PUB

- Type

    - Object: JWK
- Requirements
    - MUST be present
- Description
    - The JWK of the public key used to sign this object. This MUST be treated as the asymmetric cryptographic key that is used to sign subordinate certificates of this certificate.

3.2.1.4.2. Claims

3.2.1.4.2.1. MPOVK

- Type
    - Object: JWK
- Requirements
    -
- Description
    - Merkle Proof of Validity Key Trie Root (MPOVK)

3.2.1.4.2.2. IAT

- Type
    - Int64
- Requirements
    - MUST be present
    - MUST be greater than IAT of parent certificate
- Description
    - The issued at time encoded as a UTC0 Unix timestamp

3.2.1.4.2.3. MII

- Type
    - String
- Requirements
    - MUST be a valid url safe base64 encoded string
    - MUST be 32 bytes before encoding to url safe base64
- Description
    - MPOVK Issuance Interval. The interval at which new validity statements will be signed and written to the blockchain. (12H?)

3.2.1.4.2.4. MVI

- Type
    - String
- Requirements
    - MUST be a valid url safe base64 encoded string

    - MUST be 32 bytes before encoding to url safe base64
- Description
    - MPOVK Validity Interval. The Amount of time each validity statement may be treated as valid for. (24H?)

3.2.1.4.2.5. SIK

- Type
    - Object: JWK
- Requirements
    - At this time JWK MUST have KTY as EC-HSH-PRE-HAL, ALG as RecoverableECDSA, CRV as Secp256k1
- Description
    - Specify the Public key that will be used to sign the objects

3.2.1.4.2.6. ACT

- Type
    - Object: JWK
- Requirements
    - At this time ACT MUST have KTY as EC, ALG as RecoverableECDSA, CRV as Secp256k1
- Description
    - Will reference the location where issued certificates and revocation objects will be written

3.2.1.4.2.7. JTI

- Type
    - String
- Requirements
    - MUST be present
    - MUST be 32 bytes of url safe base 64
    - MUST not be all zeros when converted to byte format
- Description
    - Uniquely identifies this object

3.2.1.4.2.8. SUB

- Type
    - Object: Subject Description Object
- Requirements
    - MUST be present
    - MUST uniquely identify the subject
- Description
    - Uniquely defines the subject

3.2.1.4.2.9. PACT

- Type
  - Object: JWK
- Requirements
  - At this time PACT MUST have KTY as EC, ALG as RecoverableECDSA, CRV as Secp256k1
- Description
  - Will reference the location where the parent certificate issued certificates and revocation objects are written

3.2.1.4.2.10. PJTI

- Type
  - String
- Requirements
  - MUST be present
  - MUST be 32 bytes of url safe base 64
  - MUST not be all zeros when converted to byte format
  - MUST uniquely identify the issuing certificate
- Description
  - References the JTI of the issuing certificate

3.2.1.4.2.11. ISS

- Description
  - Object: Subject Description Object
- Requirements
  - MUST be present
  - MUST be equal to the parent certificate's SUB
- Description
  - Uniquely defines the parent certificate

3.2.1.4.2.12. NKS

- Type
  - Int64
- Requirements
  - MUST be greater than 256
- Description
  - The number of keys that are contained in the MPOVK Trie

3.2.1.4.3. Footer

- Signature

3.2.2. Root CA Intermediate Non-Repudiation Certificate

3.2.2.1. Description

- This is the CA intermediate root certificate that allows entity-level non-repudiation certificates to be issued.

3.2.2.2. Allowed Issuance Types

- Entity-Level Non-Repudiation Certificate

3.2.2.3. Location in Blockchain

3.2.2.3.1. Account

- Keccak256(Claims.ACT.X)

3.2.2.3.2. Index

- Claims.JTI

3.2.2.4. Fields

3.2.2.4.1. Header

3.2.2.4.1.1. HTP

- Type
  - String
- Requirements
  - MUST be present
  - MUST be equal to “mnx”
- Description
  - This field is the indication flag that the object MUST conform to this specification

3.2.2.4.1.2. PIN

- Type
  - Int64
- Requirements
  - MUST be a valid member of protocol enum
- Description
  - Protocol Identifier Number.

3.2.2.4.1.3. VER

- Type
  - String
- Requirements
  - MUST be present
  - MUST be a semver encoded object of the form <major>.<minor>.<revnum>
  - All numbers MUST be a nonnegative integer of size no greater than int64
- Description
  - Defines the version number of the protocol

3.2.2.4.1.4. TYP

- Type
  - String
- Requirements
  - MUST be equal to string jws
- Description
  - Identifying object is JSON Web Signature

3.2.2.4.1.5. CTY

- Type
  - Int64
- Requirements
  - MUST be a valid member of object enum for protocol
- Description
  - Describes the object type for parsing logic. This is protocol specific.

3.2.2.4.1.6. PUB

- Type
  - Object: JWK
- Requirements
  - MUST be present
- Description
  - The JWK of the public key used to sign this object

3.2.2.4.2. Claims

3.2.2.4.2.1. MPOVK

- Type

    - Object: JWK
- Requirements
    -
- Description
    - Merkle Trie Key for Proof of Validity

3.2.2.4.2.2. IAT

- Type
    - Int64
- Requirements
    - MUST be present
    - MUST be greater than IAT of parent certificate
- Description
    - The issued at time encoded as a UTC0 Unix timestamp

3.2.2.4.2.3. MII

- Type
    - String
- Requirements
    - MUST be a valid url safe base64 encoded string
    - MUST be 32 bytes before encoding to url safe base64
- Description
    - MPOVK Issuance Interval (12H?)

3.2.2.4.2.4. MVI

- Type
    - String
- Requirements
    - MUST be a valid url safe base64 encoded string
    - MUST be 32 bytes before encoding to url safe base64
- Description
    - MPOVK Validity Interval (24H?)

3.2.2.4.2.5. SIK

- Type
    - Object: JWK
- Requirements
    - At this time JWK MUST have KTY as EC-HSH-PRE-HAL, ALG as RecoverableECDSA, CRV as Secp256k1
- Description
    - Specify the Public key that will be used to sign the objects

3.2.2.4.2.6. ACT

- Type
  - Object: JWK
- Requirements
  - At this time ACT MUST have KTY as EC, ALG as RecoverableECDSA, CRV as Secp256k1
- Description
  - The blockchain account where non-repudiation statements and subject initiated revocation objects will be written.

3.2.2.4.2.7. JTI

- Type
  - String
- Requirements
  - MUST be present
  - MUST be 32 bytes of url safe base 64
  - MUST not be all zeros when converted to byte format
- Description
  - Uniquely identifies this object

3.2.2.4.2.8. SUB

- Type
  - Object: Subject Description Object
- Requirements
  - MUST be present
  - MUST uniquely identify the subject
- Description
  - Uniquely defines the subject

3.2.2.4.2.9. PACT

- Type
  - Object: JWK
- Requirements
  - At this time PACT MUST have KTY as EC, ALG as RecoverableECDSA, CRV as Secp256k1
- Description
  - Will reference the location where the parent certificate issued certificates and revocation objects are written

3.2.2.4.2.10. PJTI

- Type

        - String
    - Requirements
        - MUST be present
        - MUST be 32 bytes of url safe base 64
        - MUST not be all zeros when converted to byte format
        - MUST uniquely identify the issuing certificate
    - Description
        - References the JTI of the issuing certificate

3.2.2.4.2.11. ISS

- Type
    - Object: Subject Description Object
- Requirements
    - MUST be present
    - MUST be equal to the parent certificate's SUB
- Description
    - Uniquely defines the parent certificate

3.2.2.4.2.12. NKS

- Type
    - Int64
- Requirements
    - MUST be greater than 256
- Description
    - The number of keys that are contained in the MPOVK Trie

3.2.2.4.3. Footer

- Signature

3.3. Entity-Level CA Certificates

3.3.1. Entity-Level CA Polymorphic Certificate

3.3.1.1. Allowed Issuance Types

- Entity-Level Key Negotiation Certificate
- Entity-Level Key Encipherment Certificate
- Entity-Level Data Encipherment Certificate
- Entity-Level Digital Signature Certificate

3.3.1.2. Fields

3.3.1.2.1. Header

3.3.1.2.1.1. HTP

- Type
  - String
- Requirements
  - MUST be present
  - MUST be equal to “mnx”
- Description
  - This field is the indication flag that the object MUST conform to this specification

3.3.1.2.1.2. PIN

- Type
  - Int64
- Requirements
  - MUST be a valid member of protocol enum
- Description
  - Protocol Identifier Number.

3.3.1.2.1.3. VER

- Type
  - String
- Requirements
  - MUST be present
  - MUST be a semver encoded object of the form <major>.<minor>.<revnum>
  - All numbers MUST be a nonnegative integer of size no greater than int64
- Description
  - Defines the version number of the protocol

3.3.1.2.1.4. TYP

- Type
  - String
- Requirements
  - MUST be equal to string jws
- Description
  - Identifying object is JSON Web Signature

3.3.1.2.1.5. CTY

- Type
  - Int64
- Requirements
  - MUST be a valid member of object enum for protocol
- Description
  - Describes the object type for parsing logic. This is protocol specific.

3.3.1.2.1.6. PUB

- Type
  - Object: JWK
- Requirements
  - MUST be present
- Description
  - The JWK of the public key used to sign this object

3.3.1.2.2. Claims

3.3.1.2.2.1. MPOVK

- Type
  - Object: JWK
- Requirements
  -
- Description
  - Merkle Trie Key for Proof of Validity

3.3.1.2.2.2. IAT

- Type
  - Int64
- Requirements
  - MUST be present
  - MUST be greater than IAT of parent certificate
- Description
  - The issued at time encoded as a UTC0 Unix timestamp

3.3.1.2.2.3. MII

- Type
  - String
- Requirements
  - MUST be a valid url safe base64 encoded string

    - MUST be 32 bytes before encoding to url safe base64
- Description
    - MPOVK Issuance Interval (12H?)

3.3.1.2.2.4. MVI

- Type
    - String
- Requirements
    - MUST be a valid url safe base64 encoded string
    - MUST be 32 bytes before encoding to url safe base64
- Description
    - MPOVK Validity Interval (24H?)

3.3.1.2.2.5. SIK

- Type
    - Object: JWK
- Requirements
    - At this time JWK MUST have KTY as EC-HSH-PRE-HAL, ALG as RecoverableECDSA, CRV as Secp256k1
- Description
    - Specify the Public key that will be used to sign the objects

3.3.1.2.2.6. ACT

- Type
    - Object: JWK
- Requirements
    - At this time ACT MUST have KTY as EC, ALG as RecoverableECDSA, CRV as Secp256k1
- Description
    - Will reference the location where issued certificates and revocation objects will be written

3.3.1.2.2.7. JTI

- Type
    - String
- Requirements
    - MUST be present
    - MUST be 32 bytes of url safe base 64
    - MUST not be all zeros when converted to byte format
- Description
    - Uniquely identifies this object

3.3.1.2.2.8. SUB

- Type
  - Object: Subject Description Object
- Requirements
  - MUST be present
  - MUST uniquely identify the subject
- Description
  - Uniquely defines the subject

3.3.1.2.2.9. PACT

- Type
  - Object: JWK
- Requirements
  - At this time PACT MUST have KTY as EC, ALG as RecoverableECDSA, CRV as Secp256k1
- Description
  - Will reference the location where the parent certificate issued certificates and revocation objects are written

3.3.1.2.2.10. PJTI

- Type
  - String
- Requirements
  - MUST be present
  - MUST be 32 bytes of url safe base 64
  - MUST not be all zeros when converted to byte format
  - MUST uniquely identify the issuing certificate
- Description
  - References the JTI of the issuing certificate

3.3.1.2.2.11. ISS

- Type
  - Object: Subject Description Object
- Requirements
  - MUST be present
  - MUST be equal to the parent certificate’s SUB
- Description
  - Uniquely defines the parent certificate

3.3.1.2.2.12. NKS

- Type

- Int64
- Requirements
  - MUST be greater than 256
- Description
  - The number of keys that are contained in the MPOVK Trie

TODO: There MUST be a domain restriction on this type of issued cert

3.3.1.2.3. Footer

- Signature

3.4. Entity-Level Application Specific Certificates

3.4.1. Entity-Level Key Negotiation Certificate

3.4.1.1. Fields

3.4.1.1.1. Header

3.4.1.1.1.1. HTP

- Type
  - String
- Requirements
  - MUST be present
  - MUST be equal to “mnx”
- Description
  - This field is the indication flag that the object MUST conform to this specification

3.4.1.1.1.2. PIN

- Type
  - Int64
- Requirements
  - MUST be a valid member of protocol enum
- Description
  - Protocol Identifier Number.

3.4.1.1.1.3. VER

- Type
  - String
- Requirements
  - MUST be present

    - MUST be a semver encoded object of the form <major>.<minor>.<revnum>
    - All numbers MUST be a nonnegative integer of size no greater than int64
- Description
    - Defines the version number of the protocol

3.4.1.1.1.4. TYP

- Type
    - String
- Requirements
    - MUST be equal to string jws
- Description
    - Identifying object is JSON Web Signature

3.4.1.1.1.5. CTY

- Type
    - Int64
- Requirements
    - MUST be a valid member of object enum for protocol
- Description
    - Describes the object type for parsing logic. This is protocol specific.

3.4.1.1.1.6. PUB

- Type
    - Object: JWK
- Requirements
    - MUST be present
- Description
    - The JWK of the public key used to sign this object

3.4.1.1.2. Claims

3.4.1.1.2.1. JTI

- Type
    - String
- Requirements
    - MUST be present
    - MUST be 32 bytes of url safe base 64
    - MUST not be all zeros when converted to byte format
- Description
    - Uniquely identifies this object

3.4.1.1.2.2. SUB

- Type
  - Object: Subject Description Object
- Requirements
  - MUST be present
  - MUST uniquely identify the subject
- Description
  - Uniquely defines the subject

3.4.1.1.2.3. PACT

- Type
  - Object: JWK
- Requirements
  - At this time PACT MUST have KTY as EC, ALG as RecoverableECDSA, CRV as Secp256k1
- Description
  - Will reference the location where the parent certificate issued certificates and revocation objects are written

3.4.1.1.2.4. PJTI

- Type
  - String
- Requirements
  - MUST be present
  - MUST be 32 bytes of url safe base 64
  - MUST not be all zeros when converted to byte format
  - MUST uniquely identify the issuing certificate
- Description
  - References the JTI of the issuing certificate

3.4.1.1.2.5. ISS

- Type
  - Object: Subject Description Object
- Requirements
  - MUST be present
  - MUST be equal to the parent certificate’s SUB
- Description
  - Uniquely defines the parent certificate

3.4.1.1.3. Footer

- Signature

3.4.2. Entity-Level Key Encipherment Certificate

3.4.2.1. Fields

3.4.2.1.1. Header

3.4.2.1.1.1. HTP

- Type
  - String
- Requirements
  - MUST be present
  - MUST be equal to “mnx”
- Description
  - This field is the indication flag that the object MUST conform to this specification

3.4.2.1.1.2. PIN

- Type
  - Int64
- Requirements
  - MUST be a valid member of protocol enum
- Description
  - Protocol Identifier Number.

3.4.2.1.1.3. VER

- Type
  - String
- Requirements
  - MUST be present
  - MUST be a semver encoded object of the form <major>.<minor>.<revnum>
  - All numbers MUST be a nonnegative integer of size no greater than int64
- Description
  - Defines the version number of the protocol

3.4.2.1.1.4. TYP

- Type
  - String
- Requirements
  - MUST be equal to string jws

- Description
  - Identifying object is JSON Web Signature

3.4.2.1.1.5. CTY

- Type
  - Int64
- Requirements
  - MUST be a valid member of object enum for protocol
- Description
  - Describes the object type for parsing logic. This is protocol specific.

3.4.2.1.1.6. PUB

- Type
  - Object: JWK
- Requirements
  - MUST be present
- Description
  - The JWK of the public key used to sign this object

3.4.2.1.2. Claims

3.4.2.1.2.1. ASK

- Type
  - Object: JWK
- Requirements
  - At this time JWK MUST have KTY as EC, ALG as ECDH, CRV as Curve25519
- Description
  - Specify the Public key that will be used for X25519

3.4.2.1.2.2. JTI

- Type
  - String
- Requirements
  - MUST be present
  - MUST be 32 bytes of url safe base 64
  - MUST not be all zeros when converted to byte format
- Description
  - Uniquely identifies this object

3.4.2.1.2.3. SUB

- Type
  - Object: Subject Description Object
- Requirements
  - MUST be present
  - MUST uniquely identify the subject
- Description
  - Uniquely defines the subject

3.4.2.1.2.4. PACT

- Type
  - Object: JWK
- Requirements
  - At this time PACT MUST have KTY as EC, ALG as RecoverableECDSA, CRV as Secp256k1
- Description
  - Will reference the location where the parent certificate issued certificates and revocation objects are written

3.4.2.1.2.5. PJTI

- Type
  - String
- Requirements
  - MUST be present
  - MUST be 32 bytes of url safe base 64
  - MUST not be all zeros when converted to byte format
  - MUST uniquely identify the issuing certificate
- Description
  - References the JTI of the issuing certificate

3.4.2.1.2.6. ISS

- Type
  - Object: Subject Description Object
- Requirements
  - MUST be present
  - MUST be equal to the parent certificate's SUB
- Description
  - Uniquely defines the parent certificate

3.4.2.1.3. Footer

- Signature

3.4.3. Entity-Level Data Encipherment Certificate

3.4.3.1. Fields

3.4.3.1.1. Header

3.4.3.1.1.1. HTP

- Type
  - String
- Requirements
  - MUST be present
  - MUST be equal to “mnx”
- Description
  - This field is the indication flag that the object MUST conform to this specification

3.4.3.1.1.2. PIN

- Type
  - Int64
- Requirements
  - MUST be a valid member of protocol enum
- Description
  - Protocol Identifier Number.

3.4.3.1.1.3. VER

- Type
  - String
- Requirements
  - MUST be present
  - MUST be a semver encoded object of the form <major>.<minor>.<revnum>
  - All numbers MUST be a nonnegative integer of size no greater than int64
- Description
  - Defines the version number of the protocol

3.4.3.1.1.4. TYP

- Type
  - String
- Requirements
  - MUST be equal to string jws

- Description
  - Identifying object is JSON Web Signature

3.4.3.1.1.5. CTY

- Type
  - Int64
- Requirements
  - MUST be a valid member of object enum for protocol
- Description
  - Describes the object type for parsing logic. This is protocol specific.

3.4.3.1.1.6. PUB

- Type
  - Object: JWK
- Requirements
  - MUST be present
- Description
  - The JWK of the public key used to sign this object

3.4.3.1.2. Claims

3.4.3.1.2.1. ASK

- Type
  - Object: JWK
- Requirements
  - At this time JWK MUST have KTY as EC, ALG as ECDH, CRV as Curve25519
- Description
  - Specify the Public key that will be used for X25519

3.4.3.1.2.2. JTI

- Type
  - String
- Requirements
  - MUST be present
  - MUST be 32 bytes of url safe base 64
  - MUST not be all zeros when converted to byte format
- Description
  - Uniquely identifies this object

3.4.3.1.2.3. SUB

- Type
  - Object: Subject Description Object
- Requirements
  - MUST be present
  - MUST uniquely identify the subject
- Description
  - Uniquely defines the subject

3.4.3.1.2.4. PACT

- Type
  - Object: JWK
- Requirements
  - At this time PACT MUST have KTY as EC, ALG as RecoverableECDSA, CRV as Secp256k1
- Description
  - Will reference the location where the parent certificate issued certificates and revocation objects are written

3.4.3.1.2.5. PJTI

- Type
  - String
- Requirements
  - MUST be present
  - MUST be 32 bytes of url safe base 64
  - MUST not be all zeros when converted to byte format
  - MUST uniquely identify the issuing certificate
- Description
  - References the JTI of the issuing certificate

3.4.3.1.2.6. ISS

- Type
  - Object: Subject Description Object
- Requirements
  - MUST be present
  - MUST be equal to the parent certificate’s SUB
- Description
  - Uniquely defines the parent certificate

3.4.3.2. Footer

- Signature

3.4.4. Entity-Level Digital Signature Certificate

3.4.4.1. Fields

3.4.4.1.1. Header

3.4.4.1.1.1. HTP

- Type
  - String
- Requirements
  - MUST be present
  - MUST be equal to “mnx”
- Description
  - This field is the indication flag that the object MUST conform to this specification

3.4.4.1.1.2. PIN

- Type
  - Int64
- Requirements
  - MUST be a valid member of protocol enum
- Description
  - Protocol Identifier Number.

3.4.4.1.1.3. VER

- Type
  - String
- Requirements
  - MUST be present
  - MUST be a semver encoded object of the form <major>.<minor>.<revnum>
  - All numbers MUST be a nonnegative integer of size no greater than int64
- Description
  - Defines the version number of the protocol

3.4.4.1.1.4. TYP

- Type
  - String
- Requirements
  - MUST be equal to string jws

- Description
    - Identifying object is JSON Web Signature

3.4.4.1.1.5. CTY

- Type
    - Int64
- Requirements
    - MUST be a valid member of object enum for protocol
- Description
    - Describes the object type for parsing logic. This is protocol specific.

3.4.4.1.1.6. PUB

- Type
    - Object: JWK
- Requirements
    - MUST be present
- Description
    - The JWK of the public key used to sign this object

3.4.4.1.2. Claims

3.4.4.1.2.1. SIK

- Type
    - Object: JWK
- Requirements
    - At this time JWK MUST have KTY as EC-HSH-PRE-HAL, ALG as RecoverableECDSA, CRV as Secp256k1
- Description
    - Specify the Public key that will be used to sign the objects

3.4.4.1.2.2. JTI

- Type
    - String
- Requirements
    - MUST be present
    - MUST be 32 bytes of url safe base 64
    - MUST not be all zeros when converted to byte format
- Description
    - Uniquely identifies this object

3.4.4.1.2.3. SUB

- Type
  - Object: Subject Description Object
- Requirements
  - MUST be present
  - MUST uniquely identify the subject
- Description
  - Uniquely defines the subject

3.4.4.1.2.4. PACT

- Type
  - Object: JWK
- Requirements
  - At this time PACT MUST have KTY as EC, ALG as RecoverableECDSA, CRV as Secp256k1
- Description
  - Will reference the location where the parent certificate issued certificates and revocation objects are written

3.4.4.1.2.5. PJTI

- Type
  - String
- Requirements
  - MUST be present
  - MUST be 32 bytes of url safe base 64
  - MUST not be all zeros when converted to byte format
  - MUST uniquely identify the issuing certificate
- Description
  - References the JTI of the issuing certificate

3.4.4.1.2.6. ISS

- Type
  - Object: Subject Description Object
- Requirements
  - MUST be present
  - MUST be equal to the parent certificate’s SUB
- Description
  - Uniquely defines the parent certificate

3.4.4.1.3. Footer

- Signature

3.4.5. Entity-Level Non-Repudiation Certificate

3.4.5.1. Description

- This entity-level certificate is able to issue non-repudiation statements that may be updated.

3.4.5.2. Location in Blockchain

3.4.5.2.1. Account

- Claims.PACT

3.4.5.2.2. Index

- Claims.JTI

3.4.5.3. Allowed Issuance Types

- Statement Head Object
- Statement Node Object

3.4.5.4. Fields

3.4.5.4.1. Header

3.4.5.4.1.1. HTP

- Type
  - String
- Requirements
  - MUST be present
  - MUST be equal to “mnx”
- Description
  - This field is the indication flag that the object MUST conform to this specification

3.4.5.4.1.2. PIN

- Type
  - Int64
- Requirements
  - MUST be a valid member of protocol enum
- Description
  - Protocol Identifier Number.

3.4.5.4.1.3. VER

- Type
  - String
- Requirements
  - MUST be present
  - MUST be a semver encoded object of the form <major>.<minor>.<revnum>
  - All numbers MUST be a nonnegative integer of size no greater than int64
- Description
  - Defines the version number of the protocol

3.4.5.4.1.4. TYP

- Type
  - String
- Requirements
  - MUST be equal to string jws
- Description
  - Identifying object is JSON Web Signature

3.4.5.4.1.5. CTY

- Type
  - Int64
- Requirements
  - MUST be a valid member of object enum for protocol
- Description
  - Describes the object type for parsing logic. This is protocol specific.

3.4.5.4.1.6. PUB

- Type
  - Object: JWK
- Requirements
  - MUST be present
- Description
  - The JWK of the public key used to sign this object

3.4.5.4.2. Claims

3.4.5.4.2.1. IAT

- Type

    - Int64
- Requirements
    - MUST be present
    - MUST be greater than IAT of parent certificate
- Description
    - The issued at time encoded as a UTC0 Unix timestamp

3.4.5.4.2.2. EXP

- Type
    - int64
- Requirements
    - MUST be present
    - MUST be greater than IAT
- Description
    - The expiration time encoded as a UTC0 Unix timestamp

3.4.5.4.2.3. RVK

- Type
    - Object: JWK
- Requirements
    -
- Description
    - Hash of public key used for revocation issuance.

3.4.5.4.2.4. RST

- Type
    - Object: RevStatus
- Requirements
    - Must be present
- Description
    - Revocation Status

3.4.5.4.2.5. SIK

- Type
    - Object: JWK
- Requirements
    - At this time JWK MUST have KTY as EC-HSH-PRE-HAL, ALG as RecoverableECDSA, CRV as Secp256k1
- Description
    - Specify the Public key that will be used to sign the objects

3.4.5.4.2.6. ACT

- Type
  - Object: JWK
- Requirements
  - At this time ACT MUST have KTY as EC, ALG as RecoverableECDSA, CRV as Secp256k1
- Description
  - Will reference the location where non-repudiation statements and revocation objects will be written

3.4.5.4.2.7. JTI

- Type
  - String
- Requirements
  - MUST be present
  - MUST be 32 bytes of url safe base 64
  - MUST not be all zeros when converted to byte format
- Description
  - Uniquely identifies this object

3.4.5.4.2.8. SUB

- Type
  - Object: Subject Description Object
- Requirements
  - MUST be present
  - MUST uniquely identify the subject
- Description
  - Uniquely defines the subject

3.4.5.4.2.9. PACT

- Type
  - Object: JWK
- Requirements
  - At this time PACT MUST have KTY as EC, ALG as RecoverableECDSA, CRV as Secp256k1
- Description
  - Will reference the location where the parent certificate issued certificates and revocation objects are written

3.4.5.4.2.10. PJTI

- Type

        - String
    - Requirements
        - MUST be present
        - MUST be 32 bytes of url safe base 64
        - MUST not be all zeros when converted to byte format
        - MUST uniquely identify the issuing certificate
    - Description
        - References the JTI of the issuing certificate

3.4.5.4.2.11. ISS

- Type
    - Object: Subject Description Object
- Requirements
    - MUST be present
    - MUST be equal to the parent certificate's SUB
- Description
    - Uniquely defines the parent certificate

3.4.5.4.3. Footer

- Signature of header and claims by signing key

3.5. Revocation Objects

3.5.1. Root Revocation Object

3.5.1.1. Description

- This is the Revocation Object revokes the corresponding Root CA Polymorphic Certificate (denoted throughout as RCAPC) and in doing so revokes all subordinate certificates. The hash of the public key Header.PUB equals RCAPC.Claims.RVK.HSH when using Claims.HAL hash algorithm.

3.5.1.2. Location in Blockchain

3.5.1.2.1. Account

- Keccak256(Claims.ACT.X)

3.5.1.2.1.1. Index

- FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

3.5.1.3. Fields

3.5.1.3.1. Header

3.5.1.3.1.1. HTP

- Type
  - String
- Requirements
  - MUST be present
  - MUST be equal to “mnx”
- Description
  - This field is the indication flag that the object MUST conform to this specification

3.5.1.3.1.2. PIN

- Type
  - Int64
- Requirements
  - MUST be present
  - MUST be a valid member of protocol enum
- Description
  - Protocol Identifier Number.

3.5.1.3.1.3. VER

- Type
  - String
- Requirements
  - MUST be present
  - MUST be a semver encoded object of the form <major>.<minor>.<revnum>
  - All numbers MUST be a nonnegative integer of size no greater than int64
- Description
  - Defines the version number of the protocol

3.5.1.3.1.4. TYP

- Type
  - String
- Requirements
  - MUST be present
  - MUST be equal to string jws
- Description

- Identifying object is JSON Web Signature

3.5.1.3.1.5. CTY

- Type
  - Int64
- Requirements
  - MUST be present
  - MUST be a valid member of object enum for protocol
- Description
  - Describes the object type for parsing logic. This is protocol specific.

3.5.1.3.1.6. PUB

- Type
  - Object: JWK
- Requirements
  - MUST be present
  - Hash of PUB MUST equal RCAPC.Claims.RVK.HSH when using RCAPC.Claims.RVK.HAL hash algorithm
- Description
  - The JWK of the public key used to sign this object. This MUST be treated as the asymmetric cryptographic key that is used to sign this certificate and is the revocation public key corresponding to RCAPC

3.5.1.3.2. Claims

3.5.1.3.2.1. IAT

- Type
  - Int64
- Requirements
  - MUST be present
  - MUST be greater than RCAPC.Claims.IAT
- Description
  - The issued at time encoded as a UTC0 Unix timestamp

3.5.1.3.2.2. ACT

- Type
  - Object: JWK
- Requirements
  - MUST be present
  - At this time ACT MUST have KTY as EC, ALG as RecoverableECDSA, CRV as Secp256k1

        - MUST be equal to RCAPC.Claims.ACT
    - Description
        - The blockchain account where the revocation object will be written

3.5.1.3.2.3. RJTI

- Type
    - String
- Requirements
    - MUST be present
    - MUST be 32 bytes of url safe base 64
    - MUST not be all zeros when converted to byte format
    - MUST be equal to RCAPC.Claims.JTI
- Description
    - Uniquely identifies the RCAPC object

3.5.1.3.2.4. RSUB

- Type
    - Object: Subject Description Object
- Requirements
    - MUST be present
    - MUST uniquely identify the subject
    - MUST be equal to RCAPC.Claims.SUB
- Description
    - Uniquely defines the RCAPC subject

3.5.1.3.2.5. JTI

- Type
    - String
- Requirements
    - MUST be present
    - MUST be 32 bytes of url safe base 64
    - MUST not be all zeros when converted to byte format
- Description
    - Uniquely identifies this object

3.5.1.3.2.6. SUB

- Type
    - Object: Subject Description Object
- Requirements
    - MUST be present
    - MUST uniquely identify the subject

- Description
  - Uniquely defines the subject

3.5.1.3.2.7. HAL

- Type
  - String
- Requirements
  - MUST be present
  - MUST equal RCAPC.Claims.RVK.HAL
- Description
  - Hash algorithm used for validating revocation public key

3.5.1.3.3. Footer

- Signature

3.5.2. NR Root Revocation Object

3.5.2.1. Description

- This is the Revocation Object revokes the corresponding Root CA Non-Repudiation Root Certificate (denoted throughout as RCANRRC) and in doing so revokes all subordinate certificates. The hash of the public key Header.PUB equals RCANRRC.Claims.RVK.HSH when using Claims.HAL hash algorithm.

3.5.2.2. Location in Blockchain

3.5.2.2.1. Account

- Keccak256(Claims.ACT.X)

3.5.2.2.1.1. Index

- FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

3.5.2.3. Fields

3.5.2.3.1. Header

3.5.2.3.1.1. HTP

- Type
  - String
- Requirements
  - MUST be present

    - MUST be equal to “mnx”
- Description
    - This field is the indication flag that the object MUST conform to this specification

3.5.2.3.1.2. PIN

- Type
    - Int64
- Requirements
    - MUST be present
    - MUST be a valid member of protocol enum
- Description
    - Protocol Identifier Number.

3.5.2.3.1.3. VER

- Type
    - String
- Requirements
    - MUST be present
    - MUST be a semver encoded object of the form <major>.<minor>.<revnum>
    - All numbers MUST be a nonnegative integer of size no greater than int64
- Description
    - Defines the version number of the protocol

3.5.2.3.1.4. TYP

- Type
    - String
- Requirements
    - MUST be present
    - MUST be equal to string jws
- Description
    - Identifying object is JSON Web Signature

3.5.2.3.1.5. CTY

- Type
    - Int64
- Requirements
    - MUST be present
    - MUST be a valid member of object enum for protocol
- Description

- Describes the object type for parsing logic. This is protocol specific.

3.5.2.3.1.6. PUB

- Type
  - Object: JWK
- Requirements
  - MUST be present
  - Hash of PUB MUST equal RCANRRC.Claims.RVK.HSH when using Claims.HAL hash algorithm
- Description
  - The JWK of the public key used to sign this object. This MUST be treated as the asymmetric cryptographic key that is used to sign this certificate and is the revocation public key corresponding to RCANRRC

3.5.2.3.2. Claims

3.5.2.3.2.1. IAT

- Type
  - Int64
- Requirements
  - MUST be present
  - MUST be greater than RCANRRC.Claims.IAT
- Description
  - The issued at time encoded as a UTC0 Unix timestamp

3.5.2.3.2.2. LVB

- Type
  - Uint32
- Requirements
  - MUST be present
- Description
  - The last valid block number; all statements issued after LVB MUST NOT be trusted.

3.5.2.3.2.3. ACT

- Type
  - Object: JWK
- Requirements
  - MUST be present
  - At this time ACT MUST have KTY as EC, ALG as RecoverableECDSA, CRV as Secp256k1

    - MUST be equal to RCANRRC.Claims.ACT
- Description
    - The blockchain account where the revocation object will be written

3.5.2.3.2.4. RJTI

- Type
    - String
- Requirements
    - MUST be present
    - MUST be 32 bytes of url safe base 64
    - MUST not be all zeros when converted to byte format
    - MUST be equal to RCANRRC.Claims.JTI
- Description
    - Uniquely identifies the RCANRRC object

3.5.2.3.2.5. RSUB

- Type
    - Object: Subject Description Object
- Requirements
    - MUST be present
    - MUST uniquely identify the subject
    - MUST be equal to RCANRRC.Claims.SUB
- Description
    - Uniquely defines the RCANRRC subject

3.5.2.3.2.6. JTI

- Type
    - String
- Requirements
    - MUST be present
    - MUST be 32 bytes of url safe base 64
    - MUST not be all zeros when converted to byte format
- Description
    - Uniquely identifies this object

3.5.2.3.2.7. SUB

- Type
    - Object: Subject Description Object
- Requirements
    - MUST be present
    - MUST uniquely identify the subject

- Description
  - Uniquely defines the subject

3.5.2.3.2.8. HAL

- Type
  - String
- Requirements
  - MUST be present
  - MUST equal RCAPC.Claims.RVK.HAL
- Description
  - Hash algorithm used for validating revocation public key

3.5.2.3.3. Footer

- Signature

3.6. Proof of Validity Objects

3.6.1. Validity Statement Object

3.6.1.1. Description

- Needs a description

3.6.1.2. Location in Blockchain

3.6.1.2.1. Account

- Acct = $AcctKey_{NRIntermediateCerAcctKey}$

3.6.1.2.2. Index

- Index = H($MPOVK\_Key_n$)

TODO: (Acct = $AcctKey_{NRIntermediateCerAcctKey}$ | Index = H($MPOVK\_Key_n$))

3.6.1.3. Fields

3.6.1.3.1. Header

3.6.1.3.1.1. HTP

- Type
  - String
- Requirements
  - MUST be present

    - MUST be equal to “mnx”
- Description
    - This field is the indication flag that the object MUST conform to this specification

3.6.1.3.1.2. PIN

- Type
    - Int64
- Requirements
    - MUST be a valid member of protocol enum
- Description
    - Protocol Identifier Number.

3.6.1.3.1.3. VER

- Type
    - String
- Requirements
    - MUST be present
    - MUST be a semver encoded object of the form <major>.<minor>.<revnum>
    - All numbers MUST be a nonnegative integer of size no greater than int64
- Description
    - Defines the version number of the protocol

3.6.1.3.1.4. TYP

- Type
    - String
- Requirements
    - MUST be equal to string jws
- Description
    - Identifying object is JSON Web Signature

3.6.1.3.1.5. CTY

- Type
    - Int64
- Requirements
    - MUST be a valid member of object enum for protocol
- Description
    - Describes the object type for parsing logic. This is protocol specific.

3.6.1.3.1.6. PUB

- Type
  - Object: JWK
- Requirements
  - MUST be present
- Description
  - The JWK of the public key used to sign this object. This MUST be treated as the asymmetric cryptographic key that is used to sign subordinate certificates of this certificate.

3.6.1.3.2. Claims

3.6.1.3.2.1. MPOVK

- Type
  - Object: JWK
- Requirements
  -
- Description
  - Merkle Proof of Validity Key Trie Root (MPOVK)

3.6.1.3.2.2. IAT

- Type
  - Int64
- Requirements
  - MUST be present
  - MUST be greater than IAT of parent certificate
- Description
  - The issued at time encoded as a UTC0 Unix timestamp

3.6.1.3.2.3. RUT

- Type
  - String
- Requirements
  - MUST be present
  - MUST be less than 64 bytes in length before encoding into url safe base64
- Description
  - Sparse Merkle Trie root hash

3.6.1.3.2.4. POI

- Type

    - list<String>
- Requirements
    - MUST be present
- Description
    - Compact proof of inclusion in SMT Root

TODO: Needs to be fixed

3.6.1.3.2.5. ACT

- Type
    - Object: JWK
- Requirements
    - At this time ACT MUST have KTY as EC, ALG as RecoverableECDSA, CRV as Secp256k1
- Description
    - Will reference the location where issued certificates and revocation objects will be written

3.6.1.3.2.6. JTI

- Type
    - String
- Requirements
    - MUST be present
    - MUST be 32 bytes of url safe base 64
    - MUST not be all zeros when converted to byte format
- Description
    - Uniquely identifies this object

3.6.1.3.2.7. SUB

- Type
    - Object: Subject Description Object
- Requirements
    - MUST be present
    - MUST uniquely identify the subject
- Description
    - Uniquely defines the subject

3.6.1.3.2.8. PACT

- Type
    - Object: JWK
- Requirements
    - At this time PACT MUST have KTY as EC, ALG as RecoverableECDSA, CRV as Secp256k1

- Description
    - Will reference the location where the parent certificate issued certificates and revocation objects are written

3.6.1.3.2.9. PJTI

- Type
    - String
- Requirements
    - MUST be present
    - MUST be 32 bytes of url safe base 64
    - MUST not be all zeros when converted to byte format
    - MUST uniquely identify the issuing certificate
- Description
    - References the JTI of the issuing certificate

3.6.1.3.2.10. ISS

- Description
    - Object: Subject Description Object
- Requirements
    - MUST be present
    - MUST be equal to the parent certificate’s SUB
- Description
    - Uniquely defines the parent certificate

3.6.1.3.2.11. HAL

- Type
    - String
- Requirements
    - MUST be present
- Description
    - Hash algorithm used for computing index and better
    - TODO: make better description statement

3.6.1.3.3. Footer

- Signature

SMT POI For MPOVK_Key$_n$
Object SMT Root Hash
TODO: Should be confirmed that the above are satisfied

3.7. Primitive Objects

3.7.1. JWK Object

3.7.1.1. Description

- Needs a description

3.7.1.2. Fields

3.7.1.2.1. ALG

- Type
  - String
- Requirements
  - MUST be one of EdDSA or RecoverableECDSA (In future ECDH, and/or ECIES) at this time
  - MUST be less than 32 characters for any ALG defined in the future
- Description
  - Defines the signature or encryption algorithm

3.7.1.2.2. KTY

- Type
  - String
- Requirements
  - MUST be present
  - REQUIRED to be EC, EC-HSH-PRE-HAL, EC-HSH-REV-HAL, EC-SMT-RUT-HSH-HAL, EC-SMT-CHLD-HSH-HAL at this time
  - MUST be less than 32 characters
- Description
  - Describes the type of cryptographic algorithm used. HAL is Hash Algorithm and is specified in HAL

3.7.1.2.3. CRV

- Type
  - String
- Requirements
  - MUST be present
  - MUST be Curve25519 or Secp256k1 at this time
  - MUST be less than 16 characters
- Description

    - Describes the Elliptic Curve in use.

3.7.1.2.4. X

- Type
    - String
- Requirements
    - Required IFF KTY is EC OR EC-HSH-REV-HAL OR EC-SMT-CHLD-HSH-HAL
    - Url safe Base64 encoded string
    - MUST be less than 75 characters
- Description
    - Encode the X coordinate of the public key

3.7.1.2.5. Y

- Type
    - String
- Requirements
    - Optional Based on ALG
    - Url safe Base64 encoded string
    - MUST be less than 75 characters
- Description
    - Encode the Y coordinate of the public key

3.7.1.2.6. XCU

- Type
    - String
- Requirements
    - MUST be present
    - MUST contain a valid URI
    - This URI should use HTTPS
    - MUST be less than 2048 characters
    - Should be far less than 2048
- Description
    - Where authorizing JWS may be acquired

3.7.1.2.7. HSH

- Type
    - String
- Requirements
    - Required IFF KTY is of type EC-PRE-HSH-HAL OR EC-REV-HSH-HAL

    - MUST be less than 64 bytes in length before encoding into url safe base64
- Description
    - Output of hash function

3.7.1.2.8. HAL

- Type
    - String
- Requirements
    - MUST be Sha256 or Keccak256 at this time
    - Required IFF KTY is EC-HSH-PRE-HAL, EC-HSH-REV-HAL, EC-SMT-RUT-HSH-HAL, EC-SMT-CHLD-HSH-HAL
    - MUST be less than 32 characters in length
- Description
    - Specified hash algorithm

3.7.1.2.9. RUT

- Type
    - String
- Requirements
    - Required IFF KTY is of type EC-SMT-RUT-HSH-HAL OR EC-SMT-CHLD-HSH-HAL
    - MUST be less than 64 bytes in length before encoding into url safe base64
- Description
    - Merkle Root Hash of a Sparse Merkle Trie of ordered JWK objects.

3.7.1.2.10. NMK

- Type
    - Int64
- Requirements
    - MUST be greater than zero
    - MUST be present IFF KTY is EC-SMT-CHLD-HSH-HAL OR EC-SMT-RUT-HSH-HAL
- Description
    -

3.7.2. Subject Description Object

3.7.2.1. Description

- Unique descriptor of entities that may have multiple objects yet should be grouped together

3.7.2.2. Exclusive Fields

3.7.2.2.1. LEI

- Type
  - String
- Requirements
  - MUST be valid base36 string
  - MUST only be used in the context of non-repudiation
  - MUST be 20 characters long
  - Characters 1 through 4 MUST be for Local Operating Unit
  - Characters 5 through 18 MUST be entity identifier
  - Characters 19 and 20 MUST serve as checksum
  - To ensure valid LEI checksum, convert the 20-character alphanumeric string from base 36 encoding into a base 10 integer. LEI checksum is valid if integer modulo 97 is 1. Example: ‘54930084UKLMY22DS16’ == 549300843020213122342213281 6 == 1 mod 97.

3.7.2.2.2. Domain

- UNDEFINED AT THIS TIME

3.7.2.2.3. Individual

- UNDEFINED AT THIS TIME

3.7.2.2.4. Machine

- UNDEFINED AT THIS TIME

3.7.2.2.5. Authority

- Type
  - String
- Requirements
  - MUST be the Subject Description Object Type for Root Certificate Authorities
  - MUST be less than or equal to 32 Characters
  - MUST uniquely identify the issuing root certificate authority

- MUST be human readable

3.8. Application Specific Protocol Objects

3.8.1. Fundamental Objects

3.8.1.1. Protocol Identifier Object

3.8.1.1.1. Description

- Identifies a sub protocol being utilized in a blockchain account space

3.8.1.1.2. Location in Blockchain

3.8.1.1.2.1. Account

- Acct = $AcctKey_{Entity}$

3.8.1.1.2.2. Index

- FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

TODO: (Acct = $AcctKey_{Entity}$ | Index = All Ones)

3.8.1.1.3. Required Fields

3.8.1.1.3.1. PIN

- Type
  - Int64
- Requirements
  - MUST be a valid member of protocol enum
- Description
  - Protocol Identifier Number.

3.9. Accountable Omission and Revision NR Protocol

3.9.1. Statement Head Object

3.9.1.1. Description

- Needs a description

3.9.1.2. Location in Blockchain

3.9.1.2.1. Account

- Claims.PACT

3.9.1.2.2. Index

- Claims.JTI

3.9.1.3. Fields

3.9.1.3.1. Header

3.9.1.3.1.1. HTP

- Type
  - String
- Requirements
  - MUST be present
  - MUST be equal to “mnx”
- Description
  - This field is the indication flag that the object MUST conform to this specification

3.9.1.3.1.2. PIN

- Type
  - Int64
- Requirements
  - MUST be a valid member of protocol enum
- Description
  - Protocol Identifier Number.

3.9.1.3.1.3. VER

- Type
  - String
- Requirements
  - MUST be present
  - MUST be a semver encoded object of the form <major>.<minor>.<revnum>
  - All numbers MUST be a nonnegative integer of size no greater than int64
- Description
  - Defines the version number of the protocol

3.9.1.3.1.4. TYP

- Type
  - String
- Requirements
  - MUST be equal to string jws
- Description
  - Identifying object is JSON Web Signature

3.9.1.3.1.5. CTY

- Type
  - Int64
- Requirements
  - MUST be a valid member of object enum for protocol
- Description
  - Describes the object type for parsing logic. This is protocol specific.

3.9.1.3.1.6. PUB

- Type
  - Object: JWK
- Requirements
  - MUST be present
- Description
  - The JWK of the public key used to sign this object

3.9.1.3.2. Claims

3.9.1.3.2.1. PACT

- Type
  - Object: JWK
- Requirements
  - MUST be present
  - At this time PACT MUST have KTY as EC, ALG as RecoverableECDSA, CRV as Secp256k1
- Description
  - Will reference the location where the parent certificate issued certificates and objects are written

3.9.1.3.2.2. JTI

- Type
  - String

- Requirements
  - MUST be present
  - MUST be 32 bytes of url safe base 64
  - MUST be equal to the SMT Root of version zero
  - MUST not be all zeros when converted to byte format
- Description
  - SMT Merkle Root of Content Trie

3.9.1.3.2.3. RevNum

- Type
  - Int64
- Requirements
  - MUST be present
  - MUST be greater than or equal to zero
- Description
  - The current revision number of the document

3.9.1.4. Footer

- Signature

3.9.2. Statement Node Object

3.9.2.1. Description

- Needs a description

3.9.2.2. Location in Blockchain

3.9.2.2.1. Account

- Claims.PACT

3.9.2.2.2. Index

- Claims.HAL(Claims.JTI | Claims.RevNum)

3.9.2.3. Fields

3.9.2.3.1. Header

3.9.2.3.1.1. HTP

- Type
  - String
- Requirements
  - MUST be present

    - MUST be equal to “mnx”
- Description
    - This field is the indication flag that the object MUST conform to this specification

3.9.2.3.1.2. PIN

- Type
    - Int64
- Requirements
    - MUST be a valid member of protocol enum
- Description
    - Protocol Identifier Number.

3.9.2.3.1.3. VER

- Type
    - String
- Requirements
    - MUST be present
    - MUST be a semver encoded object of the form <major>.<minor>.<revnum>
    - All numbers MUST be a nonnegative integer of size no greater than int64
- Description
    - Defines the version number of the protocol

3.9.2.3.1.4. TYP

- Type
    - String
- Requirements
    - MUST be equal to string jws
- Description
    - Identifying object is JSON Web Signature

3.9.2.3.1.5. CTY

- Type
    - Int64
- Requirements
    - MUST be a valid member of object enum for protocol
- Description
    - Describes the object type for parsing logic. This is protocol specific.

3.9.2.3.1.6. PUB

- Type
  - Object: JWK
- Requirements
  - MUST be present
- Description
  - The JWK of the public key used to sign this object

3.9.2.3.2. Claims

3.9.2.3.2.1. IAT

- Type
  - Int64
- Requirements
  - MUST be present
  - MUST be greater than IAT of parent certificate
- Description
  - The issued at time encoded as a UTC0 Unix timestamp

3.9.2.3.2.2. EXP

- Type
  - Int64
- Requirements
  - MUST be present
  - MUST be greater than IAT
- Description
  - The expiration time encoded as a UTC0 Unix timestamp

3.9.2.3.2.3. MMD

- Type
  - Int64
- Requirements
  - MUST be present
  - MUST be less than or equal to 604800
- Description
  - The time duration after which a validity proof MUST be presented upon request for proof of inclusion in the blockchain.

3.9.2.3.2.4. JTI

- Type

    - String
- Requirements
    - MUST be present
    - MUST be 32 bytes of url safe base 64
    - MUST be equal to the $SMTRoot_0$
    - MUST not be all zeros when converted to byte format
- Description
    - SMT Merkle Root of Content Trie

3.9.2.3.2.5. PACT

- Type
    - Object: JWK
- Requirements
    - At this time PACT MUST have KTY as EC, ALG as RecoverableECDSA, CRV as Secp256k1
- Description
    - Will reference the location where the parent certificate issued certificates and objects are written

3.9.2.3.2.6. PJTI

- Type
    - String
- Requirements
    - MUST be present
    - MUST be 32 bytes of url safe base 64
    - MUST not be all zeros when converted to byte format
    - MUST uniquely identify the issuing certificate
- Description
    - References the JTI of the issuing certificate

3.9.2.3.2.7. ISS

- Type
    - Object: Subject Description Object
- Requirements
    - MUST be present
    - MUST be equal to the parent certificate’s SUB
- Description
    - Uniquely defines the parent certificate

3.9.2.3.2.8. HAL

- Type
    - String

- Requirements
    - MUST be present
- Description
    - Hash algorithm used for computing index and better
    - TODO: make better description statement

3.9.2.3.2.9. CJU

- Type
    - list<String>
- Requirements
    - MUST be present
    - MUST contain valid URIs
    - These URIs should use HTTPS
    - MUST be less than 2048 characters
    - Should be far less than 2048
- Description
    - The URI at which the associated content for this JWS may be obtained until either this object is revoked OR this object expires.

3.9.2.3.2.10. VSU

- Type
    - list<String>
- Requirements
    - MUST be present
    - MUST contain valid URIs
    - These URIs should use HTTPS
    - MUST be less than 2048 characters
    - Should be far less than 2048
- Description
    - Where the Validity Statement(s) may be acquired for proof of inclusion in the blockchain at a point in time.

3.9.2.3.2.11. HAU

- Type
    - list<String>
- Requirements
    - MUST be present
    - MUST contain valid URIs
    - These URIs SHOULD use HTTPS
    - MUST be less than 2048 characters

    - SHOULD be as few characters as needed to prevent collision. This SHOULD be a uri that contains two parameters or two path designators that are at least 32 bytes each of raw data before url safe base 64 encoding. This is to allow the ACCTKey and JTI to be fully defined.
- Description
    - Where the Head object that is unique to this JTI may be acquired

3.9.2.3.2.12. RevNum

- Type
    - Int64
- Requirements
    - MUST be present
    - MUST be greater than or equal to PRN
- Description
    - The current revision number of the document

3.9.2.3.2.13. PRN

- Type
    - Int64
- Requirements
    - MUST be present
    - MUST be greater than or equal to zero
- Description
    - The previous valid revision number for this document.

3.9.2.3.2.14. CHA

- Type
    - Int64
- Requirements
    - MUST be present
    - MUST be an allowable type as defined by chunking enumeration
- Description
    - Defines the chunking algorithm used to chunk content

3.9.2.3.2.15. SRT

- Type
    - String
- Requirements
    - MUST be present

    - MUST be 32 bytes of url safe base 64
    - When chunking logic is applied to content and data is loaded into Content SMT, this value MUST be equal to the resultant root hash of the Content SMT
- Description
    - The root hash of the content Sparse Merkle Trie

3.9.2.3.2.16. PCU

- Type
    - list<String>
- Requirements
    - MUST be present
    - MUST contain valid URIs
    - These URIs SHOULD use HTTPS
    - MUST be less than 2048 characters
    - SHOULD be as few characters as needed to prevent collision. This SHOULD be a uri that contains two parameters or two path designators that are at least 32 bytes each of raw data before url safe base 64 encoding. This is to allow the ACCTKey and JTI to be fully defined.
- Description
    - Where the content for the previous version, as referenced by PRN, may be acquired (optional)

3.9.2.3.2.17. PJU

- Type
    - list<String>
- Requirements
    - MUST be present
    - MUST contain valid URIs
    - These URIs SHOULD use HTTPS
    - MUST be less than 2048 characters
    - SHOULD be as few characters as needed to prevent collision. This SHOULD be a uri that contains two parameters or two path designators that are at least 32 bytes each of raw data before url safe base 64 encoding. This is to allow the ACCTKey and JTI to be fully defined.
- Description
    - Where the JWS for the previous version, as referenced by PRN, may be acquired (optional)

3.9.2.3.3. Footer

- Signature

3.9.3. Object Type Enumeration

- Type:
  - Int64
- Requirements:
  - MUST be non-negative
- Type enumeration:
  - TODO: All the types should be listed here
- Description:
  -

Fundamental type

int64

Requirements

Non negative

Type enumeration

…. Name all the types

3.9.4. IssuanceObj

- Type:
  - list<Int64>
- Requirements:
  - Array length MUST be less than 16 elements
- Description:
  -

3.9.5. Non-repudiation revocation

3.9.5.1. Description

-

3.9.5.2. Location in Blockchain

3.9.5.2.1. Account

-

3.9.5.2.2. Index

-

3.9.5.3. Fields

3.9.5.3.1. IsRevoked

- Type
  - bool
- Requirements
  - MUST be present
- Description
  -

3.9.2.3.2. TimeRevoked

- Type
  - time.Time
- Requirements
  - if isRevoked is true, this should be populated with non default values
- Description
  - time when isRevoked became true

3.9.2.3.3. MostRecentValidHash

- Type
  - byte32
- Requirements
  - if isRevoked is true, this should be populated with non default values
- Description
  - the most recent hash value before isRevoked became true