Delft University of Technology

SAT.509

Towards Minimal Certificates for Federated Space Public Key Infrastructure

Alin-Petru Roșu

esa

TUDelft

SAT.509

Towards Minimal Certificates for Federated Space Public Key Infrastructure

by

Alin-Petru Roșu

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on Tuesday July 22, 2025 at 10:00 AM.

Student number:	5772176	
Project duration:	October 16, 2024 – July 22, 2025	
Thesis committee:	Prof. dr. G. Smaragdakis	TU Delft, Thesis advisor
	Dr. E. A. Markatou	TU Delft, Daily supervisor
	Dr. A. Costea	TU Delft
	Dr. B. K. Özkan	TU Delft
	Dr. O. A. Graur	European Space Agency

Cover:	AI-generated illustration using OpenAI's DALL·E
Style:	TU Delft Report Style, with modifications by Daan Zwaneveld

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

Preface

If someone had told me years ago all the stories about how I would one day write this thesis, I am not sure I would have believed them. The road to this point has been anything but straightforward—marked by setbacks, sudden changes, and moments when giving up felt easier than moving forward. Yet my conviction that my work should have purpose sustained me, allowing me to close a chapter that has taught me more about myself than I ever imagined.

I owe sincere gratitude to Professor Georgios Smaragdakis for being a steady presence and source of unconditional support throughout my studies, always ready to offer guidance and encouragement during my most challenging moments, often in subtle yet profoundly impactful ways.

I am thankful to Professor Lilika Markatou for her supervision, patience, and for granting me the freedom to pursue my ideas, both within this thesis and in our separate research projects. Her attention to detail and thorough feedback greatly enriched my work.

My heartfelt appreciation goes to Oana Graur, to whom I am deeply indebted. Her trust in me, collaborative spirit, and constant encouragement made a tremendous difference in my journey. Her support went far beyond academic guidance—her belief in my work and her willingness to open new opportunities were essential to the success of this thesis.

I am grateful to both Professor Lilika Markatou and Professor Zekeriya Erkin for providing me with research positions that enabled me to further my learning and complete my studies. Their support was instrumental in reaching this milestone.

I sincerely thank Antonios Atlasis for his trust and for enabling me to continue working with the European Space Agency beyond my thesis.

I am truly fortunate to have had such wonderful friends and colleagues at TU Delft—especially Sandra Wolff, for her empathy and laughter, and Tjitske Koster and Maarten Weyns, for making the Netherlands feel like home. To Tamara Tataru—for her teamwork and dedication—and to everyone who shared this chapter with me: thank you for the memories and support.

I am profoundly grateful to my family for their unwavering belief in me, for their presence and encouragement, even from afar. And to those back home and here in the Netherlands who kept me grounded, your guidance and support helped me through the most challenging moments, reminding me that I was never alone.

Above all, I thank God for granting me the resilience and purpose to see this journey through.

This thesis is not just a document; it is a piece of my story, shaped by every person who walked this path with me. Ultimately, it is dedicated to those who believe in purposeful work and to the ongoing evolution—of technology, of knowledge, and of ourselves.

This work was supported by Objective Systems, Inc., by providing a temporary license for their tooling.

Alin-Petru Rosu
Delft, July 2025

Abstract

Federated Space Public Key Infrastructure (PKI) can offer a scalable foundation for secure and interoperable communications in collaborative space missions. Yet, its deployment faces challenges stemming from resource-constrained assets, architectural complexity, and the transition to post-quantum (PQ) cryptography. Current CCSDS space guidelines rely on the Internet X.509 profile, whose extensive feature set—if left unrestricted—can increase implementation complexity, certificate size (especially under PQ algorithms), and the risk of interoperability issues. In parallel, the IETF C509 Certificates draft emerges as a streamlined subset of X.509 with a compact encoding specifically tailored for constrained environments. This paper provides an empirical comparison between X.509 and C509 to inform space mission designers about the associated advantages and costs of each, specifically when PQ cryptography is incorporated into space PKIs. To help pave the way for interoperability in federated space missions, a minimal certificate profile for space PKI is proposed.

In addition, the work introduces the first open-source native C509 toolkit that supports PQ algorithms and evaluates open-source and proprietary certificate parsers. While the IETF C509 draft proposal reports a size reduction of over 50%, our evaluation confirms approximately 40% savings for classical certificates generated according to our proposed minimal certificate profile. For PQ certificates, the savings plateau at around 200 bytes, rendering the size gains negligible. However, revocation lists consistently achieve a 60% reduction for 30,000 entries, independent of the cryptographic scheme (PQ or traditional). To quantify and compare the software implementation complexity of X.509 and C509, we conduct software complexity analysis using well-established heuristic metrics (e.g., cyclomatic complexity, Halstead metrics, logical lines of code). The findings further highlight the relative simplicity of the C509 parser implementation in software. Defining a standardised certificate profile for federated space would advance interoperability; however, adopting C509 requires carefully balancing modest PQ size savings against software simplification and the uncertainties associated with a draft standard.

Contents

Nomenclature

Abbreviations

Abbreviation	Definition
ANSSI	Agence nationale de la sécurité des systèmes d'information (French National Cybersecurity Agency)
ASN.1	Abstract Syntax Notation One
BER	Basic Encoding Rules
BSI	Federal Office for Information Security (Germany)
CA	Certification Authority
CBOR	Concise Binary Object Representation
CCSDS	Consultative Committee for Space Data Systems
CDDL	Concise Data Definition Language
CLM	Certificate Lifecycle Management
CNSA	Commercial National Security Algorithm (NSA "CNSA Suite 2.0")
COTS	Commercial Off-The-Shelf
CRL	Certificate Revocation List
CRQC	Cryptographically Relevant Quantum Computer
CSR	Certificate Signing Request
DER	Distinguished Encoding Rules
DIR	Certificate Repository (Directory)
DLP	Discrete Logarithm Problem
DPD	Delegated Path Discovery
DPV	Delegated Path Validation
ECC	Elliptic-Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
ECSS	European Cooperation for Space Standardisation
EE	End Entity
FBCA	Federal Bridge Certification Authority
FCPCAG2	Federal Common Policy CA Generation 2
FIPS	Federal Information Processing Standards
FN-DSA	Fast Fourier Transform over NTRU-Lattice-Based Digital Signature Algorithm
FPKIMA	Federal PKI Management Authority
FPKIPA	Federal PKI Policy Authority
HQC	Hamming Quasi-Cyclic
IANA	Internet Assigned Numbers Authority
IETF	Internet Engineering Task Force
IGCA	Intergovernmental Certification Authority
IoT	Internet of Things
KEM	Key Encapsulation Mechanism
LMS	Leighton–Micali Signatures
LWE	Learning With Errors
ML-DSA	Module-Lattice-Based Digital Signature Algorithm
ML-KEM	Module-Lattice-Based Key Encapsulation Mechanism
NASA	National Aeronautics and Space Administration
NIST	National Institute of Standards and Technology
NSA	National Security Agency (USA)
NSS	National Security Systems (USA)
OCSP	Online Certificate Status Protocol
OID	Object Identifier

Abbreviation	Definition
PIV-I	Personal Identity Verification–Interoperable
PKI	Public Key Infrastructure
PQ	Post-Quantum
PQC	Post-Quantum Cryptography
RA	Registration Authority
RFC	Request for Comments
SCVP	Server-Based Certificate Validation Protocol
SIS	Shortest Integer Solution
SLH-DSA	Stateless Hash-Based Digital Signature Algorithm
TLS	Transport Layer Security
TLV	Tag-Length-Value
VA	Validation Authority
XMSS	eXtended Merkle Signature Scheme

1

Introduction

Public Key Infrastructure (PKI) serves as the backbone of today's vast digital ecosystem, enabling secure and trusted communication across countless networks and devices. As humanity's ambitions extend beyond Earth, the importance of PKI evolves with it, addressing new challenges posed by the unique constraints of space environments. This chapter sets the stage for a line of research dedicated to the core component of any PKI: cryptographic certificates. This study aims to support the development and deployment of PKI in space, enabling secure communication for future federated missions.

Context

Space exploration increasingly relies on collaboration between agencies and organisations. The technical and financial demands of projects like a sustained lunar presence or Mars missions often exceed the capacity of a single agency. For example, the Artemis program is a joint international effort to develop infrastructure such as the Lunar Gateway, a modular station for long-term lunar orbit operations [1, 2]. These initiatives require integrating diverse systems and practices. Ensuring reliable interoperability across independently managed assets is thus a key challenge in modern space missions.

The need for technical interoperability in international space missions has led to the development of standardised guidelines. These are defined and maintained by space standardisation bodies, such as the European Cooperation for Space Standardisation (ECSS) and the Consultative Committee for Space Data Systems (CCSDS), which include protocols and recommendations across all layers of space communication. For key management, related standards remain limited to symmetric cryptography, which lacks the scalability needed in collaborative missions [3]. While Public Key Infrastructure (PKI) and its central component, the cryptographic certificate, provide a scalable framework for trust and key management in terrestrial systems, CCSDS notes that space deployment faces architectural and operational challenges [4]. As missions become more federated and involve more independently governed participants, this remains a significant barrier to large-scale secure communication.

A notable advancement in this area is the experimental Intergovernmental Certification Authority (IGCA) specification [5]. Proposed by CCSDS, IGCA is an intended federated space PKI standard designed to promote trusted cooperation and interoperability among entities. It defines policies and requirements for the technologies and protocols that allow the IGCA and its affiliates to issue and manage certificates for software, spacecraft, ground stations, satellites, personnel, and other mission applications.

Problem Statement

Despite promoting centralised management and policy enforcement, IGCA currently does not include a precise specification of the certificate profiles to be used. As CCSDS notes, PKI components from different vendors may be unable to communicate, and users may be unable to process each other's certificates [4]. In practice, most space PKI research and standardisation efforts assume the terrestrial X.509 Internet profile [6] as the default for interoperability. IGCA builds on the CCSDS Authentication Credentials [7], which in turn references the X.509 Internet Profile, yet without any specific adaptations.

Within federated space PKI, the CCSDS Authentication Credentials Standard [7] and its reliance on the X.509 Internet Profile (hereafter X.509) [6] can pose potential challenges.

First, X.509 was designed for general-purpose terrestrial use and lacks minimalism. It includes redundant data and uses verbose encoding [8], inflating certificate size beyond core cryptographic content and straining bandwidth-limited links. Its structural complexity also broadens the attack surface. Context-dependent parsing has been linked to security issues such as memory errors [9], impersonation attacks [10], and denial-of-service conditions [11]. These problems are amplified in embedded or space systems, where implementing secure parsers is even more difficult due to constrained resources. As a result, X.509 poses performance and implementation challenges in space applications.

Second, under IGCA, the current profile allows unrestricted use of extensions. In X.509, extensions are optional fields that encode additional attributes or constraints, often governing trust, key purposes, or policies. Allowing federation participants to define their own profiles without coordination risks interoperability failures. For example, unrecognised critical extensions may break validation. While X.509 (and implicitly, the CCSDS Authentication Credentials) enables 17 standard extensions [6], it does not prohibit custom or vendor-specific ones [12]. A survey of 200 million certificates found nearly 200 distinct extensions [9], showing high variability. Another analysis of 11 million certificates reported that 21.5% were syntactically incorrect, with 5.7–10.5% still accepted by major TLS libraries [10], highlighting inconsistencies among implementations claiming the same conformance [13]. The uncontrolled use of extensions—including deprecated ones [14]—can lead to inconsistent configurations in federated settings where certificates must be deterministic and uniformly interpreted to achieve interoperability.

Third, the migration to post-quantum (PQ) cryptography raises some issues for X.509, particularly regarding interoperability and certificate size. Although X.509 accommodates new algorithms through extensible mechanisms, practical adoption requires gradual system updates to recognise new algorithm identifiers, making the transition complex [15, 16, 17]. While the IGCA specification acknowledges the need to address PQ algorithms, the CCSDS Authentication Credentials offers no specific guidelines at this stage. Moreover, PQ schemes often produce much larger keys and signatures, sometimes tens of kilobytes, straining bandwidth and embedded memory. These constraints highlight the need to minimise certificate size and impose requirements for PQ certificate support in federated space systems.

Collectively, these challenges highlight the need for a well-defined certificate profile—one that supports PQ algorithms, uses minimal encoding and functionality, and defines a specific, standardised set of extensions suitable for federated space environments.

Objectives

To address these limitations, this work explores the potential applicability to space systems of the emerging C509 profile, currently under standardisation by the Internet Engineering Task Force (IETF) [8]. Designed for constrained environments, C509 defines a restricted subset of X.509 features to lower parsing complexity and certificate size—"in many cases … with over 50%" [8]. It uses a compact encoding [18] suitable even for constrained devices operating with about 10 KiB of RAM and 100 KiB of flash memory [19]. While promising, its suitability for space remains to be assessed.

Building on this foundation, this study adopts a certificate-centric methodology to identify:

"What is the minimal, interoperable certificate profile capable of bridging traditional and PQ cryptography while supporting cross-domain federation?", by:

(i) Analysing PQ certificate formats and identifying challenges and mitigation strategies in designing an interoperable profile for federated missions.

(ii) Reviewing extension configurations used in terrestrial federated systems to inform the design of minimal and deterministic certificate profiles for federated missions.

(iii) Conducting a comparative evaluation of X.509 and C509 based on certificate size and implementation complexity to assess their relative suitability for deployment in space.

While certificate lifecycle protocols such as revocation and validation are essential to developing a certificate profile, this work focuses exclusively on certificate structure and encoding, complementing future research on lifecycle management.

Contributions

This research offers novel contributions to certificate design and implementation in the federated space:

- **Minimal federated profile:** A proposed reduced certificate profile tailored to the operational and interoperability needs of federated space missions. This includes a novel PQ format to mitigate interoperability challenges arising from diverging security guidelines on PQ transition, as well as a minimal set of profiles with fixed extension sets to enable federation-scale interoperability.
- **Open-source C509 implementation:** The first public natively signed C509 implementation with PQ algorithm support for certificates, signing requests, and revocation lists. The tool implements interfaces similar to OpenSSL for intuitive use and contains fewer than 3,000 logical lines of code.
- **X.509 vs. C509 empirical evaluation:** A quantitative comparison of C509 and X.509 based on object size and software complexity, highlighting trade-offs when adopting C509 in a PQ setting. Results show that C509 offers negligible size gains for PQ certificates but achieves significant reductions on revocation lists, independent of the cryptography used. C509 parser implementations are significantly simpler to implement in software compared to X.509. Despite its benefits, C509 has yet to be standardised, making its immediate adoption a matter of debate.
- *** Forward-looking validation strategy:** Identification of the Server-Based Certificate Validation Protocol (SCVP) [20] as a potential mechanism to offload complex validation from constrained assets and enable scalable federated trust and policy management. Although not implemented here, SCVP is highlighted as a promising future research direction.

These contributions aim to advance an interoperable, certificate-based trust infrastructure for federated space environments—particularly within the IGCA framework—by laying the groundwork for a dedicated, minimal, and well-defined certificate profile.

Document Structure

The topics analysed in this work are distinct and warrant separate examination. Rather than following a sequential build-up, the chapters offer complementary insights that together inform the development of federated space certificate profiles. This work is structured as follows:

- **Chapter 2**: Provides general background for all subsequent analyses.
- **Chapter 3**: Examines PQ algorithms and PQ certificate formats.
- **Chapter 4**: Reviews X.509 extensions and terrestrial federated profiles, highlighting challenges for constrained space assets.
- **Chapter 5**: Describes the C509 tool's design and implementation, providing the foundation for the analysis in Chapter 6.
- **Chapter 6**: Conducts an empirical comparison between X.509 and C509.
- **Chapter 7**: Synthesises findings in the context of federated space PKI and outlines limitations.
- **Chapter 8**: Summarises related work.
- **Chapter 9**: Concludes the research.

2

Background

Secure communication in modern systems relies on well-established cryptographic foundations, spanning PKIs, digital certificates, and emerging PQ algorithms. This chapter provides the necessary background for understanding the core principles of PKI, along with the structure, encoding, and implementation of cryptographic certificates. It also introduces a short summary of the current state of the PQ migration and outlines software complexity metrics that will be used in Chapter 6.

2.1. Public Key Infrastructure

A digital, public key certificate[1] is a signed data structure that binds a public key to an entity's identity, allowing others to verify the key's association with that entity. A Public Key Infrastructure (PKI) comprises the roles, policies, hardware, software, and procedures needed to create, manage, distribute, use, store, and revoke certificates. Its purpose is to establish trust and enable secure communication between parties without a prior relationship.

A typical, one-layer PKI architecture is illustrated in Figure 2.1, together with its core components:

- **Certification Authority (CA):** Issues and manages digitally signed certificates, serving as the root of the trust model in PKI systems.
- **Registration Authority (RA):** Offloads identity verification from the CA. Its sole responsibility is to validate certificate requests before forwarding them to the CA for issuance.
- **Certificate Repository (DIR):** An electronic repository that stores certificates and associated metadata. The CA publishes issued certificates here, enabling other entities to retrieve them.
- **Validation Authority (VA):** Assists users in verifying certificate status. It may provide revocation information or even perform full path validation on behalf of constrained end-entities.
- **End Entities (EE):** The consumers of certificates—typically devices, applications, or systems that rely on certificates to authenticate peers and establish secure communication.

Additionally, PKIs follow a Certificate Lifecycle Management (CLM) process [21], depicted in Figure 2.1:

- **Certificate Request and Enrollment:** An EE generates a cryptographic key pair and submits a Certificate Signing Request (CSR) [22] to a trusted authority, containing its public key and identity data, as shown in **(1)**. The CSR is self-signed to provide proof of possession.
- **Identity Verification:** The RA validates the identity and legitimacy of the CSR. Upon successful validation, the request is forwarded to the CA for certificate issuance, shown in **(2)**.
- **Certificate Issuance:** The CA generates a digital certificate binding the verified identity to the public key. It signs the certificate and includes metadata such as validity, intended usage, and algorithms used. The certificate is returned to the requester and optionally published in the directory, as shown in **(3.1)** and **(3.2)**.

[1] This work excludes attribute/authorisation certificates that bind attributes or roles to identities.

- **Revocation:** If a certificate must be invalidated before expiry (e.g., due to compromise), the CA revokes it and updates the certificate status in the directory and to validation authorities, as depicted in **(3.2)** and **(4)**.
- **Certificate Validation:** When two EEs initiate secure communication **(7)**, the receiver must validate the sender's certificate. This is typically done by querying the VA, as illustrated by **(5)** and **(6)**. Depending on system capabilities, validation may be assisted or fully delegated to the VA.
- **Renewal:** As a certificate nears expiration, it can be renewed by submitting a new CSR. This may reuse the original key pair or involve generating a new one. Renewal typically retriggers the identity verification and issuance process, forming a **(1)**–**(2)**–**(3)** cycle.

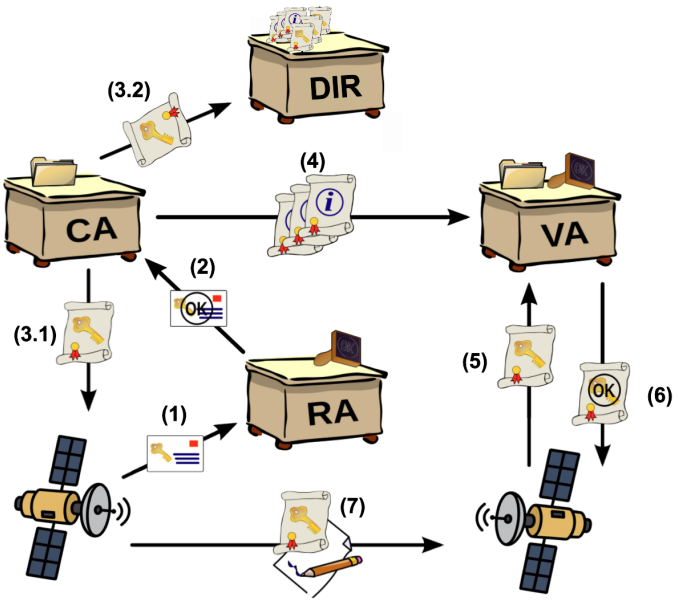


Figure 2.1: Illustration of certificate lifecycle roles in a Public Key Infrastructure (PKI). Adapted from [23] under CC BY-SA 3.0.

While the previous example illustrates a single-tier PKI architecture with one CA, real-world deployments typically employ a more complex, hierarchical structure. Relying on a single trusted CA is often impractical, particularly in large-scale, globally distributed systems such as the Internet. Instead, trust is delegated through a chain of subordinate CAs, forming a multi-layered hierarchy that reflects organisational or operational boundaries. This structure enables scalable trust management and compartmentalised control, as illustrated in Figure 2.2.

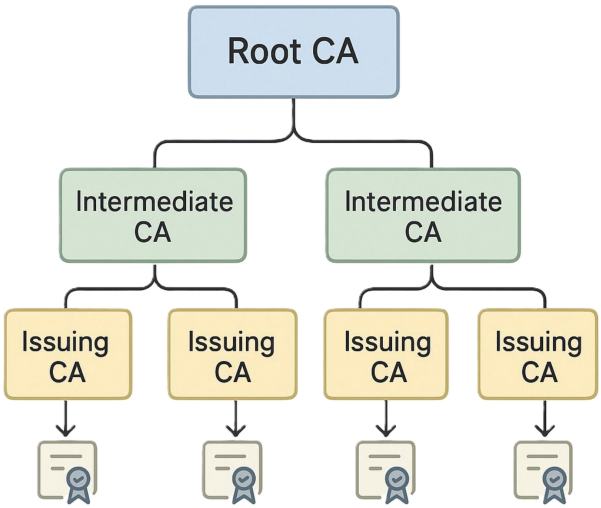


Figure 2.2: Exemplary CA hierarchy as is typically found on the Internet.

At the top of the hierarchy is the **Root CA**, which is inherently trusted by all relying parties. Its self-signed certificate is typically pre-installed on systems that participate in the PKI. To reduce exposure and enhance security, the Root CA is generally kept offline and delegates certificate issuance responsibilities to one or more **Intermediate CAs**. These, in turn, may authorise **Issuing CAs** to issue certificates to end-entities. This separation of roles supports scalability, enables policy segmentation, and limits the impact of potential key compromise. In Figure 2.2, the registration and validation authorities are abstracted; depending on the implementation, these roles may either be handled internally by a CA or delegated to independent, cooperating entities.

However, independently managed PKIs often lack a shared root, making vertical trust hierarchies insufficient. In such cases, trust must be established *horizontally*, through **federated PKI architectures** that enable interoperability between autonomous trust domains. These architectures facilitate cross-domain trust relationships using mechanisms such as **cross-certification**, **bridge Certification Authorities (CAs)**, or **extended trust lists** [24]. Each approach entails specific trade-offs:

- **Bridge CAs** act as a single intermediary that each participating PKI cross-certifies with, simplifying trust management by reducing the number of required relationships. This model supports scalability and administrative efficiency, as illustrated in Figure 2.3.
- **Cross-certification** in a mesh topology allows any two CAs to establish mutual trust directly. While this model maximises decentralisation, it does not scale well, as the number of trust relationships grows quadratically with the number of participating CAs.
- **Extended trust lists** decentralise trust decisions by relying on locally maintained lists of trusted CAs. This method provides flexibility and autonomy but increases administrative overhead, as each organisation must curate and distribute its trust list.

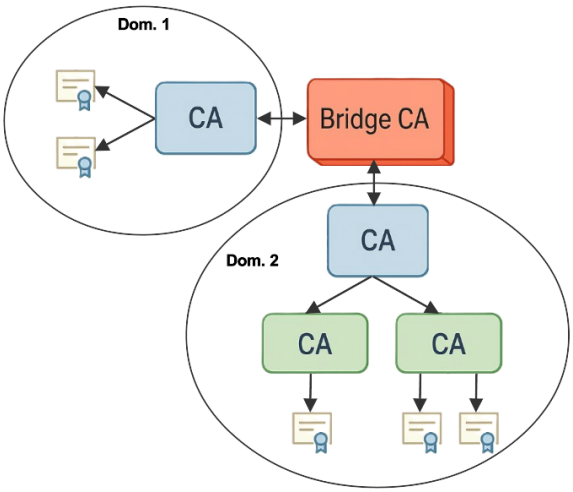


Figure 2.3: Bridge CA architecture. Root CAs of each domain generate cross-certificates with the Bridge CA, marked by the bidirectional arrows, enabling certificate path construction between different domains.

The **Intergovernmental Certification Authority (IGCA)** [5] defines a federated PKI framework tailored for space missions, structured around a bridge-CA model to facilitate cross-domain trust among participating agencies. Rather than relying on a single root authority, IGCA enables each organisation to either cross-certify its own root CA or rely on IGCA's issuing CA, fostering interoperability while preserving autonomy. The specification outlines technical, operational, and governance requirements to ensure secure and consistent certificate issuance across diverse mission environments. IGCA exemplifies both the challenges of aligning certificate policies, managing trust anchors, and validating certificates across organisational boundaries, and the potential of federated PKI to support secure, scalable, and interoperable communication in complex, multi-agency space systems.

2.2. Cryptographic Certificates

X.509 is the prevailing standard for public-key certificates, defined by the International Telecommunication Union Telecommunication Standardisation Sector (ITU-T) [12]. Its structure is specified using **Abstract Syntax Notation One (ASN.1)**, a formal language designed to describe structured data in a platform-independent manner. ASN.1 can express nested sequences, choice types, optional fields, and user-defined tags, and is itself standardised by the ITU-T [25]. It is widely used in telecommunications and cryptographic protocols, including 5G and Kerberos.

While ITU-T X.509 defines the generic certificate syntax, the Internet Engineering Task Force (IETF) specifies the **X.509 Internet profile** [6]. This profile constrains the X.509 feature set and defines strict rules for fields and validation procedures to promote consistent behaviour across implementations. In practice, the term ”X.509” often refers to this IETF profile, which also serves as the foundation for various application-specific certificate profiles [26, 27, 28, 29].

A typical X.509 certificate contains the fields illustrated in Figure 2.4, with the following semantics:

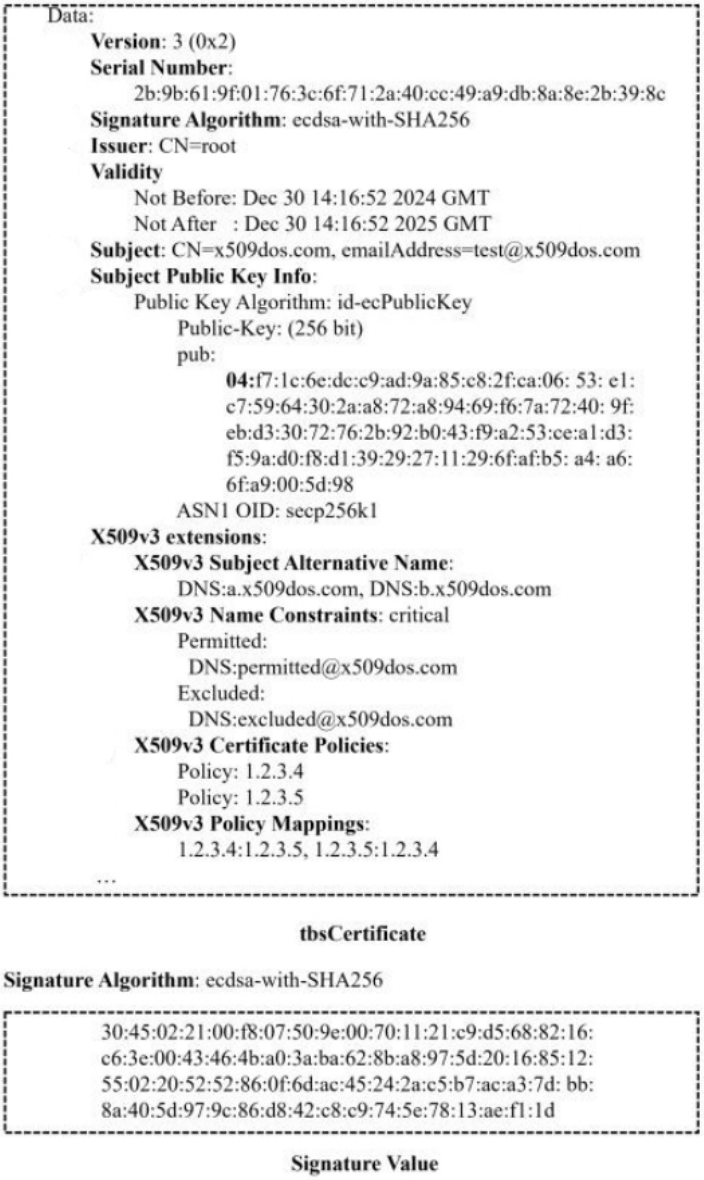


Figure 2.4: Exemplary X.509 certificate. Adapted from [11].

- **Version** – Indicates the format or generation of the certificate structure.
- **Serial Number** – A unique identifier that the issuing authority assigns to distinguish the certificate.
- **Signature Algorithm** – Specifies the algorithm used to sign the certificate, protecting its integrity.
- **Issuer** – The distinguished name of the authority that issued the certificate.
- **Validity** – A time interval, defined by a ”not before” and ”not after” timestamp, during which the certificate is considered valid.
- **Subject** – The distinguished name of the certified entity (e.g., a device, person, or server).
- **Subject Public Key** – The public key being certified, corresponding to a private key held securely by the subject.
- **Extensions (optional)** – Additional metadata or constraints, such as key usage, policies, certificate authority identifiers, alternative subject names, or constraints on certificate chaining. These extensions can be marked as *critical* or *non-critical*. This flag decides whether the relying party must process the extension or not.
- **Signature Value** – The signature computed over the certificate data by the issuing CA using its private key, allowing any party with access to the CA's public key to verify the certificate's authenticity and detect tampering.

ASN.1 protocols have the flexibility to select encoding rules that best suit their needs and priorities, such as compactness, simplicity of processing, or determinism. For X.509 certificates, the **Distinguished Encoding Rules (DER)** were chosen. DER primarily aims to provide a canonical, and therefore deterministic, binary encoding. This property ensures that every ASN.1 structure has a unique encoded representation. The canonical form is critical for digital signatures, as multiple valid representations would lead to ambiguity and potentially compromise security.

DER use a **Tag-Length-Value (TLV)** format, as in Figure 2.5. This encoding is designed to ensure that values are small enough to fit into the available memory and to allow for rapid skipping over nested values [25]. While predictable, DER encoding can also be verbose. The goal of DER is unique encoding, not message compactness. While this encoding aligns with the requirements of cryptographic certificates, it may not be ideal for constrained devices due to message size and implementation complexity.

Figure 2.5: Nested Tag-Length-Value (TLV) structure

Developed by IETF, **C509** is a compact certificate profile for constrained environments where traditional X.509 certificates are too large or complex [8]. The profile preserves a subset of the X.509 information model [18], reducing implementation and transmission overhead. The draft defines related structures, such as signing requests, private keys, and certificate chains, and includes a reference implementation[2]. A separate draft for revocation protocols is expected after the standardisation of C509.

For encoding, C509 uses the **Concise Binary Object Representation (CBOR)** [18]. CBOR is a binary serialisation format notable for its efficiency and simplicity. It is self-descriptive and can be deterministic, similar to a binary form of JSON. It also claims to offer significantly reduced encoding size and complexity compared to DER, making it more suitable for limited computational and bandwidth resources. The C509 data schema is defined using the **Concise Data Definition Language (CDDL)** [30]. A **schema** is a formal description of the structure, types, and constraints of data that defines how information is organised and validated. To some extent, this definition language, CDDL, and its encoding, CBOR, could be seen as a simpler alternative to ASN.1 and DER, respectively.

In contrast with the TLV format of DER, which prefixes every field with a separate tag and an explicit multi-byte length, CBOR compresses this metadata into a single initial byte. The high-order three bits of this byte specify the **major type** (with only eight in total), while the low-order five bits encode **additional information**. For small scalars, this information directly represents values, such as integers 0–23 or

[2] `https://github.com/cose-wg/CBOR-certificates`

the constants `false`, `true`, and `null`, or the length of arrays and maps containing up to 23 elements. For larger values or lengths, the five bits act as a size indicator: 24 denotes that the next byte contains the value or length, 25 the next two bytes, 26 the next four, and 27 the next eight, all in big-endian form. The value 31 indicates an indefinite length, allowing the streaming of strings, arrays, or maps until a dedicated break byte is encountered. Nevertheless, deterministic CBOR relies on definite length encoding. CBOR's compact encoding reduces much of the structural overhead of DER's TLV format.

Besides encoding, other notable size-reduction techniques employed by C509 include:

- mapping frequently used OIDs to small integers via IANA-assigned registries;
- applying context-specific optimisations, such as omitting the issuer in self-signed certificates; and
- eliminating redundant nested structures.

C509 can reduce the size of pre-quantum IoT-profiled certificates [28] by over 50% and claims reduced parsing code size and memory usage.

C509 defines two signature models:

1. **Re-encoded certificates** retain the original DER signature. The CBOR form is fully invertible—verifiers must reconstruct the DER-encoded octet string and validate it using standard X.509 stacks. This yields bandwidth savings while maintaining compatibility with legacy PKI.
2. **Natively signed certificates** calculate the signature over the CBOR sequence itself, eliminating DER reconstruction at validation time. This maximises code simplicity but constrains compatibility to C509-aware verifiers only.

The difference in the signature structure between C509 and X.509 for a semantically equivalent certificate is illustrated in Figure 2.6.

(a) Re-encoded C509 certificate

(b) Natively signed C509 certificate

(c) X.509 certificate

Figure 2.6: Signature value differences between the C509 and X.509 profiles.

2.3. Certificate Validation

The first step in certificate validation is constructing the **chain of trust** [6]. This is a list of certificates, linked through digital signatures, that forms a path from the EE certificate up to a trusted root CA. When two parties communicate, they may either send only their certificate or include intermediate certificates as well. If the certificate chain is complete (i.e., the end-entity already possesses or is provided with all required intermediates), construction is straightforward. However, if one or more intermediate certificates are missing, the relying party must perform **certificate discovery**. This can involve retrieving certificates from external directories using information from extensions like `Authority Information Access (AIA)`.

The second step, path validation, involves multiple checks performed on each certificate in the chain.

First, the **digital signatures must be verified**. This ensures that each certificate was issued by the previous one in the chain and has not been tampered with. A direct implication is that the longer the chain, the more cryptographic operations the validator must perform. Importantly, the cryptographic primitives used across the chain may differ, requiring the validator to support and correctly implement all relevant verification operations. For resource-constrained devices, such as spacecraft or planetary rovers, this can represent a significant computational burden, particularly in the context of post-quantum cryptography. Nevertheless, a deeper chain with multiple intermediate CAs can provide benefits such as trust delegation, operational flexibility, and risk separation. To some extent, the number of layers in the PKI architecture represents a trade-off between security, trust agility, and computational resources.

Second, all certificates in the chain must be within their `validity` periods. Each certificate defines a time interval, from `notBefore` to `notAfter`, during which its associated key is authorised for use. This interval corresponds to the **cryptoperiod** of the key, which is defined by security policy and operational considerations to limit the time window in which a key can be compromised [21]. The length of this period varies depending on the role. For example, root CA certificates often have significantly longer validity periods than EE certificates, as the root's expiration invalidates the entire chain beneath it. Crucially, validity checking assumes access to a **reliable time source**, which in space cannot always be guaranteed due to limited ground contact and challenges of maintaining onboard time synchronisation.

Third, the validator must ensure that the chain complies with all applicable **constraints and policies**, as specified in certificate extensions. These include:

- `basicConstraints`: marks if a CA owns the certificate; optionally imposes a maximum path length
- `nameConstraints`: defines permitted or excluded name patterns
- `policyConstraints`: restricts the use and propagation of policies in the certification path

Violations of any of these constraints must result in a validation failure.

Finally, the most challenging check to implement is the **revocation check**. This ensures that none of the certificates in the chain have been explicitly invalidated before their expiration. A certificate may be revoked if it was issued in error, its associated private key was compromised, or the subject's authorisation was withdrawn. In such cases, the certificate must be considered untrusted, even if it remains within its declared validity period. The issuing CA maintains revocation status, which can be distributed to relying parties via VA or directly embedded in revocation check protocols.

Two commonly used methods for revocation checking are the **Certificate Revocation List (CRL)** [6] and the **Online Certificate Status Protocol (OCSP)** [31]. A CRL is a signed list, periodically issued by a CA, containing the serial numbers of revoked certificates. Relying parties typically download the CRL from a URL specified in the `CRL Distribution Points` extension and verify its signature. To reduce CRL size, `delta CRLs` may be used to distribute only incremental changes since the last complete CRL. Each CRL includes its validity period (typically 24 hours) and must be refreshed accordingly to maintain up-to-date revocation status. However, in space environments, ground mission control may need to manually distribute CRLs to spacecraft, and both the size of CRLs and the frequency of available communication windows (defined by the contact plan and antenna access) become limiting factors.

OCSP, in contrast, provides a real-time mechanism for checking the status of a specific certificate. A client sends a request to an OCSP responder, typically specified in the `Authority Information`

`Access` extension, and receives a digitally signed response indicating whether the certificate is good, revoked, or unknown. OCSP offers reduced bandwidth and faster lookups compared to downloading full CRLs, but introduces challenges such as availability, latency, and user privacy leakage, since the responder learns which certificate is being checked. To mitigate performance, **OCSP stapling** is often employed: the server pre-fetches an OCSP response from the responder and includes ("staples") it with its certificate, so the client does not need to query the OCSP responder directly. In space, additional challenges arise: ground stations only have contact with spacecraft at predefined intervals, dictated by orbit, antenna access, and mission schedules, which limits the feasibility of real-time status queries.

While CRLs and OCSP provide mechanisms for revocation checking, they still require the client to perform certificate validation locally. More advanced solutions are needed if clients are resource-constrained and are expected to perform path discovery and complex policy enforcement. The **Server-Based Certificate Validation Protocol (SCVP)** [20] addresses this by allowing a client to delegate path construction (and validation) to a trusted validation server. In other words, an SCVP server should be able to perform all checks mentioned above. SCVP could be employed to offload certificate validation from constrained devices, providing centralised, policy-aware, and updated validation services.

2.4. Post-Quantum Cryptography

The rise of quantum computing introduces critical security concerns for traditional asymmetric cryptography. Quantum algorithms—most notably **Shor's algorithm**—can solve, in polynomial time, the underlying hardness problems of many widely used schemes based on **integer factorisation problem** and the **discrete logarithm problem (DLP)**. This threatens the security of cryptosystems such as **RSA**, which relies on factorisation, as well as schemes based on the DLP over various groups, including but not limited to **elliptic-curve cryptography (ECC)**. As a result, any asymmetric primitive relying on these problems becomes insecure against an adversary equipped with a **cryptographically relevant quantum computer (CRQC)** capable of solving them in polynomial time using Shor's algorithm.

In response, the U.S. National Institute of Standards and Technology (NIST) leads the standardisation of **post-quantum cryptography (PQC)** [32, 33]. Currently, NIST has standardised three quantum-resistant algorithms under the Federal Information Processing Standards (FIPS) framework [34, 35, 36, 37]. Namely, they are Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM, formerly CRYSTALS-Kyber), Module-Lattice-Based Digital Signature (ML-DSA, formerly CRYSTALS-Dilithium), and Stateless Hash-Based Digital Signature Standard (SLH-DSA, formerly SPHINCS+). Also, Fast-Fourier transform (FFT) over NTRU-Lattice-Based Digital Signature Algorithm (FN-DSA, based on Falcon) is yet to be NIST-standardised [38]. Lastly, the Hamming Quasi-Cyclic (HQC) algorithm was selected as a backup key encapsulation mechanism (KEM) [39, 40].

These algorithms span multiple **cryptographic families**, each defined by distinct mathematical structures and underlying hardness assumptions [33]. **Lattice-based schemes** (e.g., ML-KEM, ML-DSA) rely on the computational difficulty of lattice problems, such as **Learning with Errors (LWE)** and the **Shortest Integer Solution (SIS)**. **Hash-based schemes** (e.g., SLH-DSA) depend entirely on **collision and preimage resistance** of cryptographic hash functions, providing security based on minimal assumptions. **Code-based schemes** (e.g., HQC) build their security on the hardness of decoding random linear error-correcting codes. Although currently less represented in standardisation efforts, **multivariate** and **isogeny-based** cryptographic families offer alternative security assumptions that can diversify the set of available cryptographic primitives. Nevertheless, NIST's intentional selection, which spans different families, offers fallback options during migration if a family is later deemed insecure.

Hash-based signature schemes can be categorised into two classes: stateful and stateless, both relying solely on the security of cryptographic hash functions but differing significantly in their operational models. The eXtended Merkle Signature Scheme (XMSS) and Leighton-Micali Signatures (LMS) are examples of stateful schemes standardised by NIST and the IETF [41, 42, 43]. These use a Merkle tree of one-time signature keys, where each private key element must be used exactly once, requiring secure state management across signings to prevent reuse and ensure security. XMSS and LMS have varying signature sizes, depending on the height of the tree and selected parameters, which are typically smaller than those of SLH-DSA. However, this comes at the expense of state tracking, which can pose a challenging implementation aspect.

A key challenge in deploying PQC is its demanding computational and bandwidth requirements. Many PQC algorithms require large public keys and signatures, some exceeding kilobytes in size. They also involve mathematically intensive operations. For instance, lattice-based schemes depend on **high-dimensional polynomial arithmetic** [38], while code-based schemes such as HQC involve **costly syndrome decoding** [39]. These operations can strain constrained environments, where processing power, memory, and bandwidth are limited.

While national authorities broadly agree that PQC should be introduced with caution, their recommendations already diverge, particularly regarding hybridisation strategies. **Hybrid cryptographic schemes**, which combine traditional and PQ algorithms [44], have emerged as a pragmatic migration path [32], offering resilience if a PQ primitive is later found vulnerable.

This caution stems from the relatively recent nature of the underlying hardness assumptions, especially those of structured lattice-based schemes like ML-KEM and ML-DSA, which may be susceptible to future **structural cryptanalysis**. In response, the **German Federal Office for Information Security (BSI)**, the **French National Cybersecurity Agency (ANSSI)** and the **European Commission** strongly advocate for hybrid deployments of such algorithms. Their guidance recommends pairing lattice-based primitives with traditional counterparts, ensuring a fallback mechanism in case of cryptanalytic advances [45, 46, 47]. In contrast, hash-based signature schemes are grounded in long-established assumptions and are thus regarded as sufficiently robust for standalone use, making hybridisation optional. As for HQC, its hybridisation status remains undefined, with no concrete guidance issued to date. Overall, hybridisation is seen as a conservative strategy that preserves traditional security guarantees while PQ schemes continue to mature.

Conversely, the **U.S. National Security Agency (NSA)**, acting as the national cryptographic authority for **National Security Systems (NSS)**, takes a more assertive stance in favour of standalone PQ deployments. NSA expresses complete confidence in the security of its selected algorithms and explicitly states that hybrid implementations are not required for security purposes [48]. Rather than being a core security measure, hybrid modes are presented as transitional tools—helpful in maintaining interoperability during migration, but not necessary for ensuring cryptographic robustness.

As a result, implementers face a fragmented landscape: hybrid use is **promoted by European recommendations** for lattice-based algorithms and optional for hash-based signatures. At the same time, it is explicitly **not required by the U.S. NSA**. This **regulatory divergence** complicates international interoperability and underscores the need for flexible certificate profiles that support both hybrid and pure PQ variants, depending on the deployment context.

2.5. Software Complexity Metrics

Software complexity refers to the effort required to understand, modify, and maintain a system as its codebase and internal interactions grow. To manage the risks associated with this complexity—such as unintended side effects and bugs—these dimensions are often approximated using **heuristics**: practical, experience-based estimates that support design and evaluation. These heuristics are embedded in **static-analysis tools**, where they help quantify **software complexity** and guide design trade-offs.

This study adopts a set of such heuristics in the form of widely used static-analysis metrics, summarised in Table 2.1. They are supported by commercial tooling, such as SonarQube [49] and SciTools Understand [50], representing de facto metrics in industrial quality assurance workflows. Some have also been formally incorporated into software engineering standards. In particular, LLOC and CCN are recommended for reporting space systems implementations of all criticality levels [51]. At the international level, IEEE formally define **Halstead's token-based metrics** as standardised complexity and reliability indicators [52, 53]. These metrics are used as a proxy for the **empirical comparison** between X.509 and C509 implementations in later chapters.

Table 2.1: Summary of static analysis metrics used in the study.

Metric	Purpose	Interpretation
Logical Lines of Code (LLOC)	Measures codebase size	Higher values imply greater storage, maintenance, and testing effort.
Cyclomatic Complexity (CCN)	Quantifies control flow complexity	Indicates the number of independent execution paths. Higher values indicate more challenging testing and a higher potential for bugs.
Halstead Volume	Captures token-level complexity	Approximates cognitive load by analysing the number and variety of operations and operands. Larger volumes imply more intricate logic.
Halstead Difficulty	Assesses comprehension difficulty	Estimates the effort required to understand or modify code. Sensitive to the ratio of unique tokens to total tokens.
Function Count	Counts declared functions	While not directly tied to complexity, but rather modularity, it is useful contextually.

3

Post-Quantum Certificates

As quantum computing threatens the foundations of traditional cryptography, space systems must begin migrating to post-quantum algorithms and address the operational and interoperability challenges that accompany this transition. As traditional cryptography is approaching deprecation by 2030 [54], initiating migration planning becomes critical. This chapter evaluates the post-quantum certificate formats that support these algorithms, considering aspects such as bandwidth impact, compatibility, security, and interoperability, drawing initial recommendations for a federated space certificate profile.

3.1. Structure Definition

NIST conducts interoperability tests on certificate formats intended for use during the PQ transition [32]. Among these are hybrid certificates, which combine traditional and PQ algorithms to enable a gradual migration or enhance cryptographic resilience [44].

3.1.1. Pure Post-Quantum Certificates

Certificates contain only PQ public keys and signatures. They are trivially compatible with the X.509 structure, requiring no adaptations aside from defining new PQ algorithm identifiers [15, 16, 17], as illustrated in Figure 3.1. The encoding of the key and signature is algorithm-specific and defined in the corresponding NIST or IETF specification [35, 36, 37], which specifies any required parameters and the raw byte layout to be embedded in the standard `BIT STRING` fields.

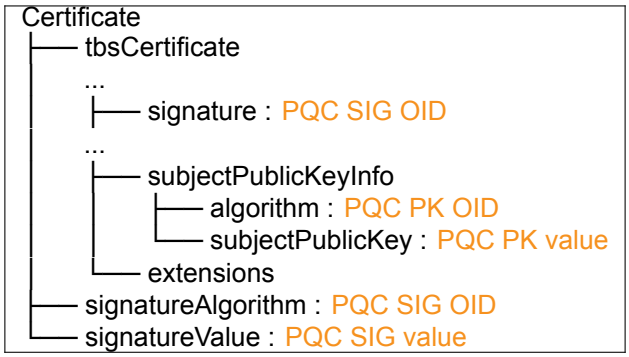


Figure 3.1: X.509 Pure Post-Quantum Certificate Format.

3.1.2. Hybrid Composite Certificates

Composite schemes combine traditional and PQ algorithms of the same type into a single public key and signature scheme [44]. Certificates using this approach embed a composite key and signature, each formed by concatenating the byte encodings of the individual components, displayed in Figure 3.2. Earlier proposals required encoding each component's OID via concatenation, leading to redundancy

and increased implementation complexity [55]. This general approach was later replaced by assigning a single OID to each supported algorithm pair, with current drafts restricting combinations to two components only [56, 57]. While the initial flexibility posed parsing challenges, the current structure remains fully compatible with the X.509 standard: only new OIDs are introduced, and the concatenation of values is encapsulated within standard `BIT STRING` fields. The format is designed explicitly based on security purposes to enforce the dual use of the component algorithms.

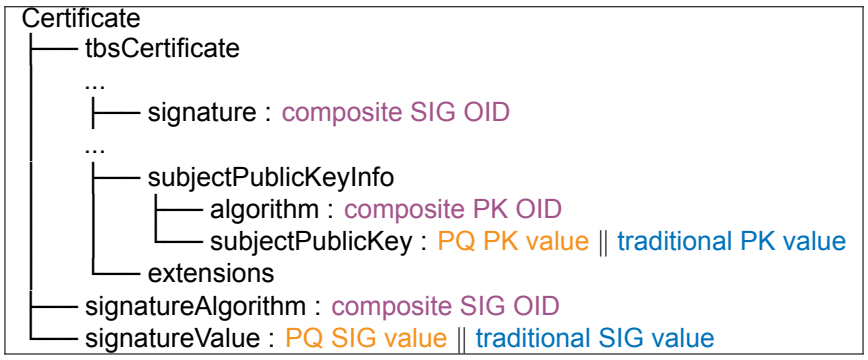


Figure 3.2: X.509 Hybrid Composite Certificate Format.

3.1.3. Hybrid with Extensions (former "Catalyst")

Certificates include extensions for alternative public keys and signatures, where the primary fields contain the traditional cryptographic components and the extensions encapsulate the PQ counterparts, as illustrated in Figure 3.3. Although the original proposal was controversial [58] and ultimately did not progress to RFC status [59], a revised version of the concept was later adopted into the ITU-T X.509 standard [12]. When recognised, the alternative extensions take precedence over the primary fields. This format was introduced as a transitional mechanism to simplify certificate management by enabling the representation of two cryptographic keys within a single certificate.

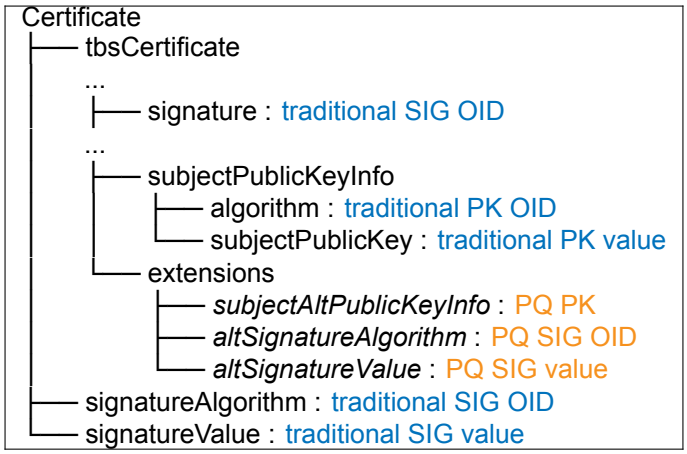


Figure 3.3: X.509 Hybrid with Extensions Certificate Format.

3.1.4. Hybrid Chameleon

Separate certificates are used to represent the traditional and PQ components of a hybrid scheme, each part of parallel chains [44]. To establish a binding between them, the traditional certificate—referred to as the Base—includes an extension that encodes the differences from the PQ certificate—referred to as the Delta [60], as illustrated in Figure 3.4. This extension allows the Delta certificate to be reconstructed on either the sender's or the relying party's side, depending on the application. Chameleon certificates were designed to support the transition, attempting to simplify certificate management in hybrid deployments. However, they are no longer endorsed by the IETF.

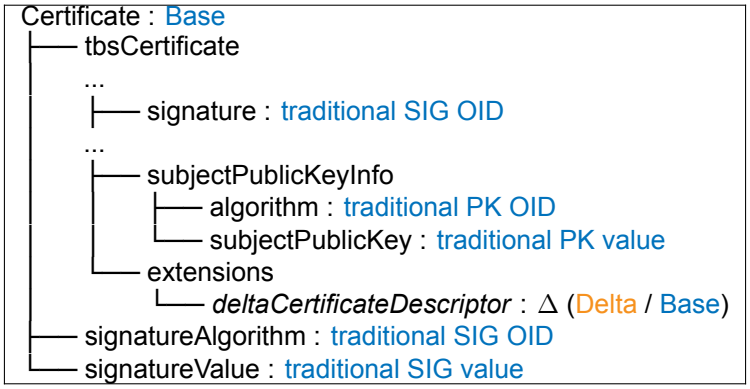


Figure 3.4: X.509 Hybrid Chameleon Certificate Format. Delta is identical to a PQ certificate from Figure 3.1.

3.1.5. Hybrid Bound

Separate certificates are used to represent the traditional and PQ components of a hybrid scheme, each part of parallel chains [44]. To establish a binding between them, an extension containing the hash of the traditional certificate is placed in the PQ certificate, as depicted in Figure 3.5. The IETF standardised extension [61] provides no security guarantees, but reassurance that the paired certificates belong to the same entity in multiple authentication protocols and during the transition.

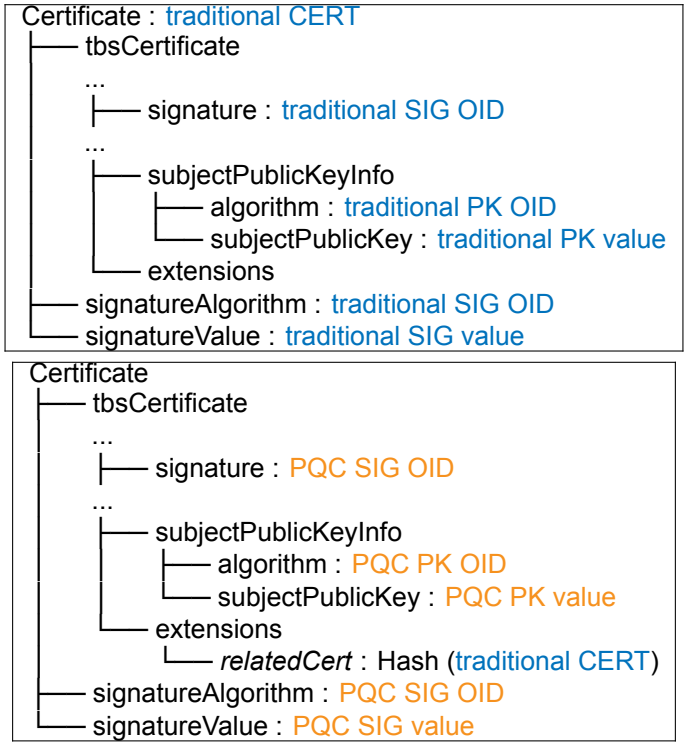


Figure 3.5: X.509 Hybrid Bound Certificate Format.

3.2. Certificate Size

To evaluate the relative transmission requirements of different certificate formats and demonstrate their structural overhead, a set of representative certificates was generated. Appendix A details the experimental profile and lists the certificates used in the assessment. The size trends are summarised in Table 3.1a, and further validated by comparison with certificates produced by Bouncy Castle during the IETF interoperability tests, as shown in Table 3.1b. The absolute values differ between the two tables due to the difference in the reference profile and naming. While Bouncy Castle's certificates represent

root CA certificates that contain additional extensions, such as `keyUsage` and `basicConstraints`, the generated certificates from Appendix A contain no extensions other than those mandated by the format. Nevertheless, it is the trend's consistency that validates the results. In this analysis, it is assumed that both the traditional and PQ components must be transmitted to the relying party. Accordingly, the size of bound certificates is computed as the sum of the sizes of their components. The pure certificate serves as the baseline for overhead, as it remains unmodified from the standard X.509 structure.

For the certificates from Appendix A, while the composite format shows a nominal 6-byte increase compared to pure certificates, this overhead arises from the differences in OIDs. Composite certificates introduce no structural overhead, and previous claims that composites have the most significant increase in certificate size [62] are no longer representative of the current specifications [56, 57].

In contrast, hybrid formats utilising extensions incur greater overhead due to the addition of three extra OIDs, each contributing to structural complexity. The overhead of Chameleon certificates varies with the number of differences between the paired certificates. Since the descriptor encodes two distinct certificates, it must explicitly encode differences such as distinct serial numbers, thereby increasing the structural overhead. Additionally, Chameleon certificates introduce an extra OID for the extension descriptor. However, this format mitigates duplication by sharing common fields (e.g., subject or issuer), which is beneficial compared to Hybrid Bound certificates when both components are transmitted.

Table 3.1: Comparison of format sizes (bytes) for pure `ML-DSA:44` and hybrid variants with `ECDSA:secp256r1`. Body size (bytes) is the non-cryptographic content (total size minus key and signature). Relative Increase shows the size increase (bytes) over the pure PQ certificate as a means of approximating the overhead. Footnotes indicate the tools used; for Hybrid Bound, sizes are summed across component certificates.

(a) Certificates generated according to Appendix A.

Format	Cert. Size (bytes)	Body Size (bytes)	Relative Increase (bytes)
Pure[1]	3894	152	-
Hybrid Composite[1]	4045	158	6
Hybrid with Extensions[2]	4112	229	77
Hybrid Chameleon[3]	4193	310	158
Hybrid Bound (Approx.)[1]	4247	363	211

(b) Certificates generated by Bouncy Castle[4].

Format	Cert. Size (bytes)	Body Size (bytes)	Relative Increase (bytes)
Pure	3958	216	-
Hybrid Composite	4157	254	38
Hybrid with Extensions	4219	334	118
Hybrid Chameleon	4332	448	232
Hybrid Bound (Approx.)	4365	480	264

Nevertheless, the relative increase introduced by each hybrid format remains marginal compared to the overall size of the certificate. Consequently, the associated overhead has no meaningful impact on the communication link.

3.3. Backwards Compatibility

Backwards compatibility enables a gradual transition from traditional to quantum-secure certificates, ensuring operational continuity for legacy systems that have not yet been updated to support PQC. Since both pure and composite certificates introduce new OIDs that are unrecognised by legacy systems, they are not backwards compatible without software/firmware updates.

[1] `https://github.com/open-quantum-safe/liboqs`
[2] `https://github.com/pqcli`
[3] `https://github.com/CBonnell/chameleon-certs`
[4] `https://github.com/IETF-Hackathon/pqc-certificates`

Formats based on parallel chains maximise flexibility and backwards compatibility by deploying separate PQ certificates alongside existing traditional X.509 certificates. Legacy systems continue to use the classical chain unmodified, safely ignoring the parallel PQ certificates, while updated systems simultaneously validate both. In theory, applications can negotiate and utilise only the certificates they support, ensuring uninterrupted operations and enabling a seamless, incremental migration. However, this can create additional security vulnerabilities.

Extension-based approaches achieve backwards compatibility by relying on non-critical extensions. Legacy validators ignore these unrecognised extensions and rely on traditional cryptography, while upgraded systems utilise the embedded PQ components. Although less flexible than parallel chains, extension-based formats reduce complexity by maintaining a single certificate chain.

3.4. Certificate Lifecycle

Pure and composite certificates require no modification to the standard certificate lifecycle. Pure certificates, containing a single algorithm, follow the exact lifecycle as classical ones. Composite certificates, despite combining two algorithms, are treated as a single unit through the use of a unique OID, which abstracts the underlying components. Even for revocation, if one component is compromised, the entire composite certificate must be revoked. This unified handling simplifies certificate management.

Hybrid with Extensions requires a two-step signing process: the alternative signature is computed first, and then the primary one signs over the extensions that contain the alternative signature and public key. Although merging two keys into a single certificate simplifies management, it breaks the traditional PKI model, where one certificate binds to one key. Validators must adapt accordingly—verifying either the native or the alternative signature based on extension support, but never both. Similar logic applies to CRLs. Since there is no way to revoke only one component, a compromise of either key necessitates revocation of the entire certificate. Moreover, once adopted by a root CA, the whole hierarchy should support these extensions. While operationally simpler than managing parallel chains, this model requires changes to the typical lifecycle.

Parallel certificate chains significantly increase lifecycle complexity. Each chain requires separate issuance, tracking, and validation, with custom CSR handling, policy alignment, and synchronised issuance order. Validity periods may differ, making the usable overlap the subscriber's responsibility. Rollover and renewal must be coordinated to avoid trust gaps. Verification logic becomes increasingly fragile, necessitating the discovery of linked certificates [63] and consistent cross-referencing. Revocation and audit processes are duplicated, and any desynchronisation—such as missing CRLs, inconsistent extensions, or out-of-order chains—can disrupt operations. Ultimately, managing parallel chains doubles the effort while increasing the risk of configuration errors.

3.5. Security Considerations

Pure certificates rely solely on PQ algorithms, eliminating downgrade risks and dependencies on traditional cryptography. While they offer a clean cryptographic break, their acceptance depends entirely on still-maturing PQ schemes.

Hybrid Composite is the only format explicitly engineered for cryptographic security during migration. Composite keys or signatures enforce dual use: all components, traditional and PQ, must validate for the certificate to be accepted. This ensures an attacker must break both traditional and PQ algorithms to forge a valid certificate, offering resilience against both conventional and quantum adversaries. That protection, however, comes with larger object sizes and introduces new OIDs, limiting backwards compatibility. Once cryptographically relevant quantum computers (CRQCs) become available, the traditional part becomes irrelevant, and composite certificates should transition to pure PQ algorithms.

By contrast, the other hybrid formats prioritise backwards compatibility over cryptographic strength. The formats embed the PQ component as auxiliary, while keeping a legacy-compatible key or signature in the primary fields. The standard for hybrid bound even states that it is not suitable for composite use [61]. This eases deployment but risks downgrade, as relying parties may validate only the traditional algorithm. An attacker can exploit this by forcing the use of the weaker traditional component, bypassing the PQ protection if downgrade safeguards are absent or ineffective. Security is thus limited to the

weakest component. These formats require strict validation policies and phased enforcement to avoid fallback to compromised primitives during migration.

3.6. Space Considerations

Maintaining backwards compatibility with traditional PKIs in space deployments presents significant challenges. While pure and composite formats avoid bandwidth waste by design—transmitting only what is strictly required—this is not the case for extension-based hybrids, where PQ components are always embedded and transmitted, even to legacy systems that ignore them. This leads to unnecessary overhead in constrained environments. Parallel chains through hybrid bound certificates may improve efficiency by allowing the selective transmission of either the traditional or PQ certificate, assuming the recipient's capabilities are known. However, managing parallel chains still doubles the revocation load, such as CRLs, which can be very difficult to manage in space networks.

Nevertheless, backwards compatibility should be secondary to security and interoperability within the federation. While it is considered desirable to enable "existing implementations that will not yet have been updated to support the PQ algorithms" [5], legacy PKI infrastructure in space is minimal, if any. Additionally, requiring strict backwards compatibility collides not only with bandwidth requirements, but also with security and interoperability. The limited timeframe [54] for PQ migration motivates prioritising quantum-resistant considerations from the outset.

In the federated space, a primary challenge is ensuring federation-level interoperability while maintaining cryptographic agility. Constrained space assets may not be capable of supporting all PQ algorithms due to their computational requirements. As a result, coordination is required to define a cryptographic profile tailored to the specific operational and performance constraints of federated space applications. However, the detailed development of such a profile falls outside the scope of this study.

Instead, attention turns to the challenge of divergent preferences for lattice-based schemes. Some agencies favour hybrid profiles with composite certificates, while others prefer pure lattice-based deployments. This creates a trade-off between interoperability—due to potentially unrecognised OIDs—and local policy autonomy. The wide range of composite combinations further demands a well-defined specification, as seen in OpenPGP [64]. To address these issues and enable an interoperable PQ certificate profile for federated space, this chapter concludes with potential mitigation strategies.

Mitigation 1 (Recommended)

Interoperability can be maintained by enforcing a single certificate profile, either pure or composite, across the federation. This ensures all parties recognise the same OIDs and parameters, eliminating cross-domain mismatches. A standardisation process would define the accepted algorithms, trading individual agency flexibility for consistent cryptographic behaviour and reduced interoperability risk. In the case of CCSDS, this can be achieved via updates to the Cryptographic Algorithms Blue Book [65].

Mitigation 2 (Not Recommended)

A complementary strategy combines alternative public-key and signature extensions with composite schemes. Here, the certificate's primary fields carry a unified pure PQ key and signature, ensuring baseline interoperability across the federation. Agencies requiring additional assurances or migration flexibility embed locally preferred composite algorithms in the non-critical alternative algorithm extensions. This allows each domain to apply its own security policies without disrupting global compatibility. Where overlap exists, entities can validate using the shared composite; otherwise, the primary PQ fields ensure continuity. The approach preserves local autonomy while avoiding the complexity of standardising a single composite profile for all participants.

Nevertheless, while theoretically viable, this configuration is impractical and is therefore not recommended. It results in the drawbacks of both approaches: a significant increase in certificate size as each certificate carries two PQ keys and two signatures, some of which may never be used due to optional extension semantics. The bandwidth overhead may not be tolerable in space networks.

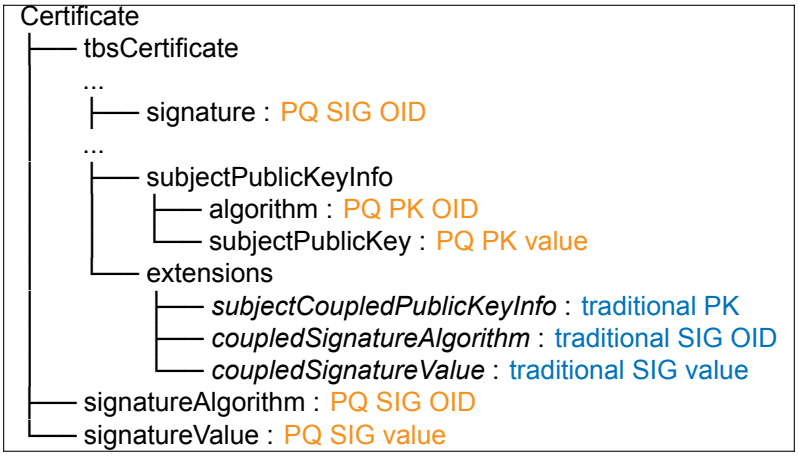
Mitigation 3 (Proposal)

This work proposes a new hybrid PQ certificate format, termed Hybrid Coupled, to enable optional composite use via defining extensions with AND semantics. Unlike the ITU-T alternative-algorithm extensions (Hybrid with Extensions), in which the components are used and verified disjointly, coupled extensions require both the primary and extended components to be validated and used together. Each component's signature is encoded in its respective field, following the structure and precedence similar to ITU-T rules; thus, the formats present similar overhead. This design preserves the simplicity of certificate management by avoiding separate chains. Moreover, as both components are jointly enforced, the compromise of either key requires the revocation of the certificate, aligning with the security model of composite schemes. Different configurations yield different properties:

- **Traditional-in-primary / PQ-in-extension** preserves backwards compatibility while enforcing composite-level security when extensions are supported. Unlike ITU-T extensions, which discard the traditional component in interpretation, Coupled extensions preserve and enforce it. Still, since the PQ component is in extensions, it has the same potential bandwidth waste as ITU-T extensions if they cannot be recognised.
- **PQ-in-primary / traditional-in-extension** creates an optional composite effect. For the space use case, federation-wide interoperability is maintained using only the PQ component, while local policies may enforce composite use by requiring extension validation. Nevertheless, these extensions should be marked as non-critical. Since traditional components are small, bandwidth waste is minimal, comparable to ITU-T extensions. This configuration is presented in Figure 3.6.

This flexible model balances transitional security and backwards compatibility, and in space systems, it could, in theory, be a potential solution to interoperability challenges during migration. Coupled extensions can be trivially deprecated post-transition (e.g., after CRQC), simplifying long-term maintenance.

Figure 3.6: X.509 Hybrid Coupled Certificate Format.



Nevertheless, the shortcomings are inherent in introducing a new format. The formalisation process is typically long. Furthermore, achieving widespread adoption and enforcing the use of newly developed extensions adds additional complexity, ultimately prolonging the deployment timeline.

Conclusion This study reviewed existing post-quantum certificate formats, each designed to prioritise either transition ease or enhanced security. Yet, none fully address the inherent tension between interoperability and local autonomy in federated environments. To bridge this gap, we proposed a new format — Hybrid Coupled — which provides optional composite-level security while maintaining compatibility through extensions. However, this may take years to potentially become a standard. Given that backwards compatibility is not a primary concern due to the limited deployment of certificates in space, the most practical mitigation (and the one most aligned with emerging standards) is to agree on a set of composite schemes, which is therefore recommended for space applications.

4

Federal Certificate Profiles

The X.509 open-ended extension mechanism enables the addition of new semantics to certificates. However, the unregulated use of extensions, combined with their context-specific parsing, poses challenges for both interoperability and implementations in space systems. This chapter reviews the Internet-standard extensions inherited by the CCSDS Authentication Credentials [7] and draws from terrestrial PKI profiles to propose a minimal set of tailored profiles with extension configurations for federated space environments such as IGCA.

4.1. Internet Extensions Review

The X.509 Internet profile defines fifteen general-purpose extensions and an additional two reserved for private Internet use by individual PKIs. Table 4.1 summarises the semantics of all seventeen extensions and the inclusion or processing rules that the profile imposes. It must be noticed:

> ”At a minimum, applications conforming to this profile **MUST** recognise the following extensions: *key usage*, *certificate policies*, *subject alternative name*, *basic constraints*, *name constraints*, *policy constraints*, *extended key usage*, and *inhibit anyPolicy*.” [6]

Table 4.1: RFC 5280 certificate extensions — semantics and inclusion requirements

Extension (OID)	Semantics and purpose	RFC 5280 requirement
Authority Key Identifier (2.5.29.35)	Identifies the public key that corresponds to the private key used to sign the certificate or CRL; accelerates path building and key rollover.	**SHOULD** be present in all non-self-signed CA and CRL certificates; applications **MUST** be able to parse it.
Subject Key Identifier (2.5.29.14)	Provides a unique identifier for subject’s public key, allowing it to be matched with private-key material and referenced by AKI.	**SHOULD** be present in all end-entity and CA certificates; applications **MUST** recognise it.
Key Usage (2.5.29.15)	Bitmap that restricts the cryptographic purposes for which the subject public key may be used (e.g. `digitalSignature`, `keyCertSign`).	**MUST** be present and marked *critical* in CA certificates; **SHOULD** be present in end-entity certificates that limit usage.
Certificate Policies (2.5.29.32)	Lists policy OIDs under which the certificate was issued; may include CPS pointers and user notices.	**SHOULD** be present when policies are asserted; relying parties **MUST** process it.

Table 4.1 (continued)

Extension (OID)	Semantics and purpose	Requirement
Policy Mappings (2.5.29.33)	Maps issuer-domain policy OIDs to subject-domain policy OIDs in CA certificates to support bridge PKI.	**MAY** appear only in CA certificates; processors **MUST** understand it.
Subject Alternative Name (2.5.29.17)	Carries additional identities for the subject (DNS, IP, URN, etc.); essential for TLS server names, e-mail addresses, spacecraft URIs, and similar identifiers.	If the subject DN is empty, SAN **MUST** be present and *critical*; otherwise **SHOULD** be used.
Issuer Alternative Name (2.5.29.18)	Analogous to SAN but for the issuer; rarely used outside specialised PKIs.	**MAY** appear; applications **MUST** be able to parse.
Subject Directory Attributes (2.5.29.9)	Encodes additional directory attributes (e.g. `dateOfBirth`, `gender`) for X.500 compatibility.	**MAY** appear; applications **MUST** be able to parse.
Basic Constraints (2.5.29.19)	Indicates whether the subject is a CA and, optionally, a path-length constraint.	**MUST** be present and *critical* in all CA certificates; **SHOULD NOT** appear in end-entity certificates unless CA=FALSE.
Name Constraints (2.5.29.30)	Enumerates permitted or excluded name sub-trees for subordinate CAs.	**MAY** appear only in CA certificates and **MUST** be marked *critical*; processors **MUST** enforce.
Policy Constraints (2.5.29.36)	Constrains acceptable certificate policies in a path (e.g. `requireExplicitPolicy`).	**MAY** appear only in CA certificates, *critical*; relying parties **MUST** enforce.
Extended Key Usage (2.5.29.37)	Lists application-specific purposes (`serverAuth`, `clientAuth`, `emailProtection`, etc.).	**MAY** appear in any certificate; if present, processors **MUST** enforce the indicated usages.
CRL Distribution Points (2.5.29.31)	Specifies one or more URIs from which the CRL for the certificate can be retrieved.	**SHOULD** be present when CRLs are published; processors **SHOULD** use the information.
Inhibit anyPolicy (2.5.29.54)	Disables use of the special `anyPolicy` OID after a specified depth in the path.	**MAY** appear only in CA certificates, *critical*; processors **MUST** enforce.
Delta CRL Distribution Point (2.5.29.46)	Identifies the location of delta-CRLs issued by the CA.	**MAY** appear; mainly used in CRLs rather than certificates.
Authority Information Access (1.3.6.1.5.5.7.1.1)	Provides access methods (URIs) for CA services such as OCSP or issuer- certificate download.	**MAY** appear; widely used for OCSP. Classified private-Internet because contents are PKI-specific.
Subject Information Access (1.3.6.1.5.5.7.1.11)	Analogous to AIA but for subject-supplied services.	**MAY** appear; private-Internet applications **MUST** be able to parse.

4.2. Federal Profiles

This section focuses on the extension configurations of federal profiles.

4.2.1. Use case: Federal Bridge Certification Authority

Bridge-oriented federations, such as the U.S. *Federal Bridge Certification Authority, FBCA* [66], are designed to connect independently governed PKIs without forcing them into a monolithic hierarchical trust model. Operated by the Federal PKI Management Authority (FPKIMA) and overseen by the Federal PKI Policy Authority (FPKIPA), the FBCA operates as a cross-certification hub, establishing trust relationships by cross-certifying with the principal CAs of member PKIs. This enables relying parties in one domain to build and validate certificate paths issued by another, up to the Federal Common Policy CA (FCPCAG2) root, thereby eliminating the need for bespoke trust-list handling. FBCA outlines specific certificate-extension profiles, including exact permitted policyObjectIdentifiers, such as mediumAssurance, PIV-I hardware, device hardware, the use of name constraints, signature algorithms, CRL/OCSP parameters, and audit requirements. These controlled profiles ensure mutual certificate interoperability: every relying party in any member PKI can validate certificates of any other member. FBCA relies on:

- an RFC 3647-structured Certificate Policy (CP); and
- a worksheet-style *extension profile* that tags every X.509 extension as *mandatory*, *optional*, or *forbidden*, ensuring deterministic behaviour across relying parties[66].

Each affiliated PKI, such as the *Foundation for Trusted Identity* (FTI) [67], closely aligns its certificate profiles with those of FBCA's for the specific applications it supports. The FTI is a non-profit Personal Identity Verification–Interoperable (PIV-I) credential provider cross-certified by FBCA. It issues PIV-I cards for businesses, containing cryptographic certificates signed by its CA that chain through the FBCA into the Federal Common Policy CA. FTI's process ensures federal agencies and other FBCA-compliant systems fully trust its PIV-I credentials.

Like the FBCA, the CCSDS Intergovernmental Certification Authority (IGCA) also relies on a bridge CA model. IGCA "can serve as a CA Bridge between member space agencies" and will "enable CAs to issue digitally signed certificates that can be used to secure access and communications paths." Adopting structured extension profiles similar to FBCA within IGCA could be beneficial:

1. *Predictable validation logic*: With every X.509 extension tagged as *mandatory*, *optional*, or *forbidden*, spacecraft and ground systems can rely on a deterministic code path for chain validation.
2. *Simplified implementations and efficient transport*: Constraining the field set both shrinks certificates (saving uplink bandwidth and on-board storage) and removes the need for generic extension parsers, reducing firmware size and complexity.
3. *Objective compliance and lifecycle management*: Explicit rules enable automated conformance checks and audits; any deviation—such as an unauthorised extension—can be flagged and blocked before it propagates, guaranteeing mission operations consistency.

4.2.2. Minimal Federal Profiles and Extension Configurations

Considering the requirements of the IGCA, the CCSDS Authentication Credentials [7]—which builds upon the X.509 Internet profile—and drawing from the FBCA and FTI. This work proposes a preliminary, minimal set of certificate profiles required to support a general bridge-architected PKI with applicability within space activities coordinated by IGCA. Their with extension configurations are outlined in Table 4.2, and include:

- *Self-Signed CA Certificate*: Root certificate that establishes the trust anchors.
- *Self-Issued CA Certificate*: Key rollover certificate (link certificate), used by CAs when transitioning from one key pair to another.
- *Cross Certificate*: Issued by a CA in one PKI domain to a CA in another PKI domain to enable interoperability through certificate policy mapping.
- *Intermediate Certificate*: CA certificate issued to a subordinate CA.

- *Signature Certificate*: Certificate where the subject is an end entity and the public key is used for signature verification.
- *Key Exchange Certificate*: Certificate where the subject is an end entity and the public key is used for key exchange protocols.

Table 4.2: Federal Profiles Extension Configurations. **M** - Mandatory, **O** - Optional, Empty - Disallowed. **P1** - Self-Signed; **P2** - Self-Issued; **P3** - Cross-Certificate; **P4** - Intermediate; **P5** - Signature; **P6** - Key Exchange.

Extension	**P1**	**P2**	**P3**	**P4**	**P5**	**P6**
Authority Key Identifier		M	M	M	M	M
Subject Key Identifier	M	M	M	M	M	M
Key Usage	M (crit.)	M (crit.)	M (crit.)	M (crit.)	M (crit.)	M (crit.)
Certificate Policies		M	M	M	M	M
Policy Mappings			M			
Subject Alternative Name	O	O	O	O	O	O
Issuer Alternative Name						
Subject Directory Attributes						
Basic Constraints	M (crit.)	M (crit.)	M (crit.)	M (crit.)		
Name Constraints				O (crit.)		
Policy Constraints			M (crit.)	O (crit.)		
Extended Key Usage					O	O
CRL Distribution Points		M	M	M	M	M
Inhibit anyPolicy			M (crit.)	O (crit.)		
Delta CRL Distribution Point						
Authority Information Access		M	M	M	M	M
Subject Information Access	M	M	M	M		

Currently, the CCSDS IGCA specification [5] and the CCSDS Authentication Credentials standard [7] define general requirements for X.509 use in space systems, covering both extension use and implementation. For example, root CA certificates are recommended to include few or no extensions. This work proposes explicitly fixed extension configurations per application, ensuring predictability and consistency across the federation while remaining compatible with CCSDS and, by extension, X.509 Internet profile requirements.

Within IGCA, extensions such as *Authority Key Identifier* (AKI) and *Subject Key Identifier* (SKI) enable deterministic path construction, allowing spacecraft and ground systems to link certificates efficiently. *Key Usage* is retained as a critical extension to guarantee that a non□conforming implementation will fail validation rather than misuse a key. *Certificate Policies* encode assurance levels for ground-based policy filtering, which is required for cross-agency trust. For cross-certificates and certain intermediates, *Policy Mappings*, *Policy Constraints*, and *Inhibit AnyPolicy* define accepted foreign policies and prevent unintended trust propagation; these are irrelevant to roots or end entities. *Basic Constraints*, also critical, enforce CA/EE role separation, while *Name Constraints*, optional, enable namespace scoping at intermediate level.

The optional inclusion of *Subject Alternative Name* follows the requirements of CCSDS Authentication Credentials [7] and provides flexibility to carry identifiers such as spacecraft names or mission-specific labels when needed, without imposing unnecessary parsing requirements on systems that do not require this data. Similarly, *Extended Key Usage* is retained only for end-entity certificates to specify the intended purpose of the public key, beyond the basic usage defined by the *Key Usage* extension. It provides a more granular control over how a certificate's public key can be used, enhancing security and preventing misuse.

CRL Distribution Points and *Authority Information Access* are maintained in line with the requirements of CCSDS IGCA to provide revocation and issuer discovery information, recognising that while spacecraft may not process these directly, they are essential for ground-based systems to support spacecraft in validating trust chains. *Subject Information Access* is similarly retained for CA certificates to help ground systems locate CA repositories and prepare the necessary data for spacecrafts.

All other extensions are explicitly disallowed to minimise certificate size and avoid unpredictable parsing on constrained platforms. By fixing the extension configurations, the profiles enable deterministic validation and policy-driven cross-federation trust, which ideally should be managed on the ground.

4.3. Considerations on Space Links

Despite the benefits of structured extension use, limitations remain. The current set does not cover specialised applications such as code signing or time-stamping, which can be integrated as needed. Additionally, the configuration relies on URI-based extensions, which might not be directly suitable for space. In some missions, the certificate and revocation information can be distributed to spacecraft via telecommand by ground control, rather than being automatically fetched by the spacecraft from a predefined location. Other missions, with more restrictive contact planning, may decide to deploy dedicated spacecraft elements that can support the distribution of certificate and revocation information to other spacecraft/users. Thus, the previously outlined profiles serve as a preliminary structured foundation for IGCA profile development, recognising that standardisation requires broader alignment.

Generally, constrained space hardware cannot be expected to accommodate the code size, memory, and processing overhead needed for full RFC 5280 path validation and the many extensions prescribed by federated profiles [68]. In practice, space links might be within a closed, *mission-local* trust domain that utilises a rigid certificate profile, potentially without any extensions. Thus, interoperability outside this domain might not be trivially achievable. In these settings, it is a matter of debate whether X.509 certificates should take a specific form or if they should be used at all, given that communication outside the local domain may not be a priority. That divergence duplicates engineering effort, complicates COTS integration, and stores up technical debt that will surface if the mission later needs to interoperate beyond its original domain.

Defining a rigid certificate profile that is broadly applicable across diverse space missions is inherently challenging. This stems from the varying technical requirements and policy mandates imposed by each mission. For example, severely constrained platforms that implement X.509 parsing in hardware often require fixed, offset-based parsing, as handling variable-length fields is non-trivial at the hardware level. To meet this constraint, fields must have deterministic lengths, potentially employing padding or other techniques to guarantee alignment. In another case, the absence of a reliable onboard time source renders certificate validity periods ineffective, necessitating alternative approaches such as offloading time validation to ground systems. As a result, many certificates adhering to the standard profiles outlined in the previous section may be difficult or impractical to support on such platforms. Moreover, some missions may face even stricter constraints, imposing additional limitations beyond those considered in existing profiles.

However, there are still optimisations and guidelines that can support mission designers in these efforts. While they do not constitute a complete profile, targeted adaptations to the current standards and X.509 certificates can support the implementation for constrained spacecraft within an isolated trust domain:

4.3.1 `subjectPublicKeyInfo`, `signatureAlgorithm`, `signature`, and `signatureValue`:
The profile **SHOULD** have fixed algorithms.

Rationale: Enforcing the use of a single algorithm at a profile level enables implementations to benefit from reduced complexity of OID parsing and preserves overall certificate offset alignment.

4.3.2 `issuer` and `subject`:
Name forms **SHOULD** be a non-empty distinguished name containing only the common name attribute, which **MUST** be a UTF8String or PrintableString that is guaranteed to have a fixed length. They **MUST NOT** use multi-attribute relative distinguished names.

Rationale: Similarly, this enables implementations to benefit from reduced complexity of OID parsing, as each name attribute is identified by an OID. Limiting issuer and subject DNs to a fixed `commonName (CN)` yields a deterministic, easily parsed structure. The rule also eliminates little-used legacy constructs—such as multi-attribute relative distinguished names—thereby simplifying implementation while remaining syntactically compatible with the standard X.509.

4.3.3 `issuerUniqueID` and `issuerUniqueID`:
The `tbsCertificate` **MUST NOT** contain the optional fields for unique identifiers.

4.3.4 `extensions`:
Extensions **SHOULD** be used infrequently, and only when really necessary.

Rationale: Since the communication is isolated within a single trust domain, the additional extensions could not bring tangible benefit, as the ground control could perform complicated operations. Therefore, removing extensions reduces certificate size and eliminates unnecessary parsing and validation overhead on the device.

Conclusion The uncoordinated use of extensions poses significant challenges to interoperability at the federation scale. This chapter proposes a structured model for profile definition, identifying a minimal set of certificate classes and extension configurations that can enhance cross-domain compatibility in federated environments, such as IGCA. While this approach strengthens interoperability, mission-specific constraints may still require deviations—introducing specialised extensions or stricter parameters—which can hinder federation-wide consistency. Ultimately, such specialisation is often unavoidable, but it remains the responsibility of each mission authority to justify and manage these deviations within the broader trust framework.

5

c509-native: A Tool for CBOR-Encoded Certificates

C509 is an emerging certificate profile, and as a result, open-source support and adoption remain limited. This chapter presents c509-native, the first implementation to support natively signed, post-quantum (PQ) C509 certificates. Beyond certificate generation and parsing, the tool provides functionality for Certificate Signing Requests (CSRs) and Certificate Revocation Lists (CRLs), enabling preliminary certification authority (CA) operations entirely within a CBOR-encoded object stack. The chapter outlines the design rationale, implementation approach, and command-line interface of the tool.

5.1. Requirements and Design

c509-native should enable the following *functional requirements*:

- Generation, verification and signing of C509 certificates, CSRs and CRLs.
- A Command-Line Interface (CLI) mirroring a restricted set of the OpenSSL workflows.
- Pure ML-DSA and ML-KEM certificates and composite with ECDSA and ECDH, respectively.
- All objects shall use deterministic CBOR to guarantee identical ”to-be-signed” data.
- All extensions from the C509 Extension Registry [8] should be supported.

c509-native should enable the following *non-functional requirements*:

- The implementation shall operate without dynamic memory allocation.
- The implementation shall be written in C++, yet with minimal use of the standard libraries and object-oriented features like inheritance, dynamic dispatches or runtime exceptions.
- Licensing shall remain permissive (MIT License) to encourage integration by third parties.
- Parsing and serialising shall be unit tested.
- The schema-driven generation using zcbor should be integrated into the build workflow (required by the experiments in the next chapter).

The tool intends to serve as a proof-of-concept, not to be deployed as operational code (potentially subject to certification) in future space missions.

The modules and the dependencies between them and external libraries are illustrated in Figure 5.1. c509-native is organised around the following components:

- **structures**: Definitions that mirror the schema from the C509 draft [8]; includes minimal printing.
- **codec**: The CBOR (de)serialiser specific to C509 objects, which delegates primitive encoding to the lightweight zcbor library, which remains encapsulated within the module.

- **crypto**: An utility module for loading OpenSSL. It loads classic algorithms and, through the `oqs-provider`, post-quantum (or hybrid). Ideally, this module should form a complete abstraction layer over the cryptographic operations.
- **core**: The main logic of the tool. Coordinates the end-to-end flow: key generation, certificate/CRL construction, and calls to crypto for signatures and codec for (de)serialisation. It exposes a small, stable Application Programming Interface (API) that the CLI can use.
- **cli**: The user-facing command-line interface. It employs the external `argparse` library to parse options and dispatches each valid command to the appropriate routine in **core**. `brotli` compression is introduced for experimental purposes only and is contained in this layer. Brotli is a lossless compression algorithm that combines LZ77, Huffman coding, and second-order context modelling and is efficient for structured and repetitive data [69].

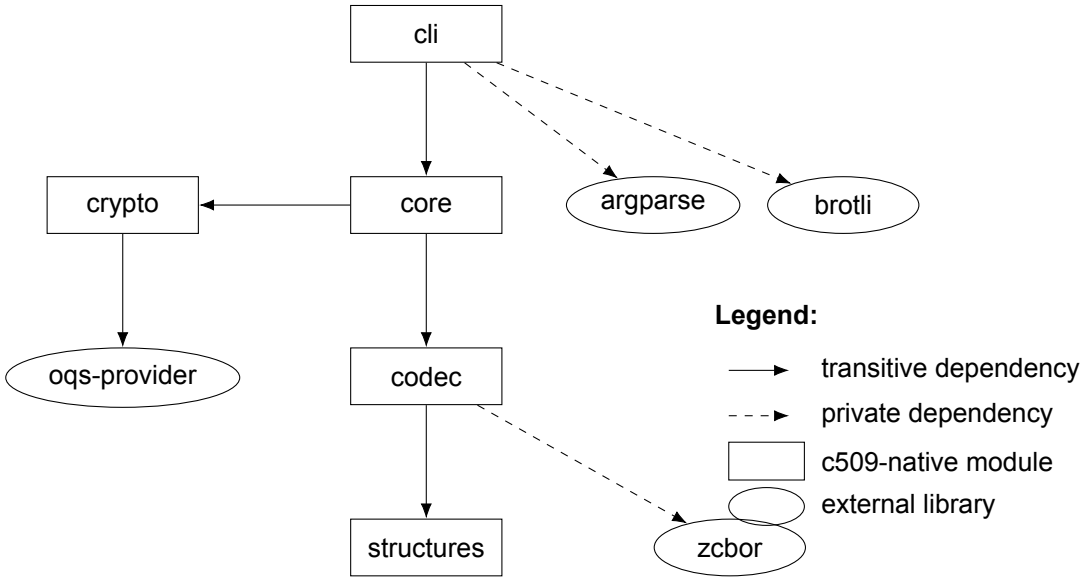


Figure 5.1: Design of the `c509-native` tool.

5.2. Implementation

The parsing and serialisation of the Certificate Serial Number (CSN) is presented to showcase how `c509-native` is implemented. Each structure is defined in its header file, and each structure that requires arrays has its bounds statically specified, as displayed in Listing 5.1. The sizes are either taken from the standard or selected empirically, as no specified upper bound has been identified. Nevertheless, these values are intended to be adjusted according to the specific application.

Listing 5.1: `c509-native` Certificate Serial Number Structure.

```
#define MAX_CSN_BYTES 20

namespace C509 {
    struct CertificateSerialNumber {
        bounded_array<uint8_t, MAX_CSN_BYTES> bytes;
        std::string to_string() const;
    };
}
```

For each structure, a pair of `encode` and `decode` functions are defined, referred to as *codec*, as exemplified in Listing 5.2. Each error path is uniquely identified by an error code. The error code format is `509'<structure_id>'<operation><error_id>`, where `<operation>` is `0` for encoding and `1` for decoding. For primitive CBOR encoding, `zcbor` functions and macros are used. It should be noticed that `ZCBOR_ERR` does not cause any runtime exception. While a codec is implemented for each structure, only a subset is exported outside the module.

Listing 5.2: `c509-native` Certificate Serial Number Codec.

```
#define C509_ERR_CSN_ENC_BSTR           509'000'000

#define C509_ERR_CSN_DEC_BSTR           509'000'100
#define C509_ERR_CSN_DEC_INVALID_LENGTH 509'000'101
#define C509_ERR_CSN_DEC_BUFFER_ERROR   509'000'102

bool CBORCodec<CertificateSerialNumber>::encode(zcbor_state_t *state, const
    CertificateSerialNumber &input) {
    if (!zcbor_bstr_encode_ptr(state, reinterpret_cast<const char *>(input.bytes.data()),
        input.bytes.size()))
        ZCBOR_ERR(C509_ERR_CSN_ENC_BSTR);

    return true;
}

bool CBORCodec<CertificateSerialNumber>::decode(zcbor_state_t *state, CertificateSerialNumber
     &output) {
    zcbor_string str{};
    if (!zcbor_bstr_decode(state, &str))
        ZCBOR_ERR(C509_ERR_CSN_DEC_BSTR);

    if (str.len > MAX_CSN_BYTES)
        ZCBOR_ERR(C509_ERR_CSN_DEC_INVALID_LENGTH);

    if (!output.bytes.copy(str.value, str.len))
        ZCBOR_ERR(C509_ERR_CSN_DEC_BUFFER_ERROR);

    return true;
}
```

The functions of the **core** API are illustrated in Listing 5.3. The **core** module defines all the registries in the C509 specification and uses them in the logic implemented for the outlined functions. The implementation of each function represents a sequence of `OpenSSL` calls, such as loading the statically built `oqs-provider`, calling the `EVP_PKEY` API for key generation or signing, and utilising the exported codecs to serialise and deserialise data, as illustrated in the simplified implementation of `key_gen` in Listing 5.4—in the real implementation, each function call is guarded.

Listing 5.3: `c509-native` Core API.

```
#include "registry.hpp"

bool map_alg_to_id(const std::string &algorithm, C509::AlgorithmIdentifier &identifier);
bool map_id_to_alg(std::string &algorithm, const C509::AlgorithmIdentifier &identifier);

bool key_gen(...);
bool csr_gen(...);
bool csr_verify(...);
bool csr_sign(...);
bool csr_self_sign(...);
bool crl_gen(...);
bool crl_revoke(...);
```

Listing 5.4: `c509-native` Key Gen.

```
bool key_gen(const std::string &algorithm, uint8_t *private_key_out, size_t &
    private_key_out_size) {
    OSSL_LIB_CTX *oqs_provider_ctx = load_oqs_provider();

    EVP_PKEY_CTX *pkey_ctx = EVP_PKEY_CTX_new_from_name(oqs_provider_ctx, algorithm.c_str(),
        OQSPROV_PROPQ);

    EVP_PKEY_keygen_init(pkey_ctx);

    EVP_PKEY *key = nullptr;
    EVP_PKEY_generate(pkey_ctx, &key);

    C509::C509PrivateKey private_key{};

```

```
    map_alg_to_id(algorithm, private_key.subject_private_key_algorithm);
    C509::PrivateKey *priv_k = &private_key.subject_private_key_info.private_key;

    EVP_PKEY_get_raw_private_key(key, priv_k->bytes.data_p(), priv_k->bytes.len_p());

    C509::PublicKey *pk = &private_key.subject_private_key_info.public_key;
    EVP_PKEY_get_raw_public_key(key, pk->bytes.data_p(), pk->bytes.len_p();

    cbor_encode(private_key_out, private_key_out_size, &private_key, &private_key_out_size);

    EVP_PKEY_free(key);
    EVP_PKEY_CTX_free(pkey_ctx);
    OSSL_LIB_CTX_free(oqs_provider_ctx);

    return true;
}
```

The commands defined by the **cli** layer are illustrated in Listing 5.5. Each command defines a pre-processing function that parses and prepares the arguments, and another function that handles the command by calling the functions from the **core** module, as presented for `genpkey` in Listing 5.6.

Listing 5.5: `c509-native` Commands.

```
void setup_genpkey_parser(argparse::ArgumentParser &genpkey_cmd);
void setup_req_parser(argparse::ArgumentParser &req_cmd);
...
int handle_genpkey(const argparse::ArgumentParser &genpkey_cmd);
int handle_req(const argparse::ArgumentParser &req_cmd);
...
int main(int argc, char *argv[]);
```

Listing 5.6: `c509-native` Key Gen Command.

```
int handle_genpkey(const argparse::ArgumentParser &genpkey_cmd) {
    const auto algorithm = genpkey_cmd.get<std::string>("-algorithm");
    const auto out_file = genpkey_cmd.get<std::string>("-out");

    uint8_t private_key_out[MAX_BUFFER_SIZE] = {};
    size_t private_key_out_size = sizeof(private_key_out);

    if (!key_gen(algorithm, private_key_out, private_key_out_size)) {
        std::cerr << "Error: Key generation failed.\n";
        return 1;
    }

    write_binary_file(out_file, private_key_out, private_key_out_size);
    return 0;
}

void setup_genpkey_parser(argparse::ArgumentParser &genpkey_cmd) {
    genpkey_cmd.add_argument("-algorithm")
            .required()
            .help("The public key algorithm (ML-DSA / ML-KEM versions supported by OQS-
                provider)");
    genpkey_cmd.add_argument("-out")
            .required()
            .help("Output file");
}
```

`c509-native` achieves 60% line coverage and 53% branch coverage over 210 automated tests. The test suite primarily targets low-level components—such as serial numbers, time fields, OIDs, and names—since higher-level constructs (e.g., certificates, CSRs, CRLs) are composed from these elements. Due to time constraints, comprehensive branch coverage, particularly across complex codec and core logic paths, remains incomplete and is left for future work.

5.3. Command-Line Interface

The `c509-native` CLI is designed to support the commands illustrated in Listing 5.5. Each of the subcommands is designed to mimic the workflows from OpenSSL, implementing a similar argument

precedence. For instance, `req`, as presented in Listing 5.8, is responsible for generating and processing signing requests, with different argument combinations leading to distinct workflows.

First, the routine distinguishes between creation and processing paths. If either `-new` is set explicitly or implicitly (i.e., -c509 is requested without an existing input), the command generates a fresh certificate-signing request (CSR). In this branch, the subject name—supplied non-interactively via `-subj` or interactively unless `-batch` is given—together with any additional extensions (`-addext`) are encoded, and the designated private key (`-key`) is loaded. The CSR is written to the output buffer. Otherwise, an existing CSR is loaded from `-in`; optional verification is performed when `-verify` is supplied, while subject or extension rewrites are explicitly disallowed to preserve request integrity.

The second stage is entered only when `-c509` is active, indicating that a full certificate, rather than a CSR, is required. Here, the command enforces additional constraints: a serial number (`-set_serial`) and validity period (`-days`, default 365) must be provided, and any previously loaded CSR must already have passed signature verification unless it was freshly generated in the same invocation. If both `-CA` and `-CAkey` are specified, the CSR is signed by the issuing certificate authority; otherwise, a self-signed certificate is produced using the requestor's key. In both cases, the resulting C509 certificate is emitted to `-out`, optionally wrapped in Brotli compression when the `-compressed` flag accompanies the request. Extensive argument validation throughout the flow ensures that incompatible options (e.g., modifying the subject of an imported CSR) are rejected with precise diagnostics.

Listing 5.7: `c509-native` Usage.

```
Usage: c509_cli [--help] [--version] {crl,genpkey,parse,req}

Optional arguments:
  -h, --help     shows help message and exits
  -v, --version  prints version information and exits

Subcommands: ...
```

Listing 5.8: `c509-native` Usage.

```
Usage: c509_cli req [--help] [--version] [-in VAR] [-verify] [-new] [-c509] [-CA VAR] [-CAkey
     VAR] [-subj VAR] [-days VAR] [-set_serial VAR] [-addext VAR...] [-key VAR] [-out VAR] [-
    batch] [-compressed]

Optional arguments:
  -h, --help     shows help message and exits
  -v, --version  prints version information and exits
  -in            C509 request input file
  -verify        Verify self-signature on the request
  -new           New request
  -c509          Output an C509 certificate structure instead of a cert request
  -CA            Issuer cert to use for signing a cert, implies -c509
  -CAkey         Issuer private key to use with -CA
  -subj          Specify the subject (Distinguished Name) in OpenSSL format: "/C=XX/ST=State/
      L=City/O=Organization/OU=OrgUnit/CN=CommonName/emailAddress=email@example.com".
  -days          Number of days cert is valid for. Default: 365 days [nargs=0..1] [default:
      365]
  -set_serial    Serial number to use
  -addext        Additional cert extension key=value pair [nargs: 0 or more]
  -key           Key for signing, and to include unless -in given
  -out           Output file
  -batch         Do not ask anything during request generation
  -compressed    Use Brotli compression
```

The source code of `c509-native` is open-sourced[1] under the MIT License.

Conclusion `c509-native` is still a prototype. While it represents an initial step toward addressing the lack of tooling that supports C509 certificates, it does not implement advanced functionality—such as tracking the issued certificates—already offered by mature projects like OpenSSL. Even so, the current implementation provides enough capability to generate CBOR-encoded objects and thus serves as a preliminary reference implementation for the analysis in the next chapter.

[1] `https://github.com/rosualinpetru/c509-native`

6

X.509 vs. C509: An Empirical Comparative Analysis

C509 promises reduced certificate size and implementation footprint compared to traditional X.509—benefits that are particularly relevant in constrained environments. This chapter presents a comprehensive empirical trade-off evaluation between the two profiles along two key dimensions: object size and implementation complexity. For object size, semantically identical certificates and revocation lists are generated under both profiles, across both traditional and post-quantum cryptographic schemes. For implementation complexity, open-source and proprietary parsers are statically analysed on metrics that quantify code footprint and implementation difficulty. Combined, the following results can offer valuable insights into the feasibility of adopting C509 within space systems.

6.1. Object Size

Key findings Switching from the classical **X.509/DER** encoding to the streamlined **C509/CBOR** profile can significantly reduce bandwidth consumption and storage footprint in some cases.

Headline Results

- **Significant reduction for classical certificates:** *ECDSA* certificates are reduced by **40–45%**, owing to compact CBOR framing and streamlined structure.
- **Limited gains for post-quantum artefacts:** *ML-DSA:44* certificates exhibit modest reductions of **≈6%** as large keys and signatures dominate overall size.
- **Substantial savings for large CRLs:** CBOR-encoded revocation lists are reduced by **≈60%** beyond **1 000** entries, irrespective of the signature algorithm.
- **Reduced redundancy:** X.509/DER has more redundancy for Brotli to achieve **15–25%** compression for certificates and up to **63%** for CRLs.; C509 yields less than **13%** further gain.

6.1.1. Experimental Setup

To support the headline results with fair, reproducible, and isolated measurements, a controlled experimental environment was established. The evaluation considers X.509- and C509-profiled certificates, as well as DER- and CBOR-encoded revocation lists (CRLs), respectively. Both were analysed under traditional and PQC suites, while ensuring strict semantic equivalence between each pair of objects. Brotli compression was applied as a proxy to evaluate redundancy. Table 6.1 summarises the experimental setup. Experimental objects and code can be found in the repository of `c509-native`[1].

[1] `https://github.com/rosualinpetru/c509-native`

Table 6.1: Experimental Setup Summary

Aspect	Setting	Description / Rationale
Profile	X.509 v3 vs. C509 v2	X.509 v3 certificates were generated using `OpenSSL` and C509 v2 (natively-signed) were generated using `c509-native`, both according to Section 4.2. CRLs with 1 to 30K entries were generated similarly, with CDDL schema defined in Appendix B.
Cryptographic Suite	• ECDSA:P-256 • ML-DSA-44 • ML-KEM-512	Captures both traditional and PQC to evaluate the profiles' impact under varying key and signature sizes. While the choice of the algorithms impacts the absolute values, the relative differences remain consistent.
Content Normalisation	Pair-wise identical: • serial numbers • distinguished names • time fields • public key • signature algorithm • extension configuration and content	Isolate encoding effects by holding semantic content constant. All subject and issuer names are limited to a single `Common Name (CN)` attribute. All time fields are set beyond 2050 to use `GeneralizedTime`. The signatures differ due to the distinct signed content; the same private key was used to maintain consistency. For certificate revocation lists, no extensions are used.
Compression Probe	Brotli (level 11, window size 22)	Empirically assesses structural and syntactic redundancy in encoded objects.

6.1.2. Results

The analysis will evaluate certificates and CRLs separately under both traditional and post-quantum algorithms. It is important to note that while C509 certificates are currently undergoing standardisation as an IETF draft, CBOR-encoded CRLs are not yet under formal development. This work identified a preliminary CDDL definition for CRLs, the status of which at the time of writing is provided in Appendix B.

Traditional Certificates

For traditional certificates, C509 achieves substantial size reductions of approximately 40–45% compared to X.509, as illustrated in Table 6.2. For example, a self-signed ECDSA certificate decreases from 424 bytes (X.509) to 232 bytes (C509), representing a 45.3% reduction. While the C509 draft reports reductions exceeding 50% in some scenarios, the results presented here consistently fall within the 40–45% range. These differences are attributable to variations in reference profiles (RFC7925 [28] vs. Section 4.2, respectively) regarding the combination of extensions and name forms.

Table 6.2: Certificate sizes (bytes), relative size reductions (%) and Brotli compression rates (%) for X.509 and C509 traditional (ECDSA/ECDH with `secp256r1`) certificates according to the extension configuration presented in Section 4.2.

Certificate	Absolute Size			Brotli-Compressed Size			
	X.509	*C509*	*Rel. Red. (%)*	*X.509*	*(%)*	*C509*	*(%)*
Self-Signed	424	232	45.3	353	16.7	236	-1.7
Self-Issued	578	323	44.1	457	20.9	288	10.8
Cross	631	358	43.2	530	16.0	313	12.3
Intermediate	581	334	42.5	438	24.6	290	13.2
Signature	497	290	41.6	426	14.3	277	4.5
Key Exchange	491	284	42.1	419	14.7	281	1.1

Pre-quantum cross-certificates that include the most complex extension configurations are the largest in absolute size, and they present a lower relative reduction of 1–2% compared to self-signed and self-issued certificates. This is explained by C509's use of context-based optimisations, such as the null

issuer encoding when the issuer and the subject are identical, thus avoiding duplication. The savings stem primarily from C509's compact CBOR encoding, the replacement of verbose OIDs with integer identifiers, and optimisations that reduce the redundancy inherent to DER-encoded, X.509 certificates.

Regarding redundancy, traditional X.509 certificates achieve 14–25% size reduction under compression, while C509 shows a consistently lower reduction of 0–13%, as shown in Table 6.2. The lower rate for C509 indicates proximity to its entropy floor, demonstrating reduced structural and syntactic redundancy. For example, DER's repetitive OID patterns (e.g., 2.5.29.* prefixes) are replaced by compact integers in C509. Brotli's fixed overhead disproportionately impacts small C509 certificates due to their minimal redundancy, which can occasionally yield negative compression. Consequently, the structure and encoding of X.509 introduce redundancy that C509 explicitly seeks to eliminate.

C509's minimised design becomes evident through a field-level analysis, as shown in Table 6.3 for cross certificates. The breakdown confirms that every data item is reduced by at least one byte, thanks to the elimination of overheads associated with tag-length-value (TLV) format and redundant nested structures. For instance, fields like `version` and `serialNumber` are collectively reduced from 8 bytes in X.509 to just 3 bytes in C509. This is primarily due to CBOR's compact encoding and its ability to represent data items in a single byte, whereas DER always requires at least three.

Table 6.3: Field-level size comparison (bytes) between X.509 and C509 cross certificates using ECDSA with `secp256r1`. Each internal node is mapped to the number of bytes required to describe that element without the nested value. Leaves of the tree are mapped to the total number of bytes contained by that element.

X.509 (Structure)	C509 (Structure)	X.509	C509
Certificate	Certificate	4	1
├── tbsCertificate	│	4	0
│ ├── version	├── version	5	1
│ ├── serialNumber	├── serialNumber	3	2
│ ├── signatureAlgorithm	├── signatureAlgorithm	12	1
│ ├── issuer *(commonName)*	├── issuer *(commonName)*	28	16
│ ├── validity	│	2	0
│ │ ├── notBefore	├── notBefore	17	9
│ │ └── notAfter	├── notAfter	17	9
│ ├── subject *(commonName)*	├── subject *(commonName)*	29	17
│ ├── subjectPublicKeyInfo	│	2	0
│ │ ├── algorithm	├── publicKeyAlgorithm	21	1
│ │ └── subjectPublicKey	├── publicKeyValue	68	67
│ └── extensions	├── extensions	8	1
│ ├── *keyUsage*	│ ├── *keyUsage*	16	3
│ ├── *basicConstraints*	│ ├── *basicConstraints*	14	2
│ ├── *subjectKeyIdentifier*	│ ├── *subjectKeyIdentifier*	31	22
│ ├── *authorityKeyIdentifier*	│ ├── *authorityKeyIdentifier*	33	22
│ ├── *crlDistributionPoints*	│ ├── *crlDistributionPoints*	42	26
│ ├── *AIA*	│ ├── *AIA*	58	33
│ ├── *SIA*	│ ├── *SIA*	58	34
│ ├── *certificatePolicies*	│ ├── *certificatePolicies*	18	6
│ ├── *policyMappings*	│ ├── *policyMappings*	23	11
│ ├── *policyConstrains*	│ ├── *policyConstrains*	17	5
│ └── *inhibitAnyPolicy*	│ └── *inhibitAnyPolicy*	15	3
├── signatureAlgorithm	│	12	0
└── signatureValue	└── signatureValue	74	66

Some fields, such as `tbsCertificate`, `subjectPublicKeyInfo`, and `signatureAlgorithm`, are not represented in C509. This is because C509 aims to eliminate excessive nesting in the former and avoid duplication in the latter. Additionally, although not shown in the table, the `issuerUniqueID` and `subjectUniqueID` fields are not supported in C509, minimising features and certificate size.

The `issuer` and `subject` fields also experience substantial reductions—by nearly half—even when encoding only a single `Common Name (CN)` attribute in each. These savings stem primarily from the elimination of redundant DER constructs, such as the nested `SET` and `SEQUENCE` wrappers mandated by the `Relative Distinguished Name (RDN)` hierarchy in X.509. C509, by contrast, restricts the use of multiple attributes at each level of the hierarchy specifically to avoid this overhead. Furthermore, C509 replaces verbose object identifiers with compact integers. For commonly used OIDs—particularly those under the X.500 Directory Services arc (e.g., 2.5.4.3 for CN)—each substitution saves approximately six bytes per OID, significantly contributing to the overall reduction in naming field size.

The benefit of OID replacement extends to extensions, where OIDs are often a significant source of overhead. For example, `keyUsage` is reduced from 16 bytes to just 3 bytes, `basicConstraints` from 14 to 2 bytes, `subjectKeyIdentifier` from 31 to 22 bytes, and `certificatePolicies` from 18 to 6 bytes. These reductions result from a combination of factors: OIDs are substituted with dedicated integer identifiers, whose sign is used to mark criticality; the elimination of nested `SEQUENCE` and `OCTET STRING` wrappers; and the use of specialised encoding tailored to each extension's data type. Thus, C509 extensions are flattened into concise key-value pairs to avoid unnecessary layering.

The validity block, comprising `notBefore` and `notAfter`, is halved in size, shrinking from 36 bytes in X.509 to 18 bytes in C509. This reduction is primarily due to C509's use of CBOR integer encoding for timestamps, which represents dates as compact epoch-based integers. In contrast, X.509 encodes time using verbose, ASCII-based `GeneralizedTime` strings, each wrapped in additional tagging and length fields, which significantly increases the overhead.

Fields that carry raw cryptographic material—such as `subjectPublicKeyInfo` and `signatureValue`—do not benefit as significantly, since the underlying cryptographic algorithm determines their payload. However, the current draft introduces algorithm-specific optimisations, such as using point compression for elliptic-curve-based cryptography, or omitting the exponent for RSA when it is 65537, thereby encoding only the modulus within the certificate. These optimisations reduce the overall size of cryptographic content, though they are currently defined only for traditional cryptography.

In total, the C509 version of this cross certificate is 273 bytes smaller than its X.509 counterpart, resulting in a reduction of over 43.2%. This detailed breakdown demonstrates how C509 achieves its goal of eliminating structural and syntactic redundancy of certificates, without altering the semantic content.

Post-Quantum Certificates

While C509 achieves notable size reductions for traditional certificates, they diminish significantly in the PQ context. As shown in Table 6.4, size savings for PQ certificates remain below 6% across all tested profiles. Still, the compression trends mirror those observed for traditional certificates, reinforcing C509's decreased redundancy.

Table 6.4: Certificate sizes (bytes), relative size reductions (%) and Brotli compression rates (%) for X.509 and C509 pure PQ (ML-DSA-44 and ML-KEM-512) certificates according to the extension configuration presented in Section 4.2.

Certificate	Absolute Size			Brotli-Compressed Size			
	X.509	*C509*	*Rel. Red. (%)*	*X.509*	*(%)*	*C509*	*(%)*
Self-Signed	4019	3855	4.1	3965	1.3	3854	0.0
Self-Issued	4174	3946	5.5	4037	3.3	3901	1.1
Cross	4227	3981	5.8	4096	3.1	3928	1.3
Intermediate	4177	3957	5.3	4048	3.1	3911	1.2
Signature	4094	3913	4.4	4038	1.4	3902	0.3
Key Exchange	4076	3945	3.2	4028	1.2	3934	0.3

C509's size reductions for PQ certificates are bounded by a profile-specific upper limit, as demonstrated in Table 6.5. Across all evaluated combinations—including pure PQ and hybrid composite formats using ML-DSA and ML-KEM, with or without ECDSA and ECDH with `secp256r1`, respectively—the absolute reduction consistently falls between 180 and 197 bytes. This overhead is fixed by structural and syntactic inefficiencies, as C509 cannot further reduce the size of the cryptographic content. However, as PQ keys and signatures dominate the certificate size, the savings become marginal.

Table 6.5: Absolute size reductions (bytes) for C509 and X.509 pure PQ/hybrid composite end-entity signature or key encapsulation certificates according to Section 4.2 for ML-DSA±ECDSA:`secp256r1` and ML-KEM±ECDH:`secp256r1`.

Signature	Public Key	X.509	C509	Difference
Security Level 1/2				
mldsa44	mldsa44	4 094	3 913	181
mldsa44	mldsa44_ecdsa_p256	4 173	3 980	193
mldsa44	mlkem512	3 576	3 395	181
mldsa44	ecdh_p256_mlkem512	3 647	3 466	181
mldsa44_ecdsa_p256	mldsa44	4 181	3 998	183
mldsa44_ecdsa_p256	mldsa44_ecdsa_p256	4 259	4064	195
mldsa44_ecdsa_p256	mlkem512	3 664	3 479	185
mldsa44_ecdsa_p256	ecdh_p256_mlkem512	3 734	3 550	184
Security Level 3				
mldsa65	mldsa65	5 623	5 442	181
mldsa65	mldsa65_p256	5 702	5 509	193
mldsa65	mlkem768	4 849	4 668	181
mldsa65_ecdsa_p256	mldsa65	5 710	5 528	182
mldsa65_ecdsa_p256	mldsa65_p256	5 790	5 593	197
mldsa65_ecdsa_p256	mlkem768	4 936	4 753	183
Security Level 5				
mldsa87	mldsa87	7 581	7 400	181
mldsa87	mldsa87_ecdsa_p384	7 692	7 499	193
mldsa87	mlkem1024	6 551	6 370	181
mldsa87_ecdsa_p384	mldsa87	7 700	7 517	183
mldsa87_ecdsa_p384	mldsa87_ecdsa_p384	7 811	7 616	195
mldsa87_ecdsa_p384	mlkem1024	6 670	6 487	183

Consequently, while C509 achieves significant reductions for traditional certificates, its impact on PQ certificate size is fundamentally constrained by the underlying cryptographic primitives, an aspect that lies beyond the scope of C509's design objectives.

Traditional Certificate Revocation Lists

When signed with traditional cryptography, CBOR-encoded CRLs demonstrate substantial reductions compared to their DER-encoded counterparts, as shown in Table 6.6. A reduction of 58% is already observed at 100 revoked certificates, with the savings stabilising around 60% as the number of entries increases to 30,000. The apparent constant increasing factor for absolute sizes under each encoding may not generalise to real-world deployments, where each revocation entry may vary in the amount of information by embedding extensions. The primary contributor is CBOR's efficient time representation, which halves the size of each timestamp compared to DER's `GeneralizedTime`.

Table 6.6: Revocation lists sizes (bytes), relative reductions (%) and Brotli compression rates (%); signed with ECDSA:`secp256r1`.

Revoked Certificates	Absolute Size			Brotli-Compressed Size			
	DER	*CBOR*	*Rel. Red. (%)*	*DER*	*(%)*	*CBOR*	*(%)*
1	183	107	41.5	182	0.5	111	-3.7
10	413	197	52.3	305	26.2	201	-2.0
100	2 664	1 098	58.8	1 204	54.8	1 032	6.0
1 000	25 159	10 100	59.9	9 854	60.8	9 016	10.7
10 000	250 118	100 099	60.0	94 759	62.1	87 915	12.2
20 000	500 066	200 099	60.0	188 137	62.4	175 522	12.3
30 000	750 035	300 100	60.0	281 076	62.5	263 050	12.3

In terms of redundancy, the repetitive and nested encoding of revocation entries in DER results in high structural overhead, which is reflected in its significant Brotli compression rates, exceeding 60% for

larger lists. In contrast, CBOR results in minimal additional compression gain, less than 13% stemming from the representation of each revocation entry. While extensions were included in this evaluation, the relative advantage of CBOR is expected to be more pronounced owing to OID substitution.

The results confirm that CBOR-encoded CRLs provide a consistent and significant reduction when signed with traditional cryptography. The streamlined syntax and structure enable saving of up to 60%. However, it must be determined to what extent these benefits apply to PQC.

Post-Quantum Certificate Revocation Lists

When PQ signatures are used, CBOR-encoded CRLs continue to demonstrate meaningful improvements, as shown in Table 6.7. While the relative reductions begin modestly—2.8% for a single revoked certificate and 7.6% for ten—they grow rapidly, reaching 59.8% at 30,000 entries. This mirrors the trend from traditional CRLs, indicating the signature's size becomes minor in large lists.

Table 6.7: Revocation lists sizes (bytes), relative reductions and Brotli compression rates (%) signed with ML-DSA:44.

Revoked Certificates	Absolute Size			Brotli-Compressed Size			
	DER	*CBOR*	*Rel. Red. (%)*	*DER*	*(%)*	*CBOR*	*(%)*
1	2 538	2 466	2.8	2 523	0.6	2 470	-0.2
10	2 766	2 556	7.6	2 682	3.0	2 555	0.0
100	5 017	3 457	31.1	3 579	28.7	3 424	1.0
1 000	27 512	12 458	54.7	12 206	55.6	11 384	8.6
10 000	252 471	102 458	59.4	97 239	61.5	90 335	11.8
20 000	502 419	202 458	59.7	190 554	62.1	177 889	12.1
30 000	752 388	302 458	59.8	283 537	62.3	265 508	12.2

The primary contributors are consistent with the previous case: efficient time encoding and flattened structure. The per-entry overhead remains the dominant factor when scaled to a large number of revocations. The size of PQ signatures is confined to the final signature block and is negligible relative to the size of the list. As a result, DER-encoded CRLs continue to exhibit high compressibility, achieving over 62% reduction at larger scales. In contrast, CBOR-encoded CRLs exhibit a modest relative compression rate, reaching up to 12.2% for 30,000 revocations.

In summary, even when paired with PQ algorithms, CBOR-encoded CRLs retain their advantages over DER-encoded equivalents. The benefits scale with the number of revoked entries, making them a potential enabler for revocation management in constrained environments.

6.1.3. Additional Considerations

CBOR-encoded profiles are not limited to certificates and CRLs, extending to nearly every DER-based PKI object. In certificate validation chains, for example, the space savings offered by C509 accumulate linearly. For instance, each extra intermediate or cross-signed certificate removes roughly 180 bytes, so a typical three-hop chain avoids more than 700 bytes of transmission overhead. Although that amount is modest, eliminating avoidable bytes is always preferable, especially on constrained links.

Additionally, auxiliary PKI messages show a similar pattern. CSRs mirror certificates in structure—subject, public key, signature, and extensions—so they also benefit from the identical CBOR-based reductions. OCSP responses, by contrast, are already compact, consisting mainly of status codes, timestamps, and a handful of extensions. While CBOR still trims some bytes, the proportional gain is far less pronounced.

Conclusion C509 demonstrates clear advantages in reducing the message overhead of PKI objects. Its value lies not only in reducing the size of certificates and revocation lists, but also in establishing a more compact and consistent encoding profile that generalises across the PKI stack. While the gains are more modest for PQ certificates, C509/CBOR has a significant impact on revocation lists, regardless of the cryptography used. Ultimately, the adoption of C509 should be assessed not only in terms of size reductions but also in regard to the complexity of its implementation.

6.2. Implementation Complexity

Key findings Transitioning from **X.509/DER** to the streamlined **C509/CBOR** profile can significantly reduce implementation complexity—and with it, the area for potential implementation vulnerabilities.

Headline Results

> **Reduced codebase footprint:** C509 implementations exhibit an average **79–80% reduction in logical lines of code** compared to X.509, easing storage, testability, and maintenance.

> **Reduced cyclomatic complexity (CCN):** The CCN is reduced by **2–3**× for C509 compared to X.509, indicating simpler control-flow and testability.

> **Lower cognitive complexity:** Halstead Volume drops by **63–74%** in for C509 relative to X.509, signalling a lower information density. Median Halstead Difficulty falls from **≈ 26–29** in the X.509 code to **≈ 7–9** in the C509 variants, indicating a tangible reduction in cognitive effort.

> **Profile-driven efficiency:** Auto-generated C509 implementations (e.g., `zcbor`) demonstrate lower complexity than generated X.509 counterparts, confirming C509's benefits stem from its structural characteristics rather than implementation style.

> **Cross-tool consistency:** Two static analysis tools corroborate similar metric trends, reinforcing the validity of the findings.

6.2.1. Experimental Setup

As outlined in the previous chapters, the primary source of implementation complexity is the parsing logic associated with X.509 certificates. A direct comparison with the corresponding logic for C509 is therefore appropriate to highlight the differences in implementation. This empirical comparison is supported through the use of static-analysis tools that apply heuristic metrics to quantify code complexity.

Corpus

The selected implementations were verified to support equivalent functionality, specifically the ability to **parse (& serialise) certificates and revocation lists under either the X.509 or C509** profile. Although additional libraries—such as OpenSSL or CycloneCRYPTO from Oryx Embedded—also implement X.509 parsing, their broader scope and deeper integration make it challenging to isolate components that precisely correspond to the targeted functionality. As a result, their inclusion could potentially undermine the consistency of the comparison. For this reason, the analysis is restricted to the implementations listed in Table 6.8. Still, due to their divergent nature, the implementations were grouped into distinct experimental settings to enable direct comparison, presented in Table 6.9.

Table 6.8: Analysed parsers: implementation language, target profile, and brief description.

Implementation	Lang.	Profile	Description
`x509-parser`[2]	C	X.509	ANSSI, open-source, custom, runtime-error-free parser-only implementation (no serialisation) including the DER decoding layer and no external dependencies, formally verified using Frama-C and ACSL annotation comments that do not affect the analysis [9].
`ASN1C (generated)`[3]	C	X.509	ASN.1 schema [6] generated parser and serialiser using the commercial Objective Systems `asn1c` compiler generating industry-grade code for BER/DER with the encoding layer delivered as a pre-compiled library.
`c509-native`[4]	(C-like) C++	C509	The custom parser and serialiser proposed in this work, relying on `zcbor` for the encoding layer, optimised for embedded systems (Chapter 5).
`zcbor (generated)`[5]	C	C509	CDDL schema [8] generated parser using the open-source `zcbor` generator producing low-footprint C encoders/decoders for CBOR.

c509conv[6]	C	C509	Open-source *re-encoder* relying on external libraries for the DER/CBOR encoding layers.
CBOR-certificates[7]	Rust	C509	IETF reference *re-encoder* [8], relying on external libraries for the DER/CBOR encoding layers.

Table 6.9: Experimental settings and selected implementations

Experiment Setting	Requirements	Implementations
1	Certificate and CRL parser-only implementations including the DER/CBOR encoding layer	x509-parser c509-native zcbor (generated)
2	Certificate and CRL parser-only implementations excluding the DER/CBOR encoding layer	ASN1C (generated) c509-native zcbor (generated)
3	Certificate and CRL parser and serializer implementations excluding the DER/CBOR encoding layer	ASN1C (generated) c509-native zcbor (generated)
4	Re-encoding tools between C509 and X.509 excluding the DER/CBOR encoding layer	c509conv CBOR-certificates

Remarks For each setting, c509-native and zcbor (generated) selectively enabled or disabled the encoding layer or serialisation logic to match the specific profile. Code-generated implementations serve as a proxy to reduce developer-induced variability, allowing for a more objective comparison of the inherent complexity of each setting. Re-encoders were included for informative purposes, as they are not comparable. Lastly, the survey revealed limited availability of C509 implementations.

Tooling

To support metric-based comparison and increase confidence in the results, two independent static analysis tools were used to extract code complexity metrics across the selected corpus:

- **Tool 1**: Jarod42/ccccc[8] (C and C++ Code Counter and Checker) is a static analysis tool specialised for C and C++ codebases. The tool computes metrics such as logical lines of code (LLOC), cyclomatic complexity (CCN), Halstead volume and difficulty, and function count, leveraging C/C++-specific parsing for greater precision.
- **Tool 2**: mozilla/rust-code-analysis[9] [70] is a static analyser supporting multiple languages, including C, C++, Rust, Java, and Python. It enables similar metrics, and its design and language coverage make it suited for high-level metric extraction and cross-language comparison.

While the tools may yield differing absolute values due to implementation details, it is the consistency of observed trends across both analysers that strengthens the reliability of the analysis.

Metrication

The used metrics were previously introduced in Section 2.5. As these metrics are computed at the function level, aggregation is required to characterise each implementation as a whole. Values are

[2] https://github.com/ANSSI-FR/x509-parser
[3] https://obj-sys.com/products/asn1c/index.php
[4] https://github.com/rosualinpetru/c509-native
[5] https://github.com/NordicSemiconductor/zcbor
[6] https://github.com/fabian18/c509conv
[7] https://github.com/cose-wg/CBOR-certificates
[8] https://github.com/Jarod42/ccccc
[9] https://github.com/mozilla/rust-code-analysis

aggregated using the sum, mean, or median, depending on the nature of each metric. This enables high-level comparisons and trend analysis across implementations.

It must be evidenced that C509 represents a purposefully restricted subset of X.509, retaining only essential features. Combined with its simplified encoding, C509 requires a less complex implementation. CBOR defines only eight major self-describing types, similar to JSON [18]. In contrast, ASN.1 DER involves four tag classes, over 25 universal types, and schema-dependent tagging and parsing [25]. From the outset, C509 is expected to be simpler to implement—a claim also made in the draft specification [8]. Instead, this quantitative analysis aims to emphasise the extent to which the expected simplification is reflected in practice.

6.2.2. Results

Each setting is analysed individually, offering insights into different perspectives of software complexity.

Setting 1: Parser

Regarding **LLOC**, as `c509-native` internally relies on `zcbor`, both C509 implementations exhibit comparable code sizes and remain significantly smaller than `x509-parser`. **Tool 1** shows reductions of $\approx 75.8\,\%$, while **Tool 2** records savings of $\approx 81.9\,\%$ and $\approx 84.8\,\%$ respectively. On average, this corresponds to an ≈ 79–$80\,\%$ decrease, roughly 4–$6\times$ less code to read, review and maintain for C509 implementations (here `c509-native`) when compared to X.509 (here `x509-parser`).

In terms of **CCN**, `C509-native` consistently reports the lowest complexity. With **Tool 1**, its Total CCN drops by a factor of $\approx 3\times$ relative to `x509-parser`, while `zcbor (generated)` achieves a $\approx 2.3\times$ reduction. **Tool 2** mirrors this trend with factors of $\approx 2.9\times$ and $\approx 2.0\times$, respectively. On average, `C509-native` realises a $\approx 3\times$ decrease in cyclomatic complexity, whereas `zcbor (generated)` settles around $\approx 2\times$—still substantial, yet leaving measurable room for optimisation. These cuts, coupled with the LLOC savings, indicate that C509 parsers offer smaller code bases with greatly simplified control-flow, improving both verifiability and testability.

Concerning **Halstead metrics**, both C509 implementations demonstrate significantly reduced token-level complexity. In terms of Volume, `C509-native` achieves a ≈ 73–$74\,\%$ reduction, while `zcbor (generated)` lands in the ≈ 63–$67\,\%$ range across the two tools—amounting to a ≈ 3–$4\times$ contraction in token space. Median Difficulty scores fall by roughly $70\,\%$, and the 95-percentile drops by ≈ 50–$60\,\%$ for both variants, signalling a substantial reduction in the variety and frequency of operators and operands that must be tracked during review. Together, these results underscore that C509 parsers not only reduce code size but also enhance conceptual clarity.

While the generated `zcbor (generated)` already offers low complexity, it includes structural overhead inherent to code generation. A handwritten implementation, such as `C509-native`, can streamline control flow and eliminate redundancy, yielding better metrics. This demonstrates that domain-specific optimisation provides benefits beyond what automated tools can achieve.

Table 6.10: Setting 1: X.509 and CRL parser-only implementations including the binary encoding layer.

Parser-only (Tool 1) ccccc	**Total LLOC**	**Mean CCN**	**Total CCN**	**Total Volume**	**Median Diff**	**Q-95 Diff**	**Func Cnt**
`x509-parser`	8 019	7.80	1 622	243 775.46	28.67	62.28	208
`c509-native`	1 939	2.99	535	66 309.72	7.88	25.14	179
`zcbor (generated)`	1 937	3.99	707	89 235.18	9.21	24.54	177

Parser-only (Tool 2) rust-code-analysis	**Total LLOC**	**Mean CCN**	**Total CCN**	**Total Volume**	**Median Diff**	**Q-95 Diff**	**Func Cnt**
`x509-parser`	7 432	7.69	1 630	214 955.33	26.39	56.57	212
`c509-native`	1 346	3.34	565	55 192.19	7.50	25.23	169
`zcbor (generated)`	1 132	4.35	805	70 315.48	7.70	21.97	185

Setting 2: Parser

This experiment isolates the complexity introduced purely by the feature sets of each certificate format. As the considered parsers are schema-generated—`ASN1C (generated)` from the full X.509 ASN.1 modules, and `zcbor (generated)` from complete CDDL descriptions of C509, human optimisation plays no role in the results. Moreover, by omitting the binary-encoding layers, the codebase contains only feature-level parsing logic. Under these conditions, the relative complexity of C509 is evident: both C509 parsers show order-of-magnitude reductions across total value metrics when contrasted with `ASN1C (generated)`. Nevertheless, the relative complexity of DER compared to CBOR would lead to even greater gaps in metric value.

Some metric inconsistencies emerge between `Tool 1` and `Tool 2` for the C509 parsers. A key reason is that Experiment 1 included `zcbor` as the encoding layer component, whereas Experiment 2 does not, inflating the former's baseline. In addition, `C509-native` relies on macros; `Tool 1` expands these during preprocessing and counts the resulting helper functions, while `Tool 2` does not. Roughly twenty trivial utility functions are therefore missing from `Tool 2`'s reports. Combined with the internal differences in LLOC calculation, these factors alter the metric values for small codebases. Nevertheless, these inaccuracies do not affect the substantive conclusion that C509 parsers remain significantly smaller and less complex than their X.509 counterparts.

Table 6.11: Setting 2: X.509 and CRL parser-only implementations excluding the binary encoding layer.

Parser-only (Tool 1) ccccc	**Total LLOC**	**Mean CCN**	**Total CCN**	**Total Volume**	**Median Diff**	**Q-95 Diff**	**Func Cnt**
ASN1C (generated)	4 611	6.73	1 090	181 210.87	12.23	52.73	162
c509-native	540	4.7	207	24 784.94	12.17	24.93	44
zcbor (generated)	538	9.02	379	47 710.39	12.29	16.36	42

Parser-only (Tool 2) rust-code-analysis	**Total LLOC**	**Mean CCN**	**Total CCN**	**Total Volume**	**Median Diff**	**Q-95 Diff**	**Func Cnt**
ASN1C (generated)	4 421	6.73	1 090	161 474.18	10.80	47.62	162
c509-native	457	7.27	189	20 333.37	12.46	30.60	26
zcbor (generated)	243	10.21	429	35 456.66	10.47	14.31	42

Setting 3: Parser & Serialiser

Extending the feature set to include serialisation preserves the overall ranking of implementations while exposing differences in how each scales. Although all projects see increases in code size and complexity, the growth is more pronounced in `ASN1C (generated)`, which widens the absolute gap in **LLOC** and total **CCN**. The C509 implementations almost double their **LLOC** yet remain significantly smaller and less complex overall than `ASN1C (generated)`. Halstead Volume grows across all implementations, but disproportionately: `ASN1C (generated)` expands by about 50% the volume of the parser-only implementation, while both C509 variants increase by 85–95%. Despite this, the `c509-native` parser–serializer maintains a 3–5$\times$ smaller token footprint compared to its X.509 counterpart. The Median Difficulty reflects stable complexity in all implementations, remaining nearly flat, whereas the 95-percentile indicates that some functions of `C509-native` may warrant targeted refactoring. These trends suggest that although substantial logic is introduced, its difficulty remains low. Minor inconsistencies between analysers persist, stemming from macro expansion and LLOC heuristics, without affecting the broader conclusion: C509's complexity scales better with expanded functionality.

Setting 4: Re-encoder

This experiment reports complexity metrics for two publicly available re-encoders for informative purposes. The projects are prototypes written in different languages—`cbor-certificates`, the IETF reference re-encoder, in Rust and `c509conv` in C—so direct comparison is not meaningful. Nonetheless, the absolute figures for **LLOC**, **Total CCN**, and **Halstead Volume** of `c509conv` sit well below those of the `ASN1C (generated)` in Setting 3, even though a re-encoder must, in principle, parse and re-encode both DER and CBOR. This apparent contradiction is explained by the *restricted domain* of re-encoding:

Table 6.12: Setting 3: X.509 and CRL parser and serializer implementations excluding the binary encoding layer.

Parser-only (Tool 1) ccccc	**Total LLOC**	**Mean CCN**	**Total CCN**	**Total Volume**	**Median Diff**	**Q-95 Diff**	**Func Cnt**
ASN1C (generated)	6 991	7.09	1 752	277 721.04	19.72	48.92	247
c509-native	966	5.98	389	48 177.98	12.89	40.01	65
zcbor (generated)	1041	7.7	647	89 113.7	11.84	16.36	84

Parser-only (Tool 2) rust-code-analysis	**Total LLOC**	**Mean CCN**	**Total CCN**	**Total Volume**	**Median Diff**	**Q-95 Diff**	**Func Cnt**
ASN1C (generated)	6 389	7.09	1 752	247 644.75	17.60	45.33	247
c509-native	796	7.89	371	39 326.12	13.53	34.29	47
zcbor (generated)	441	8.94	751	65 836.81	10.45	14.80	84

only C509-conformant X.509 certificates are accepted, so the DER side needs only to implement the subset of structures admissible under the C509 profile and can abort early when unsupported features are detected. Thus, the X.509 parsing component is simpler than in a general-purpose library. While the small sample precludes generalisations about implementation effort, the limited feature set of their target format suggests re-encoders might be no more complex, or even less so, than full X.509 tooling.

Table 6.13: Setting 4: Re-encoding tools between C509 and X.509 excluding the DER/CBOR encoding layer.

Re-encoder ccccc	**Total LLOC**	**Mean CCN**	**Total CCN**	**Total Volume**	**Median Diff**	**Q-95 Diff**	**Func Cnt**
c509conv	3 778	5.78	930	154 518.71	17.07	51.47	161

Re-encoder rust-code-analysis	**Total LLOC**	**Mean CCN**	**Total CCN**	**Total Volume**	**Median Diff**	**Q-95 Diff**	**Func Cnt**
c509conv (C)	2 584	6.19	928	129 091.91	17.61	47.51	150
CBOR-certificates (Rust)	1 359	7.69	938	116 099.58	12.50	44.42	122

6.2.3. Additional Considerations

The differences in the design of DER and CBOR raise some implementation challenges in the case of the latter. In DER, each data item includes its exact byte length, enabling efficient structure traversal and data access —a crucial property, especially considering large PQ certificates. This facilitates optimised signature verification without the need to parse or traverse the entire certificate. An implementation can read the length from the TLV root of the signed content and skip it to access the signature directly.

In contrast, CBOR encodes the number of contained elements rather than the total byte size. For example, a CBOR array specifies the number of items it holds, not the bytes required to encode them. In the case of byte strings or text, it is contextually correct that the length corresponds directly to the byte count. As a result, verifying CBOR-encoded certificates requires bounded incremental parsing over the 10 preceding items to locate the signature value. While CBOR lacks the efficient structural traversal and access convenience of DER, parsing its structure remains efficient and easy to implement, especially given that the types and ordering of fields are flattened and well-defined by the C509 profile.

Conclusion This empirical analysis quantitatively demonstrates the tangible benefits of the C509 profile compared to X.509. Across all metrics and experimental settings, C509 implementations consistently yield smaller, simpler, and more testable codebases, reducing both the attack surface and development effort. Despite minor inaccuracies in metric values due to tool limitations and implementation style, the consistency of trends across tools and settings confirms the robustness of the findings. Ultimately, the study does not claim absolute precision, but instead aims to offer a reliable and intuitive understanding of the gains achievable through the adoption of C509 in terms of software complexity.

6.3. Space Considerations

Despite its benefits, further aspects must be assessed before deeming C509 suitable for space.

6.3.1. C509 Deployment

The implications of deploying C509 in space vary depending on the variant used.

Natively Signed Certificates

Assuming C509 adoption, mission designers must choose between natively signed and re-encoded certificates. Natively signed certificates apply signatures directly over CBOR-encoded data, eliminating ASN.1/DER dependencies. This approach suits closed ecosystems but lacks practicality in federated environments that rely on interoperability with standard X.509-based PKIs.

Re-encoded Certificates

In contrast, re-encoded certificates enable transformations to achieve compatibility with traditional X.509-based PKIs. Rather than signing the CBOR-encoded certificate directly, the CBOR format serves as a streamlined representation of the same data structure originally signed in DER. This enables interoperability with existing X.509-based PKIs, while providing the benefits of C509.

The C509 draft does not prescribe where or by whom re-encoding must occur, allowing system designers to align placement with operational, trust, and resource constraints. A typical architecture offloads re-encoding to a border gateway, such as an element in the ground segment, which acts as an intermediary between constrained environment (space elements) and the broader ground network. This enables the constrained device to only pay the light penalty of only parsing compact CBOR representations, instead of the costly DER parsing. The gateway handles conversion between X.509 and C509, and vice versa, and consequently pays the heavier cost of DER parsing. In other words, the heavy cost of heavy parsing shifts from a resource-constrained spacecraft to a ground gateway.

Under such deployments, beyond reducing transmission size, re-encoded C509 can eliminate the need for X.509/DER parsing, and more importantly. As shown in Figure 6.1, the gateway follows the steps indicated by the blue arrows. First, it parses the X.509 certificate—typically the most complex and resource-intensive step—and extracts the Abstract Data Type (ADT). The ADT is then serialised into CBOR and sent to the constrained client.

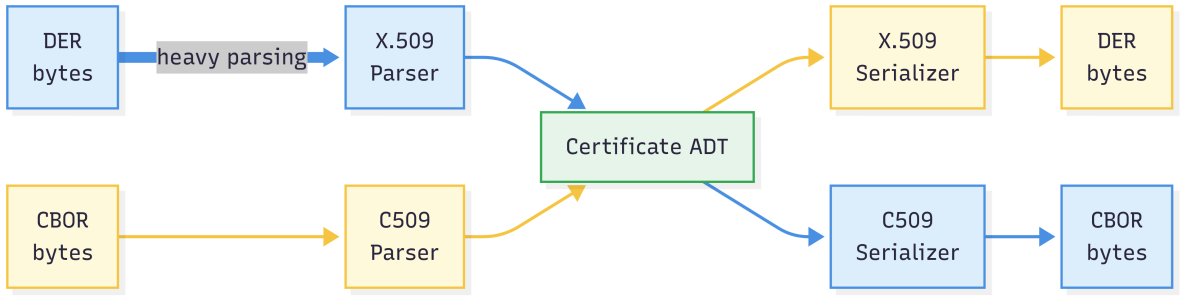


Figure 6.1: X.509 vs. C509 Parsing and Serialising.

The constrained device, in turn, follows the steps illustrated by the yellow arrows. Upon reception of the CBOR-encoded certificate, it first performs lightweight CBOR parsing. While signature verification still requires DER serialisation, this operation is more straightforward and more deterministic than DER parsing, as it can leverage the ADT's structured context. Depending on peer requirements, the device can serialise certificates in CBOR or DER as needed. What is essential is that the constrained device never needs to perform heavy DER parsing.

It is worth noting that not all X.509 certificates are convertible to C509, as C509 represents a constrained subset of X. 509 certificates. However, this can be addressed by enforcing a restricted X.509 profile to ensure re-encoding remains possible. For instance, the considerations from Chapter 4 enable convertible certificates.

6.3.2. C509 Disadvantages

While C509 offers clear benefits in bandwidth and implementation simplicity, its adoption is limited by ecosystem immaturity. Full integration requires a CBOR-based infrastructure, which is currently underdeveloped. As C509 is still an IETF draft, and CBOR-encoded PQ certificates and revocation formats (e.g., CRLs, OCSP) are not yet standardised, the lack of normative completeness hinders industry uptake and limits suitability for space-standardisation efforts.

Another factor deterring the C509's adoption of space is the lack of sufficient benefits that justify deviation from the established X.509 standard. While C509 offers substantial reductions in CRL size, the resulting compressed CRLs may still exceed practical transmission thresholds. More critically, the predominant bandwidth challenge stems from the increased size of PQ certificates—an issue C509 cannot inherently mitigate. As such, the overall bandwidth savings might be insufficient to warrant adoption.

Additionally, for hardware implementations, reduced software complexity may be less relevant. Neither X.509 nor C509 can fully alleviate the challenges associated with hardware-based parsing, sometimes necessary in constrained space applications. Variable-length bit strings and integers introduce significant complexity. To address this, systems tend to enforce minimal and rigid certificate profiles that enable fixed-length representations—typically through padding—facilitating offset-based parsing. Under these constraints, the complexity of parsing is effectively abstracted away, and the choice between X.509 and C509 becomes largely inconsequential from an implementation standpoint. From this perspective, a simplified software implementation might not justify challenging system-wide compatibility.

Conclusion This work has demonstrated that C509 can provide significant benefits in terms of bandwidth reduction and implementation simplification. In particular, smaller revocation lists—enabled by compact time encodings—may enhance scalability in constrained environments. However, the early maturity of the C509 specification is responsible for the limited industry adoption, introducing practical limitations. Notably, the lack of standardised revocation mechanisms for CBOR-encoded certificates poses an adoption barrier. Additionally, maintaining compatibility with X.509 infrastructures adds architectural complexity, as re-encoding between profiles must be carefully managed and deployed.

Rather than prescribing a universal solution, the findings suggest that C509 is a viable alternative to X.509 in appropriate contexts. Mission-specific needs, including interoperability and resource constraints, should guide profile selection. While federation-wide interoperability favours X.509, specialised missions should adopt the profile that best aligns with their operational priorities.

7

Discussion

The design of minimal, interoperable certificate profiles for space is not only a technical challenge but also an adoption and standardisation problem. This chapter examines how the proposed solutions intersect with industrial adoption, the challenges of standardisation, and the gaps between terrestrial X.509 practices and space system needs. It also outlines the study's limitations, highlighting unresolved technical and operational aspects to guide future work.

The preceding chapters examined certificate design from several independent angles. Taken in isolation, each issue may appear manageable; the real challenge emerges when they converge into a single profile. Designers must simultaneously navigate (i) cryptographic algorithm choices (classical, PQ, or hybrid), (ii) data models and on-wire encodings (DER/X.509 vs. CBOR/C509), and (iii) semantic constraints in extensions and validation policies. Any misalignment across these dimensions risks breakdowns in interoperability. In this regard, this study suggests guidelines across these aspects to support the development of a standardised profile for federated space PKIs.

Yet, defining such a profile is far from trivial and cannot be solved by a single actor. Because this work intervenes directly at the level of certificate standards, its impact ultimately depends on broad, multi-stakeholder acceptance. Formalisation, security proofs, reference implementations, and adoption timelines extend beyond the influence of any individual effort. For example, while C509 shows promise in constrained environments, its early-stage maturity and limited industry uptake currently limit its viability for large-scale deployment until further experience and standardisation progress are achieved.

Nevertheless, action is needed to establish more precise, standardised requirements for space systems. Field discussions with industry stakeholders highlighted the challenges of operating under today's general and sometimes conflicting recommendations. For example, the CCSDS Authentication Credentials [7] mandates GeneralizedTime before 2050 while referencing the X.509 Internet profile, which strictly mandates UTCTime for the same period. Similarly, IGCA lists required features—such as policy mappings and subject alternative names—without specifying clear certificate types. Balancing flexibility for mission-specific configurations with sufficient precision to ensure interoperability remains a trade-off; however, further refinements could improve consistency across space systems.

Current efforts in both standardisation and research primarily focus on patching X.509 to meet space system constraints. This includes pruning extensions, compressing chains, or tunnelling revocation checks, all to preserve compatibility with the terrestrial X.509 standard. However, this pursuit of compatibility is often more apparent than real. In practice, many actors adopt X.509 as the default or "safe" choice. Once the implementation complexity becomes evident, they attempt to mitigate it by subsequently defining and implementing such highly restricted profiles that their systems can no longer process broader X.509 certificates from the wider ecosystem. At that point, while some degree of formal compatibility with X.509 remains, it is often too limited to justify the adoption of X.509, disregarding the potential benefits of adopting alternative approaches, such as C509, or even designing a dedicated, space-native profile. In short, clinging to X.509 offers diminishing returns *in some cases*, and rethinking the underlying approach may yield greater gains.

Limitations

The initial research phase considered conducting a broad survey of space-mission classes to establish domain context. However, later discoveries identified a detailed taxonomy of 29 missions grouped into nine archetypes, rendering additional surveys unnecessary [71]. The analysis of cryptographic certificate usage is too specific to address at a generic mission level. Thus, the study redirected its focus to IGCA, a concrete space PKI application, enabling evaluation without speculative assumptions.

Similarly, the initial plan to benchmark PQ algorithms was reconsidered after identifying the EBACS benchmarks [72] and ongoing work which already provide evaluations on emulated space-grade hardware. As a result, the study focused instead on reviewing (to be) standardised PQ algorithms, using them to illustrate how different families may suit different certificate types depending on their security and computational requirements. Regarding formats, this work's proposal of Hybrid Coupled remains preliminary, offering a conceptual approach that still requires formalisation and validation to assess its practicality and potential for adoption in the future. Additionally, the fast-paced development of PQ formats continues to introduce new methods. A recent proposal based on the use of alternative extensions is currently under development [73], but was identified at a late stage in the research and remains outside of scope.

Regarding the federated profiles, the design was based on the requirements set by existing standards, with CRL support being a central defining element. If a federated space PKI were to adopt an alternative revocation mechanism, the resulting profiles would likely differ. Nevertheless, the key contribution lies in proposing a structured approach to profile design aimed at improving interoperability, while recognising that mission designers and standardisation bodies must determine the exact content. Defining a baseline set of profiles was still necessary in this study to enable meaningful comparisons with C509.

Also, the current study assumed a bridge PKI model, following the IGCA approach, and thus based its proposals on the premise of centralised policy management. Had IGCA adopted a different model, such as a mesh architecture or extended trust lists, the discussion and resulting recommendations would have differed. Nevertheless, this raises an important research question: whether the Bridge CA remains the optimal architecture for meeting the needs of a federated space PKI, or if alternative trust models may offer better scalability, flexibility, or resilience.

`c509-native` is a prototype, currently limited to experimental use. It relies on complete in-memory processing of CRLs, which is impractical for real-world applications. Integration with other libraries revealed challenges, with several traditional and PQ algorithms not yet correctly identified or supported. Additionally, as the C509 draft has continued to evolve, certain features appeared after the tool's implementation and were not incorporated. Despite these limitations, `c509-native` played an instrumental role in enabling the analyses presented in this study.

The trade-off analysis between X.509 and C509 in this study focused primarily on object size and software complexity. However, numerous additional factors influence this decision, including hardware implementation aspects and the potential for formal verification. In particular, recent research on formally verified, schema-driven parser generation [74, 75] offers promising mitigation strategies for implementation complexity and would have been highly relevant to the software complexity assessment. Due to the limited availability of isolated, open-source implementations, the analysis relied on a narrow corpus. These formal verification approaches were identified only in the later stages of the research and represent a complementary direction for future work.

While a broader range of metrics—particularly from the Halstead family—could have been included, the study limits itself to those summarised. These core measures are considered sufficient to assess complexity in the given context, as most extended Halstead indicators are derived from these exact base metrics. It must be acknowledged that Halstead metrics may not accurately reflect reality for code-generated implementations; yet, they can serve as a good comparative indicator. Finally, more accurate results would require access to commercial analysers.

8

Related Work

Establishing a federated space PKI has been a long-standing effort among researchers, alongside ongoing work to strengthen the security of X.509 certificate profiles and adapt them for use in constrained environments and post-quantum migration. This chapter outlines the related work across these areas, including proposals for federated trust architectures, compact and secure certificate profiles, and the integration of quantum-resistant algorithms into standard certificate structures.

Federated Space Public Key Infrastructure

Koisser et al. present TruSat, a decentralised extension to traditional PKI for federated spacecraft networks [76]. Central to the system is the CommonCA, a collaborative root CA abstraction jointly operated by multiple independent CAs. The CAs composing the CommonCA use the Practical Byzantine Fault Tolerance (PBFT) consensus protocol to coordinate the issuance and revocation of certificates. TruSat employs signature aggregation to reduce communication overhead, supporting both Schnorr and BLS schemes. The latter is preferred because it can compress multiple signatures into a single, constant-size proof. For revocation, TruSat accommodates both universal approaches (e.g., CRLite and Let's Revoke) and prover-based mechanisms (e.g., Merkle trees and OCSP stapling). Experimental validation on ESA's OPS-SAT CubeSat demonstrates the feasibility of the architecture, showing low runtime overhead and practical in-orbit performance. The authors argue that the IGCA model proposed by CCSDS merely relocates the trust root without addressing the governance and interoperability challenges of federated missions. In contrast, TruSat shows that verifiable consensus among authorities can provide scalable trust without centralisation.

In another work, Koisser et al. propose V'CER, a certificate validation framework for constrained networks such as satellite constellations and IoT systems [77]. The authors address the timely revocation of certificates by introducing a system that utilises Sparse Merkle Trees (SMTs) and a novel structure known as the Validation Forest (VF). V'CER enables the lightweight, epidemic-style dissemination of revocation information and decentralised repair of outdated proofs of inclusion (PoIs) between devices, without requiring real-time access to a CA. Their implementation, tested on ESA's in-orbit OPS-SAT satellite and through large-scale simulations, demonstrates that V'CER reduces reliance on centralised authorities and requires less than 3KB of storage per device to manage up to 1 million certificates. The authors conclude that V'CER offers a scalable, efficient, and secure revocation mechanism suitable for resource-constrained environments, outperforming traditional schemes like CRLs and OCSP.

In a subsequent research, Koisser et al. conduct a comprehensive systematisation of knowledge on public key infrastructure and location privacy within satellite networks [78]. Regarding PKI, the authors highlight issues such as orbital delay, intermittent connectivity, and limited bandwidth, which complicate time-sensitive operations like certificate revocation. They argue that traditional PKI models assume stable, high-availability infrastructure and may not be directly transferable to space systems without modification. The paper synthesises research on certificate distribution, revocation, and key management in the space domain, contrasting it with terrestrial assumptions. The authors also ref-

erence existing efforts, such as NASA's Artemis program, as examples of evolving space networking architectures. Their analysis reinforces the need for federated trust models that account for delayed revocation, inter-organisational trust, and resilience in disconnected conditions.

Smailes et al. introduce Keyspace, a simulation framework for PKI architectures in large-scale, interplanetary satellite networks [79]. The authors argue that the predictability of satellite orbits allows for deterministic routing and synchronisation, which opens the door to adapting terrestrial PKI systems. To test this hypothesis, they developed the Deep Space Network Simulator (DSNS), a tool that models federated satellite systems under various topologies, including Earth-Moon-Mars communication. The authors compare traditional PKI configurations, such as CRLs and OCSP, and propose OCSP Hybrid and relay firewalls, which enhance revocation propagation and limit the reach of attackers. Their results demonstrate that federated topologies can establish low-latency connections and effectively revoke keys, particularly when supported by caching and strategic relay positioning. Finally, DSNS provides a robust evaluation framework, and the results demonstrate that federated PKI architectures, when carefully tuned, remain viable for scalable and secure communications in space.

Post-Quantum Certificates

Kampanakis et al. present one of the early works addressing post-quantum certificates [80]. The authors examine the viability of integrating signature algorithms into X.509 certificates within widely deployed terrestrial protocols such as TLS, DTLS, IKEv2, and QUIC. Their research investigates the practical implications of larger post-quantum public keys and signatures, with a primary focus on transmission overhead. Through experimental evaluations, the study demonstrates that despite significant increases in certificate size, protocol mechanisms such as fragmentation, segmentation, and multiplexing effectively mitigate performance impacts. Ultimately, the authors conclude that while the overhead introduced by post-quantum certificates is not negligible, it is acceptable in most modern terrestrial networks. Nevertheless, this may not hold in constrained scenarios, such as space.

In another study, Raavi et al. conduct a performance characterisation of digital certificates using Dilithium, Falcon, and Rainbow [81]. Their study empirically assesses the impact of these algorithms on key-pair generation, certificate signing request generation, certificate generation, and certificate verification. The authors confirm the feasibility of integrating post-quantum algorithms into the X.509 standard. They note significant variations in computational and storage overhead, depending on the specific algorithm used. Conclusively, their results indicate that practical deployments must strike a balance between security and computational and storage constraints.

Ricchizzi et al. tackle the practical integration of PQC into X.509-based identity management for industrial systems [82]. After surveying existing toolchains, they identify a conspicuous gap: there is no publicly available CLI that supports the creation and validation of hybrid or composite certificates. To close this gap, the authors develop an open-source proof-of-concept, pqcli, built on Bouncy Castle. The tool can generate traditional, hybrid ("Catalyst"), composite, and partially chameleon certificates using PQC algorithms while remaining compatible with standard X.509 workflows. A comparison with OpenSSL (with and without the oqs-provider) highlights current limitations in standardisation, algorithm coverage, and validation logic, underscoring the need for further tool support and specification work before widespread PQC migration in the industry.

The Open Specification for Pretty Good Privacy IETF working group (OpenPGP) lists concrete examples of hybridised algorithm combinations for their protocol [83, 64]. OpenPGP is a standard for encrypting and signing data that uses a decentralised trust model and a packet-based certificate format. Their current draft defines composite public-key encryption based on ML-KEM, composite signatures based on ML-DSA, both in combination with ECC, and SLH-DSA as a standalone signature scheme. While different from X.509, this reference on OpenPGP certificates demonstrates the migration of mature protocols through hybridised algorithms.

Wang et al. provide a qualitative analysis of four approaches to integrating quantum-safe algorithms into X.509 certificates [62]. Each method is discussed in terms of compatibility, migration strategy, and structural implications. For example, quantum-safe certificates introduce minimal structural changes but require all systems to fully adopt PQC simultaneously. In contrast, hybrid and composite certificates combine traditional and post-quantum elements into a single certificate, allowing for phased migration

or enhanced cryptographic strength. Parallel certificate chains, which issue multiple certificates with different algorithms for the same identity, offer more flexibility but double the amount of data transferred. While the paper offers valuable insights into the design trade-offs and potential use cases of each format, it does not provide quantitative comparisons using metrics such as message size.

Certificate Profile Design and Secure Implementation

Prior work by Forsby et al. has proposed a CBOR-based profile for IoT devices, named XIOT, focusing on space savings by fixing the cryptographic algorithms within the profile and eliminating non-essential fields [84]. XIOT significantly reduces certificate payloads while preserving compatibility with existing PKI implementations. Evaluated on Contiki-based sensor nodes, XIOT significantly lowers handshake traffic and energy use, demonstrating that certificate-based security becomes practical for large fleets of constrained IoT devices through the use of CBOR encoding. XIOT appears as an early version of re-encoded C509; however, it cannot be established whether the profiles are related.

Ebalard et al. demonstrate that a carefully crafted “meet-in-the-middle’’ development process can bring mainstream verification tools within reach of complex C code: they re-implement an X.509 parser in idiomatic C99 and then use Frama-C to prove the absence of run-time errors (RTEs) [9]. Their experience report demonstrates how incremental ACSL annotation and plugin-assisted analysis enable them to discharge all RTE warnings, resulting in the first X.509 parser with machine-checked memory-safety guarantees in C, albeit without addressing higher-level semantic checks such as validation.

Barenghi et al. take a complementary approach, directly addressing ambiguous syntax in the X.509 specification. They formalise the whole grammar, generate a parser with proven termination, and feed it 11 million certificates covering the public IPv4 space [10]. Their large-scale measurement reveals that 21.5% of real-world certificates violate the grammar, and yet seven popular TLS stacks still accept hundreds of thousands of them, enabling practical impersonation attacks. The work thus positions precise, unambiguous parsing as a prerequisite for secure validation.

While prior studies often rely on fuzzed inputs, Tatschner et al. shift the focus to in-the-wild certificates. ParsEval exercises six widely used parsing libraries on 186 million certificates and analyses their error codes [13]. The study uncovers systematic divergences — most notably an anomaly in wolfSSL’s parser — showing that parser behaviour still diverges significantly in production software and that inconsistencies are observable by end users, not just by attackers with crafted inputs.

Moving beyond parsing to end-to-end trust decisions, Debnath et al. introduce ARMOR, the first formally verified implementation of the X.509 Certificate Chain Validation Logic (CCVL) [85]. By decomposing the CCVL into separately verified modules (parsing, chain building, semantic checks, and canonicalisation), they expose latent non-compliances in existing libraries; however, the design currently incurs non-trivial runtime overhead and omits revocation processing, limiting immediate deployability.

Availability threats are the focus of Shi et al., who study denial-of-service attacks triggered by pathological certificate structures [11]. Their automated tool crafts certificates that amplify computational hotspots in parsing code; applied to seven mainstream libraries, it uncovers 18 previously unknown and 12 known resource-exhaustion bugs. The authors conclude that strict standards compliance alone does not preclude resource-exhaustion vectors, highlighting a dimension that has been largely overlooked by earlier parser-safety work.

Although centred on CBOR rather than X.509, Ramananandro et al.’s work shows how separation-logic-based parser/serializer combinators can scale to real-world, recursive binary formats while guaranteeing non-malleability, constant-stack parsing and end-to-end correctness [75]. Their EverCBOR/EverCDDL toolchain demonstrates that strong guarantees can be achieved even for richly typed, extensible formats, providing a blueprint that could be ported to X.509 in future research.

Finally, EverParse, introduced by Ramananandro et al., pioneered the generation of verified, zero-copy parsers for tag-length-value protocols [74]. By coupling an F* parser-combinator library with a DSL-to-F* compiler, the framework produces C-compatible code that is memory-safe, bijective with its serializer, and non-malleable. EverParse’s successful integration into miTLS and other applications established the feasibility of automated, high-performance, verified parsing, and it remains the foundation on which several later systems, including PulseParse, are built.

9

Conclusion

This work explored the design and evaluation of minimal, interoperable certificate profiles for federated space PKI, addressing the challenges posed by PQ cryptography and constrained environments. This chapter concludes the study by summarising the methodology and key findings, and also highlights a promising direction for future research on certificate validation.

Summary

Future federated space missions must balance constrained onboard resources, the very challenging PQ migration deadline (2030-2035), and cross-domain interoperability. This paper evaluated PQ certificate formats and their trade-offs in a federated space setting and outlined a minimal, structured extension profile for space links drawing from terrestrial federated PKIs. Then, the work performed a quantitative comparative analysis between X.509 and C509 in terms of message size and software complexity. Together, the findings aim to provide additional insights and support standardisation bodies, such as CCSDS Security Working Groups, in tailoring a certificate profile for federated environments, with a focus on space links.

Through the review of PQ formats, this work highlighted that regulatory divergence—e.g., between composite and pure approaches—poses interoperability risks. This work advocates for federation-wide support of composite certificates to ensure dual-algorithm trust during PQ migration.

The study then addressed the issue of unrestricted extension use by analysing profiles in terrestrial federated PKIs. Based on these, the work proposed a minimal set of extension profiles to reduce complexity while preserving essential functionality, in alignment with the CCSDS IGCA requirements to support federation-wide interoperability.

To address the lack of tooling for C509, the `c509-native` prototype was developed as part of this work. The tool supports PQ algorithms and enables the creation of certificates, signing requests, and revocation lists. Released as open source, it serves as a proof of concept to promote accessibility and broader adoption of C509. Nevertheless, `c509-native` requires additional improvements to mature as a potential tool for future CBOR-encoded PKI.

The comparative analysis confirms that X.509's verbosity and implementation complexity hinder its use in constrained systems. C509 mitigates these issues through compact CBOR encoding and reduced structural overhead, yielding 40–45% size reductions for traditional certificates and up to 60% for CRLs. For PQ and hybrid certificates, the gains are negligible. Using well-established heuristics and practical experiments, this work quantifies the substantial difference in software complexity between X.509 and C509 implementations. The results demonstrate approximately 80% reduced codebase footprint when using C509, as well as 2-3x reduced cyclomatic complexity and a decrease of over 60% for total Halstead volume (lower cognitive complexity). The reduction in software complexity is perhaps more relevant than the size gains, especially when considering the high demands of space software qualification requirements and security certification requirements.

Among C509 deployment options, re-encoded certificates with gateway-based translation offer a practical compromise. They preserve X.509 compatibility while offloading DER parsing from constrained clients (such as spacecraft) to the gateway (ground and mission control). However, adoption remains limited by the draft status of C509 and the absence of standardised CBOR-native revocation protocols.

Future Work

While this study focuses on defining a minimal and interoperable certificate profile, the greater challenge is ensuring its efficient validation across heterogeneous nodes, from constrained satellites to ground systems. Limited computation, intermittent links, and unreliable time references render onboard path construction, revocation checks, and policy enforcement disproportionately demanding, making certificate validation a significant bottleneck.

A promising mitigation is the Server-based Certificate Validation Protocol (SCVP) [20], which enables delegated validation (DPV) or path discovery (DPD) to a trusted server [86]. SCVP offers particular advantages: it offloads computational burdens, enables signature verification across differing cryptographic stacks, and centralises policy enforcement. Validation policies can mandate anchors, revocation sources, and required extensions, promoting cross-domain consistency.

SCVP meets key security requirements: messages can be signed or MAC-protected, include nonces to prevent replay, and allow client-specified time references—relevant for delay-tolerant networks. SCVP can support relayed requests, such as those from lunar relays to terrestrial authorities, thereby reducing the need for clients to process CRLs or OCSP responses directly. It is already supported by commercial solutions [87, 88, 89] and has seen practical use in mobile networks [90].

Nevertheless, further research is needed to assess SCVP's performance and trust implications in space. Open questions include protocol latency over deep-space links, the resilience of centralised validation, and compatibility with delay-tolerant revocation schemes. Future work should involve prototyping a bridge validation authority within a CCSDS IGCA-aligned federation, evaluating it under simulated space conditions, and developing tooling to manage and apply SCVP policies. Overall, SCVP emerges as a promising enabler for scalable, policy-driven validation in quantum-ready federated space PKIs.

References

[1] NASA Office of Inspector General. *NASA's Management of the Artemis Missions*. Tech. rep. IG-22-003. Accessed: June 2025. National Aeronautics and Space Administration (NASA), Nov. 2021. URL: https://oig.nasa.gov/wp-content/uploads/2024/02/IG-22-003.pdf.

[2] S. Fuller et al. "Gateway Program Development Progress". In: *Proceedings of the 75th International Astronautical Congress (IAC)*. IAC-24,B3,1,10,x85927. Accessed: June 2025. Milan, Italy: International Astronautical Federation (IAF), Oct. 2024. URL: https://ntrs.nasa.gov/citations/20240012550.

[3] Consultative Committee for Space Data Systems (CCSDS). *Symmetric Key Management*. Issue 1. CCSDS Recommended Practice 354.0-M-1. Washington, DC, USA: CCSDS Secretariat, National Aeronautics and Space Administration (NASA), Dec. 2023.

[4] Consultative Committee for Space Data Systems (CCSDS). *Space Missions Key Management Concept*. Issue 1. CCSDS Informational Report 350.6-G-1. Washington, DC, USA: CCSDS Secretariat, National Aeronautics and Space Administration (NASA), Nov. 2011.

[5] Consultative Committee for Space Data Systems (CCSDS). *Intergovernmental Certification Authority*. Issue 1. CCSDS Experimental Specification 357.1-O-1. Washington, DC, USA: CCSDS Secretariat, National Aeronautics and Space Administration (NASA), Dec. 2024.

[6] D. Cooper et al. *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. RFC 5280. Internet Engineering Task Force (IETF), May 2008. URL: https://datatracker.ietf.org/doc/html/rfc5280.

[7] Consultative Committee for Space Data Systems (CCSDS). *Authentication Credentials*. Issue 1. CCSDS Recommended Standard 357.0-B-1. Washington, DC, USA: CCSDS Secretariat, National Aeronautics and Space Administration (NASA), July 2019.

[8] J. P. Mattsson et al. *CBOR Encoded X.509 Certificates (C509 Certificates)*. Internet-Draft draft-ietf-cose-cbor-encoded-cert. Accessed: June 2025. Internet Engineering Task Force (IETF), Mar. 2025. URL: https://datatracker.ietf.org/doc/html/draft-ietf-cose-cbor-encoded-cert.

[9] A. Ebalard, P. Mouy, and R. Benadjila. "Journey to a RTE-free X.509 parser". In: *Proceedings of the Symposium on Information and Communications Technology Security (SSTIC)*. Rennes, France, 2019, pp. 1–20.

[10] A. Barenghi, N. Mainardi, and G. Pelosi. "Systematic Parsing of X.509: Eradicating Security Issues with a Parse Tree". In: *Journal of Computer Security* 26.6 (2018), pp. 817–849.

[11] B. Shi et al. "X.509DoS: Exploiting and Detecting Denial-of-Service Vulnerabilities in Cryptographic Libraries using Crafted X.509 Certificates". In: *Proceedings of the 34th USENIX Security Symposium*. Seattle, WA: USENIX Association, 2025.

[12] *Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks*. ITU-T Recommendation X.509. International Telecommunication Union, Oct. 2019. URL: https://www.mhlw.go.jp/content/10808000/001012539.pdf.

[13] S. Tatschner et al. "ParsEval: Evaluation of Parsing Behavior using Real-world Out-in-the-wild X.509 Certificates". In: *Proceedings of the 19th International Conference on Availability, Reliability and Security*. ARES '24. Vienna, Austria: Association for Computing Machinery, 2024, 143:1–143:9. DOI: 10.1145/3664476.3669935.

[14] OpenSSL Project. *Deprecated Extensions in x509v3_config*. OpenSSL Documentation, Version 3.5. Accessed: June 2025. 2025.

[15] S. Turner et al. *Internet X.509 Public Key Infrastructure: Algorithm Identifiers for ML-KEM*. Internet-Draft. Accessed: June 2025. Internet Engineering Task Force (IETF), Apr. 2025. URL: https://datatracker.ietf.org/doc/html/draft-ietf-lamps-kyber-certificates.

[16] J. Massimo et al. *Internet X.509 Public Key Infrastructure: Algorithm Identifiers for ML-DSA*. Internet-Draft draft-ietf-lamps-dilithium-certificates. Accessed: June 2025. Internet Engineering Task Force (IETF), Apr. 2025. URL: https://datatracker.ietf.org/doc/html/draft-ietf-lamps-dilithium-certificates.

[17] K. Bashiri et al. *Internet X.509 Public Key Infrastructure: Algorithm Identifiers for SLH-DSA*. Internet-Draft. Accessed: June 2025. Internet Engineering Task Force (IETF), May 2025. URL: https://datatracker.ietf.org/doc/html/draft-ietf-lamps-x509-slhdsa.

[18] C. Bormann and P. Hoffman. *Concise Binary Object Representation (CBOR)*. RFC 8949. Internet Engineering Task Force (IETF), Dec. 2020. URL: https://datatracker.ietf.org/doc/html/rfc8949.

[19] C. Bormann, M. Ersue, and A. Keränen. *Terminology for Constrained-Node Networks*. RFC 7228. Internet Engineering Task Force (IETF), May 2014. URL: https://datatracker.ietf.org/doc/html/rfc7228.

[20] T. Freeman et al. *Server-Based Certificate Validation Protocol (SCVP)*. RFC 5055. Internet Engineering Task Force (IETF), Dec. 2007. URL: https://datatracker.ietf.org/doc/html/rfc5055.

[21] E. B. Barker. *Recommendation for Key Management – Part 1: General*. NIST Special Publication 800-57 Part 1 Rev. 5. Accessed: June 2025. National Institute of Standards and Technology (NIST), May 2020. URL: https://csrc.nist.gov/pubs/sp/800/57/pt1/r5/final.

[22] M. Nystrom and B. Kaliski. *PKCS #10: Certification Request Syntax Specification Version 1.7*. RFC 2986. Internet Engineering Task Force (IETF), Nov. 2000. URL: https://datatracker.ietf.org/doc/html/rfc2986.

[23] Wikimedia Commons contributors. *Public-Key Infrastructure*. Accessed: June 2025. 2020. URL: https://commons.wikimedia.org/wiki/File:Public-Key-Infrastructure.svg.

[24] R. Prodanović, I. Vulić, and I. Tot. “A Survey of PKI Architecture”. In: *ERAZ 2019 – Selected Papers*. Belgrade, Serbia, 2019, pp. 169–175. DOI: 10.31410/ERAZ.S.P.2019.169. URL: https://doi.org/10.31410/ERAZ.S.P.2019.169.

[25] International Telecommunication Union (ITU-T). *ITU-T X.690: Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*. ITU-T Recommendation X.690. Accessed: June 2025. International Telecommunication Union (ITU), Feb. 2021. URL: https://www.itu.int/rec/T-REC-X.690.

[26] S. Santesson, M. Nystrom, and T. Polk. *Internet X.509 Public Key Infrastructure: Qualified Certificates Profile*. RFC 3739. Internet Engineering Task Force (IETF), Mar. 2004. URL: https://datatracker.ietf.org/doc/html/rfc3739.

[27] S. Tuecke et al. *Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile*. RFC 3820. Internet Engineering Task Force (IETF), June 2004. URL: https://datatracker.ietf.org/doc/html/rfc3820.

[28] H. Tschofenig and T. Fossati. *Transport Layer Security (TLS) / Datagram Transport Layer Security (DTLS) Profiles for the Internet of Things*. RFC 7925. Internet Engineering Task Force (IETF), July 2016. URL: https://datatracker.ietf.org/doc/html/rfc7925.

[29] J. Peterson and S. Turner. *Secure Telephone Identity Credentials: Certificates*. RFC 8226. Internet Engineering Task Force (IETF), Feb. 2018. URL: https://datatracker.ietf.org/doc/html/rfc8226.

[30] C. Bormann and P. Hoffman. *Concise Data Definition Language (CDDL): A Notational Convention to Express CBOR and JSON Data Structures*. RFC 8610. Internet Engineering Task Force (IETF), June 2019. URL: https://datatracker.ietf.org/doc/html/rfc8610.

[31] S. Santesson et al. *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*. RFC 6960. Internet Engineering Task Force (IETF), June 2013. URL: https://datatracker.ietf.org/doc/html/rfc6960.

[32] W. Newhouse et al. *Migration to Post-Quantum Cryptography: Preparation for Considering the Implementation and Adoption of Quantum Safe Cryptography*. NIST Special Publication 1800-38. Accessed: June 2025. National Institute of Standards and Technology (NIST), Dec. 2023. URL: https://csrc.nist.gov/pubs/sp/1800/38/iprd-(1).

[33] L. Chen et al. *Report on Post-Quantum Cryptography*. NIST Interagency or Internal Report 8105. Accessed: June 2025. National Institute of Standards and Technology (NIST), Apr. 2016. URL: https://csrc.nist.gov/pubs/ir/8105/final.

[34] G. Alagic et al. *Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process*. NIST Interagency or Internal Report 8413. Accessed: June 2025. National Institute of Standards and Technology (NIST), July 2022. URL: https://csrc.nist.gov/pubs/ir/8413/upd1/final.

[35] National Institute of Standards and Technology (NIST). *FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard*. FIPS 203. U.S. Department of Commerce, Aug. 2024. URL: https://csrc.nist.gov/pubs/fips/203/final.

[36] National Institute of Standards and Technology (NIST). *FIPS 204: Module-Lattice-Based Digital Signature Standard*. FIPS 204. U.S. Department of Commerce, Aug. 2024. URL: https://csrc.nist.gov/pubs/fips/204/final.

[37] National Institute of Standards and Technology (NIST). *FIPS 205: Stateless Hash-Based Digital Signature Standard*. FIPS 205. U.S. Department of Commerce, Aug. 2024. URL: https://csrc.nist.gov/pubs/fips/205/final.

[38] P.-A. Fouque et al. *Falcon: Fast-Fourier Lattice-based Compact Signatures over NTRU*. Technical Report. Accessed: June 2025. IBM Research, Jan. 2020. URL: https://research.ibm.com/publications/falcon-fast-fourier-lattice-based-compact-signatures-over-ntru.

[39] C. Aguilar Melchor et al. *HQC: Hamming Quasi-Cyclic – Fourth Round Specification*. Technical Report. Accessed: June 2025. HQC Project, Feb. 2025. URL: https://pqc-hqc.org/doc/hqc-specification_2025-02-19.pdf.

[40] G. Alagic et al. *Status Report on the Fourth Round of the NIST Post-Quantum Cryptography Standardization Process*. NIST Interagency or Internal Report 8545. Accessed: June 2025. National Institute of Standards and Technology (NIST), Mar. 2025. URL: https://csrc.nist.gov/pubs/ir/8545/final.

[41] D. Cooper et al. *Recommendation for Stateful Hash-Based Signature Schemes*. NIST Special Publication 800-208. Accessed: June 2025. National Institute of Standards and Technology (NIST), Oct. 2020. URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-208.pdf.

[42] A. Hülsing et al. *XMSS: eXtended Merkle Signature Scheme*. RFC 8391. Internet Engineering Task Force (IETF), May 2018. URL: https://datatracker.ietf.org/doc/html/rfc8391.

[43] D. McGrew, M. Curcio, and S. Fluhrer. *Leighton-Micali Hash-Based Signatures*. RFC 8554. Internet Engineering Task Force (IETF), Apr. 2019. URL: https://datatracker.ietf.org/doc/html/rfc8554.

[44] F. Driscoll, M. Parsons, and B. Hale. *Terminology for Post-Quantum Traditional Hybrid Schemes*. Internet-Draft draft-ietf-pquip-pqt-hybrid-terminology. Accessed: June 2025. Internet Engineering Task Force (IETF), Jan. 2025. URL: https://datatracker.ietf.org/doc/html/draft-ietf-pquip-pqt-hybrid-terminology.

[45] M. Rossi et al. *ANSSI views on the Post-Quantum Cryptography transition (2023 follow up)*. Technical Report. Accessed: June 2025. Agence nationale de la sécurité des systèmes d'information (ANSSI), Dec. 2023. URL: https://cyber.gouv.fr/sites/default/files/document/follow_up_position_paper_on_post_quantum_cryptography.pdf.

[46] Federal Office for Information Security (BSI). *Cryptographic Mechanisms: Recommendations and Key Lengths*. Technical Guideline TR-02102-1. Accessed: June 2025. Federal Office for Information Security (BSI), Mar. 2025. URL: https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TG02102/BSI-TR-02102-1.pdf.

[47] European Commission. *A Coordinated Implementation Roadmap for the Transition to Post-Quantum Cryptography*. Tech. rep. Part 1, Version 1.1. Accessed: June 2025. European Commission, June 2025. URL: https://digital-strategy.ec.europa.eu/en/library/coordinated-implementation-roadmap-transition-post-quantum-cryptography.

[48] National Security Agency. *CNSA Suite 2.0 and Quantum Computing FAQ*. Cybersecurity Information Sheet U/OO/194427-22 | PP-24-4014. Version 2.1. Accessed: June 2025. Fort Meade, MD: National Security Agency, Dec. 2024. URL: https://media.defense.gov/2022/Sep/07/2003071836/-1/-1/0/CSI_CNSA_2.0_FAQ_.PDF.

[49] SonarSource. *Understanding Measures and Metrics*. SonarQube Server 10.8 documentation, accessed 3 July 2025. 2024. URL: https://docs.sonarsource.com/sonarqube-server/10.8/user-guide/code-metrics/metrics-definition/.

[50] Scientific Toolworks, Inc. *Metrics Overview*. Accessed: June 2025. 2021. URL: https://support.scitools.com/support/solutions/articles/70000582289-metrics-overview.

[51] *ECSS-Q-HB-80-04A: Space Product Assurance — Software Metrication Programme Definition and Implementation*. Accessed: June 2025. European Cooperation for Space Standardization (ECSS). Noordwijk, The Netherlands, Mar. 2011. URL: https://ecss.nl/wp-content/uploads/handbooks/ecss-q-hb/ECSS-Q-HB-80-04A30March2011.pdf.

[52] *IEEE Standard Dictionary of Measures to Produce Reliable Software*. New York, NY, USA: IEEE Computer Society, 1989.

[53] *IEEE Guide for the Use of IEEE Standard Dictionary of Measures to Produce Reliable Software*. New York, NY, USA: IEEE Computer Society, 1989.

[54] W. Burr et al. *NIST Migration to Post-Quantum Cryptography*. NIST Interagency Report IR 8547 (Initial Public Draft). Accessed: June 2025. National Institute of Standards and Technology (NIST), Mar. 2024. URL: https://nvlpubs.nist.gov/nistpubs/ir/2024/NIST.IR.8547.ipd.pdf.

[55] M. Ounsworth et al. *Composite Public and Private Keys For Use In Internet PKI*. Internet-Draft draft-ounsworth-pq-composite-keys. Accessed: June 2025. Internet Engineering Task Force (IETF), July 2021. URL: https://datatracker.ietf.org/doc/html/draft-ounsworth-pq-composite-keys.

[56] M. Ounsworth et al. *Composite ML-KEM for use in X.509 Public Key Infrastructure and CMS*. Internet-Draft draft-ietf-lamps-pq-composite-kem. Accessed: June 2025. Internet Engineering Task Force (IETF), Mar. 2025. URL: https://datatracker.ietf.org/doc/html/draft-ietf-lamps-pq-composite-kem.

[57] M. Ounsworth et al. *Composite ML-DSA for use in X.509 Public Key Infrastructure and CMS*. Internet-Draft draft-ietf-lamps-pq-composite-sigs. Accessed: June 2025. Internet Engineering Task Force (IETF), Mar. 2025. URL: https://datatracker.ietf.org/doc/html/draft-ietf-lamps-pq-composite-sigs.

[58] Philip Lafrance. *ISARA Dedicates Four Hybrid Certificate Patents to the Public*. Message to the pqc-forum mailing list. Accessed: June 2025. Oct. 2022. URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/QjNsza6bQ-U/m/3DWBw4kYDAAJ.

[59] A. Truskovsky et al. *Multiple Public□Key Algorithm X.509 Certificates*. Internet□Draft. Accessed: June 2025. Internet Engineering Task Force (IETF), Aug. 2018. URL: https://datatracker.ietf.org/doc/html/draft-truskovsky-lamps-pq-hybrid-x509.

[60] C. Bonnell et al. *A Mechanism for Encoding Differences in Paired Certificates*. Internet-Draft draft-bonnell-lamps-chameleon-certs. Accessed: June 2025. Internet Engineering Task Force (IETF), Apr. 2025. URL: https://datatracker.ietf.org/doc/html/draft-bonnell-lamps-chameleon-certs.

[61] A. Becker, R. Guthrie, and M. J. Jenkins. *Related Certificates for Use in Multiple Authentications within a Protocol*. RFC 9763. Internet Engineering Task Force (IETF), June 2025. URL: https://datatracker.ietf.org/doc/html/rfc9763.

[62] C. Wang, W. Xue, and J. Wang. “Integration of Quantum-Safe Algorithms into X.509v3 Certificates”. In: *2023 IEEE 3rd International Conference on Electronic Technology, Communication and Information (ICETCI)*. Changchun, China, 2023, pp. 384–388. DOI: 10.1109/ICETCI57876.2023.10176713. URL: https://doi.org/10.1109/ICETCI57876.2023.10176713.

[63] T. Okubo et al. *A Mechanism for X.509 Certificate Discovery*. Internet-Draft draft-ietf-lamps-certdiscovery. Accessed: June 2025. Internet Engineering Task Force (IETF), Apr. 2025. URL: https://datatracker.ietf.org/doc/draft-ietf-lamps-certdiscovery.

[64] S. Kousidis et al. *Post-Quantum Cryptography in OpenPGP*. Internet-Draft draft-ietf-openpgp-pqc. Accessed: June 2025. Internet Engineering Task Force (IETF), Apr. 2025. URL: https://datatracker.ietf.org/doc/html/draft-ietf-openpgp-pqc.

[65] Consultative Committee for Space Data Systems (CCSDS). *CCSDS Cryptographic Algorithms*. Issue 2. CCSDS Recommended Standard 352.0-B-2. Washington, DC, USA: CCSDS Secretariat, National Aeronautics and Space Administration (NASA), Aug. 2019.

[66] Federal Public Key Infrastructure Policy Authority. *Federal Bridge Certification Authority (FBCA) X.509 Certificate and CRL Extensions Profile*. Tech. rep. Version 2.0. Accessed: June 2025. U.S. Federal PKI Policy Authority, Oct. 2022. URL: https://www.idmanagement.gov/docs/fpki-x509-cert-profiles-fbca.pdf.

[67] Booz Allen Hamilton Inc. and National Institute of Standards and Technology. *Federal Public Key Infrastructure (PKI) X.509 Certificate and CRL Extensions Profile*. Tech. rep. Accessed: June 2025. Federal PKI Policy Authority, Oct. 2005. URL: https://www.foundationfortrustedidentity.org/federally-certified/uploads/fti-ca/Federal%20Public%20Key%20Infrastructure%20(PKI)%20X.509%20Certificate%20and%20CRL%20Extensions%20Profile.pdf.

[68] Consultative Committee for Space Data Systems (CCSDS). *Rationale, Scenarios, and Requirements for DTN in Space*. Issue 1. CCSDS Informational Report 734.0-G-1. Washington, DC, USA: CCSDS Secretariat, National Aeronautics and Space Administration (NASA), Aug. 2010.

[69] J. Alakuijala and Z. Szabadka. *Brotli Compressed Data Format*. RFC 7932. Internet Engineering Task Force (IETF), July 2016. URL: https://datatracker.ietf.org/doc/html/rfc7932.

[70] L. Ardito et al. “rust-code-analysis: A Rust library to analyze and extract maintainability information from source codes”. In: *SoftwareX* 12 (2020), p. 100635. ISSN: 2352-7110. DOI: https://doi.org/10.1016/j.softx.2020.100635. URL: https://www.sciencedirect.com/science/article/pii/S2352711020303484.

[71] Consultative Committee for Space Data Systems. *Mission Planning and Scheduling*. Green Book, Informational Report CCSDS 529.0-G-1. Washington, DC: CCSDS, June 2018.

[72] Virtual Applications and Implementations Research Lab. *eBACS: ECRYPT Benchmarking of Cryptographic Systems*. Accessed: June 2025. URL: https://bench.cr.yp.to/index.html.

[73] S. Sun, Y. He, and H.-Y. Lin. *Convertible Forms with Multiple Keys and Signatures For Use In Internet X.509 Certificates*. Internet-Draft draft-sun-lamps-hybrid-scheme. Accessed: June 2025. Internet Engineering Task Force (IETF), Apr. 2025. URL: https://datatracker.ietf.org/doc/draft-sun-lamps-hybrid-scheme.

[74] T. Ramananandro et al. “EverParse: Verified Secure Zero-Copy Parsers for Authenticated Message Formats”. In: *28th USENIX Security Symposium, USENIX Security 2019, Santa Clara, CA, USA, August 14-16, 2019*. USENIX Association, 2019, pp. 1465–1482.

[75] T. Ramananandro et al. “Secure Parsing and Serializing with Separation Logic Applied to CBOR, CDDL, and COSE”. Accessed: June 2025. 2025. URL: https://arxiv.org/abs/2505.17335.

[76] D. Koisser et al. “TruSat: Building Cyber Trust in Collaborative Spacecraft Networks”. In: *2022 IEEE Aerospace Conference (AERO)*. 2022, pp. 1–12. DOI: 10.1109/AERO53065.2022.9843330. URL: https://doi.org/10.1109/AERO53065.2022.9843330.

[77] D. Koisser et al. “V’CER: Efficient Certificate Validation in Constrained Networks”. In: *Proceedings of the 31st USENIX Security Symposium*. Boston, MA: USENIX Association, 2022, pp. 4491–4508. URL: https://www.usenix.org/conference/usenixsecurity22/presentation/koisser.

[78] D. Koisser et al. “Orbital Trust and Privacy: SoK on PKI and Location Privacy Challenges in Space Networks”. In: *Proceedings of the 33rd USENIX Security Symposium*. Philadelphia, PA: USENIX Association, 2024, pp. 6093–6111. URL: https://www.usenix.org/conference/usenixsecurity24/presentation/koisser.

[79] J. Smailes et al. “KeySpace: Public Key Infrastructure Considerations in Interplanetary Networks”. In: *arXiv preprint arXiv:2408.10963* (2024). Accessed: June 2025. URL: https://arxiv.org/abs/2408.10963.

[80] P. Kampanakis et al. “The Viability of Post-Quantum X.509 Certificates”. In: *Cryptology ePrint Archive* 2018.063 (2018). URL: https://eprint.iacr.org/2018/063.

[81] M. Raavi et al. “Performance Characterization of Post-Quantum Digital Certificates”. In: *2021 International Conference on Computer Communications and Networks (ICCCN)*. Athens, Greece, 2021, pp. 1–9. DOI: 10.1109/ICCCN52240.2021.9522179. URL: https://doi.org/10.1109/ICCCN52240.2021.9522179.

[82] N. Ricchizzi, C. Schwinne, and J. Pelzl. *Applied Post Quantum Cryptography: A Practical Approach for Generating Certificates in Industrial Environments*. May 2025. DOI: 10.48550/arXiv.2505.04333. URL: https://arxiv.org/abs/2505.04333.

[83] J. Callas et al. *OpenPGP Message Format*. RFC 4880. Internet Engineering Task Force (IETF), Nov. 2007. URL: https://datatracker.ietf.org/doc/html/rfc4880.

[84] F. Forsby et al. “Lightweight X.509 Digital Certificates for the Internet of Things”. In: *Interoperability, Safety and Security in IoT*. Ed. by G. Fortino et al. Cham: Springer International Publishing, 2018, pp. 123–133. DOI: 10.1007/978-3-319-93797-7_14. URL: https://doi.org/10.1007/978-3-319-93797-7_14.

[85] J. Debnath et al. “ARMOR: A Formally Verified Implementation of X.509 Certificate Chain Validation”. In: *IEEE Symposium on Security and Privacy, SP 2024, San Francisco, CA, USA, May 19-23, 2024*. IEEE, 2024, pp. 1462–1480.

[86] D. Pinkas and R. Housley. *Delegated Path Validation and Delegated Path Discovery Protocol Requirements*. RFC 3379. Internet Engineering Task Force (IETF), Sept. 2002. URL: https://datatracker.ietf.org/doc/html/rfc3379.

[87] HID Global. *ActivID Validation Authority: SCVP Support*. Accessed: June 2025. URL: https://docs.hidglobal.com/activid-validation-authority-v7.4/docs/overview/scvp-validation.htm.

[88] Ascertia. *ADSS SCVP Server*. Accessed: June 2025. URL: https://www.ascertia.com/products/adss-scvp-server/.

[89] Axway. *Validation Authority Suite*. Accessed: June 2025. URL: https://www.axway.com/en/products/axway-va-suite.

[90] Motorola Inc. “Utilizing a stapling technique with a server-based certificate validation protocol”. Patent Application US20130159703A1. Accessed: June 2025. 2013. URL: https://patents.google.com/patent/US20130159703A1/en.

A

Post-Quantum Certificate Format Experiments

Test Certificates

Characteristics of the testing certificates:

- Are self-signed X.509, version 3.
- Have the serial numbers equal to the maximum positive signed integer on 8 bytes.
- Use either ML-DSA:44 alone or in combination with ECDSA:P-256.
- Have the same issuer and subject ”Test Root CA”.
- Have the same validity, with the times encoded as GeneralizedTime.
- Contain no extension other than the ones needed to store the hybrid components.

Listing A.1: Pure Self-Signed Certificate with ML-DSA:44.

```
-----BEGIN CERTIFICATE-----
MIIPMjCCBaigAwIBAgIIf/////////8wCwYJYIZIAWUDBAMRMBcxFTATBgNVBAMMDFRlc3QgUm9vdCBDQTAiGA8yMTAwM
DcyMjEyMDAwMFoYDzIxMDAwNzIyMTQwMDAwWjAXMRUwEwYDVQQDDAxUZXN0IFJvb3QgQ0EwggUyMAsGCWCGSAFlAwQDEQ
OCBSEAI2zBmU3x4bLH+ymqb/g31vtQK3jaJi05xhV0XXoblcQTwmaKC+M3KAv08phzXgviQ/jMczjwGLdLz3uJJR5QTHy
kekprBlpF5k1Q+tvi8psbp5k8r9B9VC3/PG6wDXtMrX+iZLEhPr6/ev0x4dN3qJsFjfKHlPAHtvN91WBtggMdAqyeuRCF
QrE4np2KLxOevLOUsUmoeO/Om1N049SRx/3ya5+UuyGeL5b77K9ssezBxI+yxVgDx+Vii7eAuVsNiiHG1cX8BJDoSsRhf
2HvaWIoq+mSKmbDon8Dv+YbFLu2pDm782CrFuEXW4z9SBtwl10RMpMyQnzF2vLmYKRo3Ec25vZ8q7JHaOnZyxYeMsH/uc
Ry1co0IER3eKa6QzN2ZOv+quDD96VU6EOJuJiuPAflmNQ6eEeZHypZIcuJ5AkUXNUCj65ul2FOQl3UqtQ/XbF9DUqImRy
yl3j3U+hZ/v4Ufs2hwLH40Q6XyjfeZU32J9xQlBTo/D9hhLHb1ZIIl4zdE8Yn3NfjUYSIwkYJTp6qDoH/ANf4MCqMcl1k
qPov8Ar7G+28YP3+rmfJFNcxkdvVhVI3Nslu94/+NVrJtLIKkZzNTzAXkMGdvBkk2lcQlKy62ReI12eL85dHvlNBfQ2UQ
6tr+an7Itiz+XKrZkVLCnXQC5itds2BOFKOmUqZTA4/l+xSBj85M33nPX7YmAXUdCzKZb5wZ+9oj0+mOfwOo8KSJjmMaB
wDu3PA6F3ghrwuo6LoHXXJ0Vx3OOdhD3PpMm7KnvMA1esOcU/NPXsItPWolcjn/dYicTiXUHBkCMiHOxVpPWbDakiS4u1
rmXzjuqixvk0zkNJaB0bwkIYhE9ZEQDU4fPlfTvuLGRoKedc5V5VUg5iE7jkVfOJhXXEY+53mBe9zScGnr6sCi6KJlYoK
XvYei9yI7Igv1Qapr5hEU6bhsV/RVUrydhjwndklzxsIXOoTYCM+2HWMRNvsaLxcL2yIK2cRRrhRWMLrcmHNRlGn7XFwb
cePulj478QGQ7rQUrCKmMfuqkE5N+Xf6a+tUIQxrphOFz3QBQVexPEEWd9m0j4ZZvtapRyywTXsyL+aWYWH4G+it7i0s/
jwl+lCPteBnxNiiPqQxJRz13d3QFzkVH1SNlJUtRXu9xnYOzMy//qimm08ayFRYl3O1KeE+VOWlTjjQZe2+fY6vyUWLkI
RPS9StkYNPLdVwF9x5OOv7NG9wbVWytTQImgq2Aqn49d32F1lUt/TwO8GQSgVEPTSzRNxefI/mJ7ohyQlSOlH7s1xcxET
L572E07CheuCaCCm2XYTjRTxbRDrO5/io2FiGGNgNWVTQzQvzVKi5+RuxHiEaZBPLm3RWcyCI9YKp78QrxHHwXH1iVYBw
/VGp1Hc8RJUTTu2wqdqnmwTCehYu+SN54KNhDaYeL2UH96bWsjmVxsShF7bfIB2UZyjqcFQ1APLDwwqcd4pWH6RmUu4Xv
JZI6GqmKeA2RnDXAOPtUhpgTVGKGQ2Lagh9Joh4OzsnDBZT4tukgljk1/zF2Z12MACOh/4c1pQLKaYSAqI9c68ZvdrAPp
CUs6piaz2Zh+UaEuE2SshaNNMuYjo/gxh/9rAjBjNdpesUw0oJwK3J/URxpucqIq+H4bpKvxcq91+dOF+YGxvFkDXM0FJ
X084Z37TrkkqLJKOPMzpc4tC9SBRwEOt8ks1COwo5oIhNbCuGzSJ1nFjvVFxvfpUtoqN073dvB/iu6JbZjALBglghkgBZ
QMEAxEDggl1ADW2GvWrxGVPoPL1kj0tNxvdxg0CYNSuSqiWQJCg6DvtyDQ0w1jWc4qeQtC2r5Uf1f4ZZlaI8/tBc5gYjO
BrrClZwAX17JhrMCQc9ZmMj5HrV5jw3ZU07Eec0Xtp8R/AmzaaY8cwUZbE6w5sQNcdXU9BIPhwBTD6g1AQ9KtK9Zao/9a
AAt+Q2a+vrkg5smUP32Alh8og1LhfZ0cLb+qRycgw7pPSuw+IuDcVWeb74C79npNX8i7Oq94XKH/RrSSvdL3FanvyPUbk
MRU7DEoOPZ55nvQMDx8Ngo7KkD21ZdGTgqg/UKYl/Ci9Akdlgzyp BLJ8ADxGJXceToicXC9vDK3X2kkVHspc2tBkWoRMf
1gZLUffoGHSGRfJEjFLHXWe4KyixeUqfC5Lu/u2NKF/zE6ztLjkkAMGZuVW+SmIKgpws25CNCW7B1i0Wut/co9bi6uT81
kgB1Cv84W11rRlxPI0ktGSJ3/OQeZu9OfPadFu3/DiwCJUFILYCBsy5YcwY55U2RpQm+5nbK75ZGMov/qnDvzgerWkLLy
KsiyNxGZe0/GYJK3ErQOxtYRpVVwlQOlRRbjbjovxQ2dcfuQ298KySVZm0GLlx8s+4OYGqPtYbYuZwOuPYGlJrhBdjlNf
dHNZPRR1mU8FNxrygwkfIouXvE1tb93My/Nm6T8RpMEN6IKFIpGXyJyK554uqwI+bo86IHjBxOqsAImFGYFw9xhWwCwyY
```

```
fzo6NKYUKZ3zNKVLUS7cDCw7rAJLHljSe10jlhpGVEyXOFic+4DVbMWGqod8trt5PQmLokTDe56od7XmUFrsCbOCyKZ9i
lKGxruUHldZ4Za7v0SSgP8nneY5mzMkxkkkHKP1xBa78LQntkAPb2c1AwXFkNAoszu64E5cgNA+M6zB8Kcvk9Lvym4Xoj
lVdLuM/gZfU1lmVePP43kR4MUtu+Lx4inKIMUBT9WzkH4rHEFm+iQAVfco+wWhrD5kSD7qZRIM9vgNy7DUxDPLK31yu3u
GAO/G8snFF3U08/2kZXcUZ/OM4SbrONQydxNgsX+C62JKtEHo9viubK6d0za0QxbAySV/rYdnb3TTqMlSn7QdIwePV8N7
o0xJeuUI5JS/ZsNhNu+mOAmktx2TjhHZsirlEjE2XC0raOHyx8CTHguBUo2y2Xc377Kx2S+k8Y/aM6u+08OtetCYD/57j
TJg6QKtdqelaUEkM5ML41g7OFWzgyIkLgBEJxmX48ek4fUIIZXCQ/POQmia7ORvNM/j3ksJ9Xs8sy1vXOm+AmuEHpECNT
/o+gOw1rRPSaiAZa4CFhokuhkKZTlzZtDyxaGiUN+MngU5pQJ3AfDQFOEKKVGiYnHaw4ZGJ5+WTn5SKpp6RapIa0fUtqK
lCgzRu3a094i6NPr+8stcJqD/HbyUVaTR4Pz6D3F8R1Bxv/l4i0T2T6SOJk4/58ErvBOzPizQd8t99qTyt3klXhEAEFXV
z7Mb9lhg5WxGIEGgRj8jeHRrdHWqi8YiVS8wDisuUxIpaFm650nm/KeHh7g0hvMd38P+N3P2m28SZmt9bzKlgE+XS666e
0oGI/Yrtv820fv1Kb6Ybq7lmBVNqz1omW+UgPOzdgG3bbmfUPtEjJjIXKXob5jZKY7czihQSdm5OjGBM5/RsauK1AhdvB
atbDzcVDveYNcI9B/nFGSstGYbrW9VWMGQHdtHQtK6l/0pljpyM7nSZfyBajAyVYTt+XA1hesu7I608zb0NvZ4kqdq1F8
NeGrRx8iIUe+V2LAIc/VrmdbDcWJ7198RFzcWKG67yCCRy33xOphqSHRpk6RvCZNhhj7A9S3g3/8SJgasJzw0zVN/7eMd
JOF2UYi/H6M+tXvMQ5ZyfDmAdxpOAnxevMtIveQ1AXVg8InLZ6DBcxhMIIQ3OR3pkQYm5Er0ZTmH/zSwrmzYkvesZS1Ge
n8km8xiW7hJMYMRv6xBfPOvv6SNUQq1es3lBMQPqtpVM5bbZ7rLAYmW69MxbB8ejQUjjXDm/oA8sHoGU3YbYc5uPd/OD1
NY+ruaST2fe0f/T+WgY9aK3L/ro7dU6oCQiETppHUaOKbDN4WKbtR1rnbcsVaB5g1F5OqicHEWh/1ERA3bnzHM1J3OHgZ
uiyvm9No2MMrAbxW4jDqOSv4Q4eVL0RMlwim2pb/DkK6rCQAaNbBAdCXmW6pvIqLbxpQGsHlYuim8zLs769+0tX0CXEjh
CzstAZQt3BLJiMLBt9e0VINWmpsn8bxvjvlfUHoL714nTJB93v4H11d1SVrCGidZPvEEL7/AyOjHSdlO3iGoz0nt9IJVM
h8Qk3zJnp0mplnaM1woNViFV2l3LuvtKUFxaJJwk0c3QD/WPqIXFAGVruTwjtGICMc8sgAViOaj7Is1hwY7zx9ppRTJGk
/mT6IMYhxic0eRj851H8RtLtzXTtvVh8as8OGraGja85vS2AFyCJUeccb7E9eaHeCLdbfrnczGy3S0NnOsBnyS1759Z5U
MCtGQEr9rxWBXMKdrKplV6EqERzEQhpjFEOcQ5/8pq3POQ2aE+mIdFJ9f8WBbTn2mfXYP5OnymR/V650gsUiQlur3e3+q
dt+3PJkZIRO2k0bCg5FeMbvGHL0vD2VMngY5GI7bLOOW8/b+JnBDkK6yIXL05jbbc+aKnwwOgh1aZEpsfWFBFScaGbkDA
HEdKLC6YtFkOy9iH1CE7fPAyv53kl6dpbZde1JkyGMPrqKQED6kZLrVMHn+ZWbpoOH2+GdAM/UILZHnnsv48cqSCvEYan
MV1kASSAHIf3hXRy6RTFudGJxAGUo3+CDsnSlGkAHwoaPNoWQc4ecT77gbklcmzNfO4zoX0lS4HWksA7H/Yp2J+J2pDeW
mYo9pOgn+kW4XRfhBt/pl5Q4xaWOW0y4A9rWbFpz6PNuB10QSDoboj3FZMog4tbQRYRnZ38q3Yh7RaBhwGGoO0bgg0Nwk
SkO49W1I7iuRHmpAdkXmat3HFkfJQdTH5mqa+Wgwgv0zK9niDY9wFVchaMAq+m36inb7qhd8kX7G0jpzT/ZsqFGFnhIs7
rTe5EFvyCjauENO2NA8kZu0P2GLu+d0DgpSv5eUxk73bKKqJLn65ItX8IHDA0VHCErOj1BRmGNjqa9wM/SFEhTV2J5fpa
Z9/oOFho6QUxlioyisrvFzuDi+Pr9AxwnLUNPVXN/iZDg5u7yAAAAAAAAAAAAAAAAAAAAAASHTA/
-----END CERTIFICATE-----
```


Listing A.2: Composite Self-Signed Certificate with ML-DSA:44_EC:P-256.


```
-----BEGIN CERTIFICATE-----
MIIPyTCCBfOgAwIBAgIIf/////////8wDQYLYIZIAYb6a1AIAQQwFzEVMBMGA1UEAwwMVGVzdCBSb290IENBMCIYDzIxM
DAwNzIyMTIwMDAwWhgPMjEwMDA3MjIxNDAwMDBaMBcxFTATBgNVBAMMDFRlc3QgUm9vdCBDQTCCBXswDQYLYIZIAYb6a1
AIAQQDggVoAASCBSA/qLfIfCsZp57z2vQozbigPedTSTMRbTZ1R1CKIn1Sqm2oZur7Xq+7FHGW6Uo3T4TBxr3so5Ubd+s
I+exUDPdr9GViVQ+wToAC7NadNP6baoJ1fHnAkfNZ8/RhC8mRlD2nsPlQY5i+vA4TCrHgEFhJm12m/4Q6AHNUxfNmR/RZ
fv1g3pFr16OK/Vb45FdNDvOULgpSTvdQbbZFxeEMIEV87uZ14n4+2ostEIAFZpkL9ppxz6iUPejlp9Jx6pB8G1uxhDola
TFmZw7A0vHGnLrnBuqu13ox3LOfMadRLDeYDZEjXQUnjbTfWttjSc/iedmdIxVGJ8cEHxPGWE39ExO91lYJtto2OwtJlN
wwVIdI724G6WfKusHXAQ3Y8imJx+7xXQPWN7J3TqcGuWlTSLfHO+aiwpWgsNfV4JTVP8k9YwkBB1Zt3VrM6gc79mMAbYd
jkhhO9wWP2OAxNYpfQUhtoIgBZn25CR0wZMcaMbvCDdT7vWFNg9jQqcsdo5+Fsh0hJmymOwQpHAno+VZTv6EWYAaSp9+E
zPcKeNh55yDp818uO1iwyf0EYfIAtoc8XP7jm5XaVyWM58x5uCciqz2y7F4A8khmyu3qy77t5Ki198zu5v+5lBVSlBFOe
0XVC9JlenGDQ56Bt/pVdOXqdyr9UJhne4Z6MBbLoUSysjqPPpBegWFxU4BeF7S1I6K1vii685L10aLQf/Fw/LZ/ShCecD
FQmYqi6/rMnDI6cr9siwQila1YaBt3e1rh58HjY4uX6cYz2qqK2FJLaAKlZDxayntOPnk5dM5ZmBpCMIFX4djfCHOORuz
DzmJ/wY1tqgmH3Z8DeXS5keWslAcVgo9JVf3PMf47OWbYOHaL9l9J9HlDngn9b4c1YIWOtAhEavPulQtLRZF5jHRLPWdY
v4QBa90/ox1r24upPqT98x0U345vXzAufklMYiLGeHNq0fP4nN7vsr+05ksKLgoMdNTibC2qv0OskbgdP5tF74/psXe+H
NmhK/+NWXvKNTATzdih5V9ikAh5ciCqwYYh9+XUd09YxlHY6+K7hVZnsal9o/Xvx1E/iVJinCNczAwXN3InnFFVuchJ1S
WtO6NeUruYfCUIC8TpoqTS90N2Tr7uEeH1mppSmdi2quBOIBDrHVIEq3HPZYqvUfTrwPAT3drmGfpMj53wDgFdCUKZzkZ
bg58O2/lxSg7YnU941d49LWTdW6nLVEBBNKIcA7Gp19PNNg5Uo2pwaYVyky0UmehOd3NJZj0EJn5Ffv6GuUbh04B3Kwb2
vtIOV1Qs94NJJ9IQLFbgD3ioSQPr5N604+50XfjkBBYPkWyC7ukWLr0st9eCwVFNclgBdrV8fz940v3LoEx49YrNC2H+r
5e2jHcrZI8cumCCsvORnfq2liAUe34sjXycz7M4ecLV3ygUmk1gNySXcUQdxLACeojVfFuGghJupiNsG2gNd3uOf99wUj
/MQDUYcqTbrF0rSup9C9O8zt8+EIDd8C8lRStZLfqPmszwQkK7p26PcIo2KY//RWZg1efUmXgja+f5ArioO++HnijRMS4
qif07CmzPgTiVgjenQzV0D4tROexk558AUbn6L6wKADe4CEYxvhBpQjKUpoEf8wix44Dea1Yb+idZrsQaFKkgup/M76J4
AL69D69Sj7fJCXz0FsHorUCTSEWUt3FlHDAuWuP+VHEK3P5i91sw7SWC/B1yhbgBHp2QDCStbl4u3AXBPRLDpCSmUWGdB
EEEjubdI0h7/U/weNg8gg8XSW51eaZh9a/l7DKM/1t5XBtYVllaI2LhvrRG7FaeWgwoCuVcou02OibKX1tvnLXSZzANBg
tghkgBhvprUAgBBAOCCb8ABIIJdJmatfr0BBoKPALeA4icmOSKF+RHEyE/96jo/0TkVmC78bztXni9HGRbUpSrlCV4ngU
Bdv+TPntaqtpWDMf6wc09jDi41cvodLM97bW9aBgJum1uhEOsHQJSNNq9cUbP3Gjiwi8WupjD5aUX62YRvuayixLsqIR4
aLeZeSo525RPZDQkbdmPp06oKX2qVXA00l/7D3Dy2LxVbU2c0whlrPZyM0agqPNFTj2INDJ1nn8c1u70sf8XMM0GlVHox
9iAEz0NHn7KbI781lk/tfNYOHvr2y57xKLSgfZE3uSJoH9DnHb7tYYpIfnlHBorDrnn0BpnVe7zDvKCvsxoZNHYaZAB6p
WtYu6kqm2HaJsLHNiUitbOqynRd4QV7Cj21C2SzDgexYWqeTauL+zVq0aw9oRk9+Gnsrc+CrQ+p+vFlsQx72KU9TVaqb6
nfDj/dGr9/Md5sMhEIipH972BPKdOU1f6YB0cm0sA/SpPtAfoZ8vAeKK67BpLYxfJ7L7ol1AyHztTbu3RqW2Q+sRCOYQU
S3NaZjo0k/QSwAjLgt191MspE+NGTdhAL3WzHdqyOxk0oeTUibLKF92AVrDil9q4ScGfwqW17XnW5euBBvGgNYdIcFa6f
SHVds/FE3EsOat13+NoapQSVl9i3jXa8haCJ9UBDXXhK3DobK/RI7eqFZsnAcUmaLzgSS/G/CnW4XJfTVRTDrJhThCpNX
K4ZNCyewGnRVP2ZkbtOdFCBMmFyotLi4hfAn7tT6dkWGnKdUnXj0QAYmhnB/aiv9dITGUntXs2ywG3arLINuVBVlfk7yS
Ha6rX25xfVo9NaGgamqu+9OSr/CR85H2c15noWYOHlLxFL9km1M8iz6p/0mZAgywql9Je4EofOpOvG1NA05aNuK6mRT+E
8qpXxwF5VGoo10lQnHfEFnb6Z97LHB1nHrJadw9uVixc0s+u1xIX7Rtr881XcCOLUde0xLbAn4Nudi95NJRcMVZWWbRbL
nCXejtEHu2mYLozkwtO960gAqNIfSKkvwEDzc2/zvZ/ut7oLvOpAKVzt3CKksvnmAe8hFyp+DZYDRpQWnp8vbiAjNm1Ut
lmaRWt4orLrP4Mq0sVk8HzquZy4b5/jD6rFBZ6gpHBP80RjGqt8lwfyqLZbG97gA6JTKakCpVNA+K9pdmmr7uVkKJpo0X
CaZxuCG+1QuLpv4HMCVDBmNQqTKv9ozPCxG9M1A0oaSsxrrdurVyLdJOswSTLvMc9vNlnve6mmDzEK52vZRbH3zzJQ5UO
o9XA5f9q8e9akffC2JoifhX0k15GOHLwwOSi318uJkYdg9R20s2ztqhnY5vbcnr9trqq0xtXlhwTmKIk3txkhL6J5Dnph
TDE5rH9awcayrxoR0tW/4V3aQTnwKM8dBb4G/5vw+MovMjZVLga+4eBLLR+ye57Ys0yz8jkjvXlgz57tHB9EkKvtflleJ
JJ4+DHHXLOdy55EzU33I0E053W2GxoBiJcvRvUL2qDTULYyVYShlA+oR2X7kf1KwHscDnDHtDFTqRdU2h9DEMkwpBMXpB
```

```
+UbcrOlWHjg6gAxF+6Am5JyjxfDSzjMl74XOYBRcBjPR47ILvjoKQ4PWHzVdrGTxddzNb6knVYnyRZfjBsJhfKLyvQHOD
8qa9z1hnmMgq0tcCtvI0IwXHE2LapZ8dHgJPuoEvy4RBfvEu4h4GxqEifqagFV/AvgTtDwnJ1Gpg4SqpJnmhM8mhb+Wd9
s9A2wRKV/CR5rwdFSWcrjsKJeWNpn6ZKCS8X2tb1kuGcVo+r+oUqKjJb00ti3Yso7mwQPlISRmRH3xh4ehU0a3xqis4lH
/frMOUS1xmnpwxAUb0l8LECEUPNAroHFGVwAu0BOho72mJYH3MpslktCuTH9+QMke6j+gRPU3MrYSJIjBr9UIv0wKAThtpL1gO3EiyrtVrKCJfcw3uX4UzlS0+qAr91JnsMOUXuKBBvsHiZRC9+oUJX+c6ATDNhXjITu2kdseo4m5skTjBTQGXBKvVJu6qPkYyUUTNwQ46BDnL8Wa8h8CiCHvUsaqE7Nkl8wI7XuBBW5YYCCvcfUxvUhWQ3e7WiaQ+SNsxDqXpirNO5jIvORlznP6JQmNhNtlBkGGErwZh9vaXQuvpz//b7tgfcKZ36Ns3uM6vLrHI1nVGhSTlUDicT5N7t7s7bke1+io2uuo9cw70FlB7BIeoraRx9pKDCKrN475M6YW9Lk3fKxnXCStsnh74dhtVdAPVLbQWRbF2VeNO9Geg2jneMWAD7MsoGCN8SqFAkWaRGqr/8ut749cUZTFh0wFqgx95XMBYOTHZ7etNd1NdWQg3X1sW3glvf90jTaoFWgJEORiQLWfifGEzDHxZ3AbzKg1UupYgadIF4QZ7LDAxmm+ytCjSPUE9oAT/15h5HMq+Xe7xxBseBykpoAZOmnTvTV1cGhc0lJVQ4BCtKY2BdNaZBl4P9xIPv3x1+xDHIc3hITj40e9WhpRo4XF4Dy+modZGVJGjj0Mu4z6fuz/C6H3tEcVQs0zXETW4rW2XnSfC1YpyI/7oqRLVGPsJ/XG8Cwz1+TtWv/V0xPWOYx7vSxsnqyxJNhpMVWbvIwA/N1lpPZYJHSj79w568fnyhvNp7Jazgk8mNA4j4a0RecggzNHJu7OOEYwKwxDbt42Jpq/yrkD1potWMv3llxTOHcPfEECrQ/CjAqu0xH2E/uk5R4q6nW8YkkPxTMjwNa/Dr/zv9aIoSp9SyWPst9/uPCVL3vhNU6i+a+0la4QqaY4yP95wbnOJIavWixZ5lU/98hMCKDRNo6IytEAIYvdcu09hUfaJbXRV8xnw7JWhW8SXANk9Lx5y/Arud16xKXAZUerIYxO9YgMIoXunKsEgvpxwzaa5rsy3Try2Qi7MkgCALbJT9kJfdKN9+Mk6sDUtk/mROUPCPnFT5iKf3q7heJQSFQ7RHYrwI0bXB2/hUmstwwm/O24SmgAcCLSnjgxgL1hbvBTGAXS7HDcELK6MBA0z+Xp0XdjOZKIDO0y04M8dyR1vu76MiftYKlcJE6BEZWvsIK2kRjeFxJRf77AZL7NhDfTzpIHPnhPUw3IMNfeIYBPP1LU9X5ePAA1FBRIZLlVZYm15maartcfqECUnKz5TY2lxgoaMlKmtrsbK7O88Slpfc5CjzdzxGR4iLDdPUWJoj5GnzNXZ7fT1+wAAAAAAAAAAAAAAAAAAAAPIy1AMEQCIE493D4Zat6t0fvlfkdq+wYe9a2laiDUBsNVYNZehtY/AiBJB5v7k6wz/NoIWXzXUsyfQ2+SE7J/wyBp5DZjBO7Cjw
==
-----END CERTIFICATE-----
```

Listing A.3: Hybrid ("Catalyst") Self-Signed Certificate with ML-DSA:44_EC:P-256.

```
-----BEGIN CERTIFICATE-----
MIIQDDCCD7OgAwIBAgIIf/////////8wCgYIKoZIzj0EAwIwFzEVMBMGA1UEAwwMVGVzdCBSb290IENBMCIYDzIxMDAwN
zIyMTIwMDAwWhgPMjEwMDA3MjIxNDAwMDBaMBcxFTATBgNVBAMMDFRlc3QgUm9vdCBDQTBZMBMGByqGSM49AgEGCCqGSM
49AwEHA0IABGjyo1xhsQEMaTjabfIhE7SuKiVpsO12fZzvb9l6HW3qp3MBCtjeDIHBDlNwov0vcb4RgQcb9RY3pDToRVb
R5Wujgg7jMIIO3zCCBT8GA1UdSASCBTYwggUyMAsGCWCGSAFlAwQDEQOCBSEAFXiG7bAaWIQ2gRBmX197MxcUf7Pc43tu
NVSFumPq2wdB/Qd+FiiInGb7c5SDRo3M+ChNTN12kF9oKo2gcej3Bm3TU7+RP7B48P2Tt/j1aPXzKO2RBo6SvNyvYtDyj
X1OMPIOpJIK9TKevUCLonIifcPV8OiXEyxOaySnRLUrKVXA7Nkb6CXipBGUntVRN9VrW54w9CywVoZ6jLk9C/XgIx6uvV
J6t+SCSQ7dJfpvi+qnb6fmAlfmUeeQl125YvZbMskeX1Ht1eHuATGS2xXY9WiNRB9XPzt4mLMdqKxNSDzBjcUERbEucQd
PhAY4w+qLuOT9dqdyGuENj702/DNFH9t8LX1C5LZQt/mrXw3yNbFfQYlYusSSqWPLWg55jaWJ/mhDhgv7f/lGSmyy6Pd1
q9nj5E8Ag1Kvz5csnSsHoa/YWtBHEBdKj2gr3X2LXvFy3ojCoxMEfEUOLC5gI5ym1exshuYhCkMhz420U0VgNxdQbPVk4
WGXLc44FYqzRvXIuBuZjErgyClminVoGXOnn14IEy6skirepqKe5KmYXBe5qfJmNDa1GQPsa3/R/UiOyEdtjluGgSdv1S
IjtNzh3uNeejikLZWxKURejqYxhHFVVWW6NBEuD4BUfx4hKuyQOtgCMRgaY9q9evslHuS8vlzO9AoZNEt/CVQZvLHsSFH
t7k1fs0i6wC2Q7HU2Y9iGdNamWsx25FO7cXhd0AzHJJO2g6P2vY1sUag7a5KYbqfyFGYwys0b7VLAUSj3aZ7LOOz1uuOk
wDHEkGNQzfYwqs5+oOCLmntVXXA6DxYngLX2HfzyKh/9svXDotqpufSPnpVEsIInfF3PruFTgeFcFXDTEgPNmCp3FmMu+
+ZCLVqc1EP52gVEVYJFnpeiolt0iDHaki9twyz/3opMJCXtgHW2AcJmHcfxYoYhhPdZaUKNZMA0rjfWQyozdc57STCQ6G
3yuk6yywe3nVmCr7XLH6GGp5MKkptA2k6XjVuykZBG4BBfXVgvAobwtkedSK46ny3lOgxkaxfuxKVAoeNkX0tCjVtekM+
VK2juwPnXmMp3uGFRzFqZowj6sy+IhuluNbxBOPycGcmRDYqSJT4ZGByVlDCw9bL99QL3sWTVnSz2aZNRZ2uP2vhPLMl
Wi4Lv3czzjp4uRlIuqj4wdOVD2zZCHe8QyjBE7nLxKOGHpjq+zmpr4q8WRigM8kqOwNZ2aa1Aq5OwCkm3HboA91UoL+Hh
rSILVKlNghnzhxHEiLnoiIRP0QEnUk7fUOZjlkZ/se6vBN68UdIOxIdvSehvn9zNUv3Sk7u1Ed3HqRIrUBWfXkrDWe8Nv
ZU79lPKqxTYwNYaVAo6vIeR2J1ScoeCIUdLwA9I2//Da1NpYzAYYFEErXVhsHb9NDT56/e0ehXNrOkGyV2/bk4RXC9BFa
4NUSRZjbOwLMQhDLikz+ll64uIZ1yqcN9u/VU7YZhsdm/9fU/C6fDIf3xh9yYSOdS6f7Ee9xEmJKIKfVxVW/NInQ3nqcy
fTW2EF0fWKL/dYEUCkZAz/NuosLIQY63JRDhqIXrCUs+SCF46TFbCxIzUV+6X5CWfVOm2OomMj+VhPM6zuGuoORH+cl9V
R1VkiVu18CbC8zUFWQ0zvW8DsfUfmEEaPvjTN9g95qTyx7rVzih/lLaOPumpcbWwmjmIadpojcU7rrQOotzPdI4wD0/E9
hJUZl8mQFp6G//Us5b3UgQuYwt60I4Io7rtFXTD+VCNTAUBgNVHUkEDTALBglghkgBZQMEAxEwggmCBgNVHUoEggl5A4I
JdQDtmfcJ+k+K8ga7GCCvxDTTjHxD/0Z+uBSa2Nn/JTNHazl+RcO+Sf9fwmvMqzB9t+yrz+N5r51HViBTnzf4Ok1yEpig
oamWfR6wM99EfuDGuxsl2iHO/K/TgPqUwE45VaWtnllieujpJqhO/tby5CUlnrGt29iXvszoyfX3Bx7ZCi4TZEZXGw2Wq
EyUgMtNl8Tu4J4vmMfAYkvBeMpKS4bpN1Nnl+xwGmiM6Upx5sp0bz0QGXss0yq4a542SuhZYrQOk0Ko9uUForO5EVVr9V
mQiQU38/QqzIN5uJFBL385ZfBc9xvTfGOKIYKNzl14JztmFrRBXy5fPBCeHlLOVIgwoUarlaTCmoqaugUKB4JrSWkvnku
gI2fxmQIQ4ZnHp5BIKusDXRiFmdFQ+AI/UJiDWc9706hMqgsse9WD0kFTL6PSjumRpwH9S8itgtUJ2iEgOVUdWGXo/TRk
ogIoLZ7RA52MF5q0nBcxdXvXzl0QMhf3co/kOkGgmiO3lVNszULz1Bc724E2b6t7uq7SpW2orGeXIFvcXybjbBOQw5rj0
ISMqxCTdgd6ouhBb6+en9O9ET7gGyFbNiHHOF/8eyqIC3XRfg8BUc4nvmkpKOJZ4aOpcnboLnPUUHOykBgT6HBzWALuFz
HUn0vq5gzjC3TPX4SGOXcRS0LzT3wzBvsxKRBetSrPSU2Ha9HPUR34odbvyl7Gjon++SbVbXF7A9qOZ77JepSSf+ApaCC
cBjvAzBaY14m7Pffbvq4iSnLFSpL/M7U72Dp4Wd3S6Z7bdPE/xbmAauqz/UO/Opr747OVnK6onUdCOowYxB1mXMi0xbu0
w1rahsz1b09wMduqxzu25slKIbLAXsEuQ0Qkxd9HoCx9vaZTkrGP0M/+UB5SMNJPze3+JQ+a/2KpR34qhTRdBJgfKhmkR
IN1WjoLwqI8G8Q63/tckQhLHoKt2vwWbbI/ATik+JmDPwb2ASvt0id3jVRlU5dN4McqOd7bQmU38FeyLi2IETmyEOElld
PwHzQN2GAraJrKLBLLD7vZ88Unt71pJ7E0pIo/WFWQznv79d2azlNfK7l3TFztsKYODC2VHJFFoI8lzF1ZsvM7LM72WjU
iJ8wQCm/7dY0tUsMbnTr77gnU7MDiUitZRxpMOBDPlMEYEOCG32KytrJTgAK2Ss0YoZ9uxukS/gmeDv5Tuhqa0aLpecgb
tTGI1wnTm3laAoGmYYCVyO3T/v3p36S+SCXDGHFClJMxX9vVFmekeXYQIixdWKY1OS2XvkQ228bFJB7u1E5kXzCIHfIP6
F3fcXOOn5jc4xg6CpyjMROTKHKmoFudmKYpCDJshV5ckXnHFPSowGd0dCugFKny7QyU7r2faJzDByIXro69g6prW1H3tW
HN+LzBPDiGiPSKKr49uFNYDscg7ZVNP1eGHsmXwmTYau0yV3+9arwN1dOW1LghBWDdFckBsDYGGJWlZ7obxws+BZTom/v
jxu+jJ+cTOduaKGdobOsraP9+rNORnUYJfX3Uwmit5yCbpxpI7PGVDKNOM6vgE0YFIbxgtH5S5bs5wV46PVhMPKfyLieK
ZMu2I+sZmc4JkTwluIaOJxceGGwzHTTw2s+MBFw1jSSmPF+Vv7mXkTPBw6QHJlQnrbYVzSCZ0dvWe6fWYdNSgRx99ZGyq
Nzi1QMiR3z9Aw6mTvyVCaVTnjYEsssgJgETu61QqdOCOr4ONh+fVLPGmTRPnDTzGI9jMicoPlkhvDwM7792m3jrmEZUYz
fhKEYqRZ3mhAq7L5GDywF3j8+E1rvAvt3Ip4lAeN5j3dQKNXC6qwAEfaL9eMoozbA2NZbFkl3KYW/+X9vdu1Mpf/uwdSk
SOn1v/Nozbysmbncujq3ua+Pwd9I27HDZ0j+7c9brl13p50JhsI9E+qCxnF2fSRDQypp3R8poB6V9vSWrlPnUPKfjiHxk
zDA1s37d4W9CnXdl7J8OajpHcxxZp2AUtBJWyY3jo/hXkYl9j46t23Ith9CDSUDfVjekKIjiGILWMHCog9ZzS3aLDXlr5
rFUPyV6+yggCsRA8C2SLybs0aNLQilwllhq/K5EslPBoKIronMV9rYvKpywjscaRDFCsV4SmJDYGqdz+8T1D0MfY9C+Pr
```

```
GUtgbu21ulCb3g0tNeyx83dXmjpRIntzDsnghP80Xwu2BUwHQgUdYZBYAl6SnkaX7D04UEPVWrM35+LDFXR9TZ0qJJd0u
bzXEDLofu576gbd82voMyA8XC3/smwq4jIb7b/52yWffBb4Nd7gKW+LUAarlDY9l1Cqz2jZLb8g8NefmasFAauBLR6DMh
sXuyxGQDeduDhzgVtjfOHi/pIuyaLgrpk/FqKYUr0exUq8XIhXSFm3S3qx2MSzqGxjxQ6BXiwm7bjByVlrPOjl34f5Ozi
YO55jfRKNATeTCGoW8+bjPKxt56Zkt2YHQoSckf2PlYtRt3r973qAvg8NFy5PnvWOwo9zpemHtQSGFzlQncQNRg4W4hHK
V8atw0ruollBZD1OhaRk4KapDoWR95fmQ5v2a80qyJhYiEAChD4e36nY7zxZhEhdiZfcWwa26Cv+sL/WtJeDfI/nUivJC
MIGi/xUoMJ1S+kCfKrGdcExDLYir4eazKlZjhaho8OL+yKs8FY9Q2XF7vF/ckW1QZ5AAd64vIm4M0X5FG7kTlMQquXvSL
ZKdAb7+gC7bg4kNJqZHZIE/yvmls8pO+JqmF6fSz9m71pwEky5hD/DGZlgs3YiKNzTUZcqyHpKBQtlpPM7cIrGk1aCDld
J0Kym9CzkIsoLiAkZQyx6373+xwFPVdQNaJeJ8P0XvQ2Y3GB+plUF4TZxDlb8SXLli4NFeRWaaK0nUtGH+BuJfG57gpGD
k2J7/diw8ajVMNF+I2+uyhM8xsEW89PNHwmg99WE4KIDCa15DboidqyNKUY3BceY1UrOkBClx2O1+axn1BarQh+4v4irw
ACQr9rR1xrN3C9a6lWudmBC9mZdGBxc5BCK/BusVzvq6+WNBoZmF3EBGluFzSWtIh2s8OB0oYJ/Fr78gMPlLA84PgY/SA
HJdv+UgNCCqAbmbTHkKFTCf0tMKRD8QRwl0IZb+98Mpuk4cYPhIZICsyNDdRb8zb6vr/BRYqLjWKn6Klvb/FzN/g/gYMJ
0NWfIKKkaOlwtvt8AOVQ15rb3eStc7v+wAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AOzae4OEmABSGKjfeWQ65f2capwPOTyv6C/IW5An2CtwIgDqutivYWYynCRiMfZl4OJUBIuuhn1qAl+VwfvLpgp4w=
-----END CERTIFICATE-----
```

Listing A.4: Pure ML-DSA:44 acting as Delta Certificate (issued first).

```
-----BEGIN CERTIFICATE-----
MIIPQDCCBbagAwIBAgIIf/////////4wCwYJYIZIAWUDBAMRMB4xHDAaBgNVBAMMEyBUZXN0IFJvb3QgSW5uZXIgQ0EwI
hgPMjEwMDA3MjIxMjAwMDBaGA8yMTAwMDcyMjE0MDAwMFowHjEcMBoGA1UEAwwTIFRlc3QgUm9vdCBJbm5lciBDQTCCBT
IwCwYJYIZIAWUDBAMRA4IFIQCvx99OQZZizta7JphBTSwQov3iUtb6Cfpd2t1z6lJvGlaQCwiaDBk6dnQDEbwIRTbwenh
7i0hY3vFFXtQq7JDhh7OQ0M//oKjJL1zk3tOPAzdmQOqTOChWg5rfqBPT9SG9+uMxemyy0sHzQ0iJ/BIbtfvDuMUQJeio
xmgrXpqM3s4bQQI+x84fP0PfE03mSaYrsb931bBHEayepfadetK/4CJgNLhDRApg/6Itmjr9CbDNF2DsESEBwa/5MW5fu
wfKqtUafVGUY+b83zvur0/qNcIrmOK+kTafwiN6c//466vuv/hkNZyZUVCTEgWvh0/RXnNsJI4M3lSl1yazAh9dgkjFrO
Mm5muhD4iV9jf4vcNS5BAtiswosiZpklcBEMpTGE9h9cAPZ6hOhH3Yo3wfDhpA2QivXUivXOchgupxBugM22i9vn2ZDs0
2wEx9FrZ8sbCAPjDNbV8a5qSfMghg0dg7aa9DDbUH3DKKOQdXVz3xMRvm4HNZvAJRSvfHfRaplmDKuGgwBKTLjMZn/0ex
hLzNxu+qedYVy7FGR0P9zh3QI1si+c2VYVo7kDDACO35OEllbYEzgBGpgx8d25SvzKNmk2uvA/Bin5wHtoYIOj0zuZFDp
Y+k/puvZlJOzHE43biM6RuFogE2Tu6RrZXZNZ55qkcYF9i83cOnqk0f0Z6zh8bKqCoEzKZZZiy22C31outjONWVXm7jmC
HA/WbQxr3gq78HYfjJ8phfDGQWjQDVi9Dl2d+TL5chw1gx3FkKQGTwXaCfnjpSWLGaqiwj/pbWq7qJyCOfQXpRiFDTBJs
pLTwEriS/pVGGnjwTIE1Rq5AlW8bsSKOOMjtK+r0AyBd/R/YzV4EVlfPb62jCql7qVcbez1Y3gf4i5X9fq1Xg4LHt+HKg
ABw19j8YzdEJRxDSKjHN14hfVMn5+qYEwZqfIQDSAKAUp5kXMdyUCvh9a54iKqEPs2+VcXt9u8AgWrpXZ8POFUst0eiQT
yUODTp4Ahtwx9l4F/qa0CbObbuab7C0b3kYqIZdgcF+AJB84aKLEedwBBE9fUF5BFRtGIjMONNgNrPAb3xYc89rPMxx48
HvmMug5YFIFHSYH2e/K3nyImqeOZY6zI37+ezEs8N4eAgk007LyVTqbEbrdoa+/cpLWtLekdsZ5Ob/iLfwzVaIQMsekhV
W1NR+drGjxNkbxLdysfjDPHrbxmhmHU3um/ENg8H+FIGvEgieD+ADO4v1wCIlNiqCacM53oyjAFNZqH+QhW+G+5oGC3eK
hoWLw9JOifwMKboPH9e7Fg2uhxxXFux/Qt1scIev7ZNxzvMpYS97Wwf+LS3Te5DBJOQGfpoy+pPo/8h2MEvrOl73ua2IG
Nx8I3OsBB5qdoqS/gx/l1U9mlLnW0wqFbJ291pG0hBTREQo3sTbgpdg557LtNFunJsm6g8NU2coCp9vFrYi1xEEa8LGwP
prhJWvCJS9ziAI83MXoCJO/HgUTO8g8fyKqcq2ccashH5xqKhJapewfcOut7XYh3QxD2npe4hi4b8cSonKi+ZrVlTHpOp
F2AeLriaB4UF2MowI+MPdrk/4CNm3xQKMqCq7MSFEKhcMQr1MufVwrpkEo45muav/3QEx6d7X9cYPgeZJMKwTkQmcs9Ug
UbKwXdfK5KHVE9OdQsozNZqPwouSNNIWxqDGkE9VjuZK6DDnbAW0MI5tZ9+7fWAvJ9oQET9JNTJHOCPOVLjRVhJ3cPC7b
7zwogMxMAsGCWCGSAFlAwQDEQOCCXUAAL6bvIV7ZgZe5i/snKZ60y1RxyRhiijYiKYCru5DhKfkUVVSPqf9IoVjR8OR8P
CZ4YpEAtyem0MReO3gPX5FmSGX6c7wSS/7tqudMI8GaQM1g9AiJvrKALqp6kph6VCsHcporwnl6421lTfPSO2Nce7d/Af3
1cEIrSTkb7Ga5t3akdGOnJ/LIRiC1Y4gKd62zf/OvZi4W75Rmk5R2yqsbnmkmhLLSlbP/oip7E1dUCHIXkeig3zXo57F0
qJ5REmNuKWLkUl01sR3hl0WTflHXxG7J67RNRwC7haNRV9x0zY9s0gS4jOsjQA/k833Sb4kO8N619dZZi26Jx8yaYRQcG
fhp4r93TTp4tikg1PtSHp4ZxkG0xlLyX1LZh7pb8io7x1cSUZ6M75mZxw369PrMGIf6BrP9SmABLy8pjt7UFOQpa3z6MJ
fT5gTtzmeKK3MzV7Zd1winzYGd1jMntS9fQnWdl/HvQgz3LIl9emTKOARJpj5ojxamV2hVGtOwhMp0kc84EvQcKsq1mZK
f27RC6PLbcKXu10QXyshJfibmeQvbYAcX+ixcU10/u0kEUuFpF/ZNxBvWpX7klrUzMU8R2f9W3hlK35ImLS+Qk3ZciDKp
Afyy4GTgK9dfY965eX/B8bIcLqWr/2nVaxxg0T1XZ+niBj0BHKZff/A5dSAKFXIsW7+GMbN46edB2t8BjPI9s2OAfsJi3
SwJph8u89ZoTryeaELiX+XOOZKslxMDXfRaXh8NC1d8mRCo1gFkKYXxNhUxZmUSAmgrDUGGQXTMP8AhonyWpZMf93z5Up
XTEdGMVWJVcdkMgCG4Nj4hdirqGqh101yoUAGHRHpkWCYb2FeF2HapXPRpmbVdszox2/G4E/BTfjLe0AJuL35tU8e90MS
5ceHxS5/bBdW3IsiDzFTZoHW2QkgybFOzcVoku7iWExgqEZ3LRXpAbQFTpjuJHoCxuibvebfWwJXkLOhd6MpQCOHaXf+T
FkeY8jGscgcZYvJ3lA5UBgcarzbLJ/ifTF6z8xv7kt5ENpHBsAbPfMp/QJx+ZSAKhcq1o0b/Wn26De/czq91hnEr9mdAa
VSE9kwWKVAm/x0RVkkfON6h7KeZo/dGehkW2P92H4oMJ+u0w5BXkJV71haQDpNEieRUJXO8dbi5WZGF6yJUWpcgx8r66M
w/5c0GpRF1GjECyaWMNg8iv2+x3sqJoKDJqrAp1KicCo7dL4VADtxOx4m0qMKgwU1DbRemtefmrwgNwd2Hz3dPFqXB+zI
FIUkVdr8ZWALMVjpHYh1CyVC1injKBSJfvpSP4G7Mo9EO2COVAZkDSoe1SFJHHZ3/MPS5JEFLwBUxtXWOpUryWXHtVFed
6YrIEg3XlZ4BxcYF6PQJr/JxLbEKWdh38OKh27yql3xoMYJK+8dSpKKZFO+9RxKJy+I+DMdfhOGFtLyLw/IkhY2GGT/MA
pLKyUDaqakIq4Wh6zDrgwkAlowtX+UzFnP5dS3+QSQ2cdnGO1Wa77lRbbXfnY9XlnUpEzROfqvObei5nAsaO8yYaChWYZ
yEQTpb2LTdvdM+VGzTHfArhTbUdoEVb5LFBR9a81v77ARId5p4LzgmlSJNjdXeEYuehsyyaiAgE/Fq3FQy+wyABVej/po
rKmwJEx7ESUR9QLLopAn81qq5bBaIjlpjF5fwnQT6xJQ7ua95YojlBtzBJ+m/T+EkgrK9tWVMuW04upohaQmtZDdNzye+
46L4Lg0aJkOfPLPhAd8unhbi+lXXTxLewNTdubiK9g7vcdW30WI7uT2smoD/RDuLbiDMU7RRG7Y/6fCRkKpCwLepua38+
tkseuHO45bFBsscP+zXNwL38hzqoUrKGUmNcCLi9BA3/ST6rJPdo73eIrT9KZcPIURp5wOpBj9c/novsbmixqe/DIltvx
QTi/T+Jz+eKPpqXINuhE11iDoIlG3RvX5rILkSnCz70emIFa3XXO2rY/Df+mrG90AhIesEVtKRbuMO/6VUFcgtSGEww0O
BajZbijq/AM04lO/yuIngKh4P9W4X7WZvdC+VSixFnuCbP+mFBYrkLAOx2x4hYy8VYNSOFdgb2u8gfAbCvzoxR3zdB51d
9bbysZDzNllG29HwogDdYqZjl2tGPjAvkubtu7Ds5WZPC9NRthH8Rngw3UCtigHBU4kVoev64ExP9/baDC25CmWQEkVy9
c/6h0MIWQd7zsWFjOJ3mvRnLGBXLjKfo8kmR2UJApB6sJ/A8VsJZwHWxHUi/DLMj1BkowWzR1YqWRuV1UBJhd2LSMD9Pu
JavRZatdYl9l1chAiU62DO277mpPMrQmMEHZOptOmPk31qOg0bmT3iYMDPpRWHcnw8cMaO7Q+UOOZEr11h8obw1xj5V7j
E4CKM6AD7Kwm3SB/qXf/bkjPjITs68/qKQ82SQlXzSS37QpJthbja0kjjpkZox4E4uw0qnpsvNEugSyOidjYu2kUXp8A2
/1v82xn0xwJ8SvTDb2Wmf0V1Ps+1qvMW646CGvv7HRAAlEXR9IT+8eWz42t8N8zGvtz1OsUOseYVpnGGdZPINQis9oSyg
ACoul1qZU+sy4B6teaOJHMO7FqLi8iURWot/S00q5npBYjaRkEhYnLci9jsBjYl4idkljedqjPfWltHDZnmqr3PLIqfCZ
RcWFckAF0AZEh7CW4HIPDUuGBhEnjT/DbwpjyCGkzpx9Sdt5oKS60dfVqx+VVCboaTZpXglARv6W5SINzsNYEEQMcqwkP
MiCyb/+FoUSJW+3KN4Kf2+NqVOEokcs5fs3Cyg4WlmXbTD8dVeHVf5QfGZe1V6QEWQwEabMCT+tRlBVOHMimuw+odAKyX
V9PyC/pBsJCC0i5ASJMUhOCxOVZwMKd2Zb1kOJNh/txQk0ACTnbV3nT0Rx2b5o7fgJg8ezFRzzNEblmA8JRJuIMYHuc/h
```

```
td/euCuJ37h/Z2OKDX7ZVA2lNhc8cTjOcr7EKrEaqG3a7wCtdqjNSprPjvv9+mhp9iAMUbQsSbGefujN7Fgv+0ParlTGo
7Np8Jn0Fl1KPjgryJApDNKFvUhgRPUv1Uh674OceZreaPdY8bpfQCeQ3M1Z+/bUuRqDrc9G4h5e1YAHyZCR5ed1+PzByl
JXHiTqKzJys/k8ys0NkdhYnaQlaXS2unqDCQvO0tnaoOrubvLzO3u8/oAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAoX
JTY=
-----END CERTIFICATE-----
```

Listing A.5: Hybrid Chameleon Certificate with ML-DSA:44_EC:P-256 containing Delta Certificate Descriptor extension (issued second).

```
-----BEGIN CERTIFICATE-----
MIIQXTCCEASgAwIBAgIIf/////////8wCgYIKoZIzj0EAwIwHjEcMBoGA1UEAwwTIFRlc3QgUm9vdCBPdXRlciBDQTAiG
A8yMTAwMDcyMjEyMDAwMFoYDzIxMDAwNzIyMTQwMDAwWjAeMRwwGgYDVQQDDBMgVGVzdCBSb290IE91dGVyIENBMFkwEw
YHKoZIzj0CAQYIKoZIzj0DAQcDQgAEQiVI+I+3gv+17KN0RFLHKh5Vj71vc75eSOkyMsxFxbFsTNEMTLjVuKFxOelIgsi
ZJXKZNCX0FBmrfpCkKklCcqOCDyYwgg8iMIIPHgYKYIZIAYb6a1AGAQSCDw4wgg8KAgh//////////qALBglghkgBZQME
AxGhIDAeMRwwGgYDVQQDDBMgVGVzdCBSb290IElubmVyIENBoyAwHjEcMBoGA1UEAwwTIFRlc3QgUm9vdCBJbm5lciBDQ
TCCBTIwCwYJYIZIAWUDBAMRA4IFIQCvx99OQZZizta7JphBTSwQov3iUtb6Cfpd2t1z6lJvGlaQCwiaDBk6dnQDEbwIRT
bwenh7i0hY3vFFXtQq7JDhh7OQ0M//oKjJLlzk3tOPAzdmQOqTOChWg5rfqBPT9SG9+uMxemyy0sHzQ0iJ/BIbtfvDuMU
QJeioxmgrXpqM3s4bQQI+x84fP0PfEO3mSaYrsb931bBHEayepfadetK/4CJgNLhDRApg/6Itmjr9CbDNF2DsESEBwa/5
MW5fuwfKqtUafVGUY+b83zvur0/qNcIrmOK+kTafwiN6c//466vuv/hkNZyZUVCTEgWvh0/RXnNsJI4M3lSl1yazAh9dg
kjFrOMm5muhD4iV9jf4vcNS5BAtiswosiZpklcBEMpTGE9h9cAPZ6hOhH3Yo3wfDhpA2QivXUivXOchgupxBugM22i9vn
2ZDs02wEx9FrZ8sbCAPjDNbV8a5qSfMghg0dg7aa9DDbUH3DKKOQdXVz3xMRvm4HNZvAJRSvfHfRaplmDkuGgwBKTLjMZ
n/0exhLzNxu+qedYVy7FGR0P9zh3QI1si+c2VYVo7kDDACO35OEllbYEzgBGpgx8d25SvzKNmk2uvA/Bin5wHtoYIOj0z
uZFDpY+k/puvZlJOzHE43biM6RuFogE2Tu6RrZXZNZ55qkcYF9i83cOnqk0f0Z6zh8bKqCoEzKZZZiy22C31outjONWVX
m7jmCHA/WbQxr3gq78HYfjJ8phfDGQWjQDVi9Dl2d+TL5chw1gx3FkKQGTwXaCfnjpSWLGaqiwj/pbWq7qJyCOfQXpRiF
DTBJspLTwEriS/pVGGnjwTIE1Rq5AlW8bsSKOOMjtK+r0AyBd/R/YzV4EVlfPb62jCql7qVcbez1Y3gf4i5X9fq1Xg4LH
t+HKgABw19j8YzdEJRxDSKjHN14hfVMn5+qYEwZqfIQDSAKAUp5kXMdyUCvh9a54iKqEPs2+VcXt9u8AgWrpXZ8POFUst
0eiQTyUODTp4Ahtwx9l4F/qa0CbObbuab7C0b3kYqIZdgcF+AJB84aKLEedwBBE9fUF5BFRtGIjMONNgNrPAb3xYc89rP
Mxx48HvmMug5YFIFHSYH2e/K3nyImqeOZY6zI37+ezEs8N4eAgk007LyVTqbEbrdoa+/cpLWtLekdsZ5Ob/iLfwzVaIQM
sekhVW1NR+drGjxNkbxLdysfjDPHrbxmhmHU3um/ENg8H+FIGvEgieD+ADO4v1wCIlNiqCacM53oyjAFNZqH+QhW+G+5o
GC3eKhoWLw9JOifwMKboPH9e7Fg2uhxxXFux/Qt1scIev7ZNxzvMpYS97Wwf+LS3Te5DBJ0QGfpoy+pPo/8h2MEvrOl73
ua2IGNx8I3OsBB5qdoqS/gx/l1U9mlLnW0wqFbJ291pG0hBTREQo3sTbgpdg557LtNFunJsm6g8NU2coCp9vFrYi1xEEa
8LGwPprhJWvCJS9ziAI83MXoCJO/HgUTO8g8fyKqcq2ccashH5xqKhJapewfcOut7XYh3QxD2npe4hi4b8cSonKi+ZrVl
THpOpF2AeLriaB4UF2MowI+MPdrk/4CNm3xQKMqCq7MSFEKhcMQr1MufVwrpkEo45muav/3QEx6d7X9cYPgeZJMKwTkQm
cs9UgUbKwXdfK5KHVE9OdQsozNZqPwouSNNIWxqDGkE9VjuZK6DDnbAW0MI5tZ9+7fWAvJ9oQET9JNTJHOCPOVLjRVhJ3
cPC7b7zwogMxA4IJdQAAvpu8hXtmBl7mL+ycpnrTLVHHJGGKKNiIpgKu7kOEp+RRVVI+p/0ihWNHw5Hw8JnhikQC3J6bQ
xF47eA9fkWZIZfpzvBJL/u2q50wjwZpAzWD0CIm+soAuqnqSmHpUKwdymivCeXrjaVN89I7Y1x7t38B/fVwQitJORvsZr
m3dqR0Y6cn8shGILVjiAp3rbN/869mLhbvlGaTlHbKqxueaSaEstKVs/+iKnsSV1QIcheR6KDfNejnsXSonlESY24pYuR
SXTWxHeGXRZN+UdfEbsnrtE1HALuFo1FX3HTNj2zSBLiM6yNAD+TzfdJviQ7w3rX11lmLbonHzJphFBwZ+Gniv3dNOni2
KSDU+1IenhnGQbTGUvJfUtmHulvyKjvHVxJRnozvmZnHDfr0+swYh/oGs/1KYAEvLymO3tQXRClrfPowl9PmBO3OZ4orc
zNXtl3XCKfNgZ3WMye1L19CdZ2X8e9CDPcsiX16ZMrQBEmmPmiPFqZXaFUa07CEynSRzzgS9BwqyrWZkp/btELo8ttwpe
7XRBfKyEl+JuZ5C9tgBxf6LFxTXT+7SQRS4WkX9k3EG9alfuSWtTMxTxHZ/1beGUrfkiYtL5CTdlyIMqkB/LLgZOAr119
j3rl5f8Hxshwupav/adVrHGDRPVdn6eIGPQEcpl9/8Dl1IAoVcixbv4Yxs3jp50Ha3wGM8j2zbQB+wmLdLAmmHy7z1mhO
vJ5oQuJf5c45kqyXEwNd9FpeHw0LV3yZEKjWAWQphfE2FTFmZRICaCsNQYZBdMw/wCGifJalkx/3fPlSldMR0YxVYlVx2
QyAIbg2PiF2KuoaqHXTXKhQAYdEemRYJhvYV4XYdqlc9GmZtV2zOjHb8bgT8FN+Mt7QAm4vfm1Tx73QxLlx4fFLn9sF1b
ciyIPMVNmgdbZCSDJsU7NxWiS7uJYTGCoRnctFekBtAVOmO4kegLG6Ju95t9bAleQvSF3oylALQdpd/5MWR5jyMaxyBxl
i8neUDlQGBxqvNssn+J9MXrPzG/uS3kQ2kcGwBs98yn9AnH5lIAqFyrWjRv9afboN79zOr3WGcSv2ZOBpVIT2TBYpUCb/
HRFWSR843qHsp5mj90Z6GRbY/3Yfigwn67TDkFeQlXvWFpAOk0SJ5FQlfTx1uLlZkYXrIlRalyDHyvrozD/lzQalEXUaM
QLJpYw2DyK/b7HeyomgoMmqsCnUqJwKjt0vhUAO3E7HibSowqDBTUNtF6a15+avCA3B3YfPd08WpcH7MgUhSRV2vxlYAs
xWOkdiHULJULWKeMoFIl++lI/gbsyj0TTYI5UBmQNKh7VIUkcdnf8w9LkkQUvAFTG1dY6lSvJZce1UV53pisgSDdeVngH
FxgXo9Amv8nEtsQpZ2Hfw4qHbvKqXfGgxgkr7x1KkopkU771HEonL4j4Mx1+HQYW0vIvD8iSFjYYZP8wCksrJQNqpqQir
haHrMOuDCQCWjC1f5TMWc/l1Lf5BJDZx2cY7VZrvuVFttd+dj1eWdSkTNHR+q85t6LmcCxo7zJhoKFZhnIRBOlvYtN290
z5UbNMd8CuFNtR2gRVvksUFH1rzW/vsBEh3mngvOCaVIk2N1d4Ri56GzLJqICAT8WrcVDL7DIAFV6P+misqbAkTHsRJRH
1AsuikCfzWqrlsFoiOWmMXl/CdBPrElDu5r3liiOUG3MEn6b9P4SSCsr21ZUy5bTi6miFpCa1kNO3PJ77jovguDRomQ58
8s+EB3y6eFuL6VddPEt7A1N25uIr2Du9x1bfRYju5PayagP9EO4tuIMxTtFEbtj/p8JGQqkLAt6m5rfz62Sx64c7jlsUG
yxw/7Nc3AvfyHOqhSsoZSY1wIuL0EDf9JPqsk92jvd4itP0plw8hRGnnA6kGP1z+ei+xuaLGp78MiW2/FBOL9P4nP54o+
mpcg26ETXWIOgiUbdG9fmsguRKcLPvR6YgVrddc7atj8N/6asb3QCEh6wRW0pFu4w7/pVQVyC1IYTDDQ4FqNluKOr8AzT
iU7/K4ieAqHg/1bhftZm90L5VKLEWe4Js/6YUFiuQsA7HbHiFjLxVg1I4V2Bva7yB8BsK/OjFHfN0HnV31tvKxkPM2WUb
b0fCiAN1ipmOXa0Y+MC+S5u27sOzlZk8L01G2EfxGeDDdQK2KAcFTiRWh6/rgTE/39toMLbkKZZASRXL1z/qHQwhZB3vO
xYWM4nea9GcsYFcuMp+jySZHZQkCkHqwn8DxWwlnAdbEdSL8MsyPUGSjBbNHVipZG5XVQEmF3YtIwP0+4lq9Flq11iX2X
VyECJTrYM7bvuak8ytCYwQdk6m06Y+TfWo6DRuZPeJgwM+lFYdyfDxwxo7tD5Q7RkSvXWHyhvDXGPlXuMTgIozoAPsrCb
dIH+pd/9uSM+MhOzrz+opDzZJCVfNJLftCkm2FuNrSSOOmRmjHgTi7DSqemy80S6BLI6J2Ni7aRRenwDb/W/zbGfTHAnx
K9MNvZaZ/RXU+z7Wq8xbrjoIa+/sdEACURdHOhP7x5bPja3w3zMa+3PU6xQ6x5hWmcYZ1k8g1CKz2hLKAAKi6XWplT6zL
gHq15o4kcw7sWouLyJRFai39LTSrmekFiNpGQSFictyL2OwGNiXiJ2SWN52qM99aW0cNmeaqvc8sip8JlFxYVyQAXQBkS
HsJbgcg8NS4YGESeNP8NvCmPIIaTOnH1J23mgpLrR19WrH5VUJuhpNmleCUBG/pblIg3Ow1gQRAxyrCQ8yILJv/4WhRIl
b7co3gp/b42pU4SiRyzl+zcLKDhaWZdtMPx1V4dV/lB8Zl7VXpARZDARpswJP61GUFU4cyKa7D6hOArJdXO/IL+kGwkIL
SLkBIkxSE4LE5VnAwp3ZlvWQ4k2H+3FCTQAJOdtXedPRHHZvmjt+AmDx7MVHPMORuWYDwlEm4gxge5z+G13964K4nfuH9
nY4oNftlUDaU2FzxxOM5yvsQqsRqobdrvAK12qM1Kms+O+/36aGn2IAxRtCxJsZ5+6M3sWC/7Q9quVMajs2nwmfQWXUo+
OCvIkCkMOoW9SGBE9S/VSHrvg5x5mt5o91jxul9AJ5DczVn79tS5GoOtz0biHl7VgAfJkJHl53X4/MHKUlceJOorMnKz+
TzKzQ2R2FidpCVpdLa6eoMJC87S2dqg6u5u8vM7e7z+gAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAChclNjAKBggqhkj
OPQQDAgNHADBEAiBYe69MQiUJgg3Yg3ylVt+m2oT/ZzyBEsJqpW/8rgXMrwIgP17FcQAWJupC+maHisIMiG/l4l/a1Uy2
KsueU45h9c4=
-----END CERTIFICATE-----
```

```
-----END CERTIFICATE-----
```

Listing A.6: Certificate with ML-DSA:44 with Related Certificate extension (issued second). The contents of the certificate were manually modified. Thus, the signature will not verify.

```
-----BEGIN CERTIFICATE-----
MIIPaDCCBd6gAwIBAgIIf/////////8wCwYJYIZIAWUDBAMRMBgxFjAUBgNVBAMMDSBUZXN0IFJvb3QgQ0EwIhgPMjEwM
DA3MjIxMjAwMDBaGA8yMTAwMDcyMjE0MDAwMFowGDEWMBQGA1UEAwwNIFRlc3QgUm9vdCBDQTCCBTIwCwYJYIZIAWUDBA
MRA4IFIQBWh5Y28JmWczCYYySxkYrUjsSV9ULj/ceCL16lyXFi8HYCOadmgQeNhcGf9uBb/YnIzYfOeti4ynFl/tq2WMP
UM6t1GqQP1ZqyIU9PO8I3BhsFTIFqpycAbSUozbArfMcWDBdMkDKGZddi4eJPU5sVw2m8IVd866IlVM9NkYDte+7iR+w4
EkOZyPxIleFKJ9KRUPvNmyBJHOIf5cQlH6BQgPp3OY1bXts/YznlcDjtdRdH3a8sJ7I4IilwSO+rnjOkESZDVF6qfsgpy
84XvxGqC5QS2jL2Ufy5qrANu83prdPGmzNXzJ8REyFUislVGfq1zyDarHg810kgepiS8WOCBaQqos/woSdIF77Yhifa5i
SW8H41BOr1rm478/vca89g1iL1clRbVRfPj6xdraOiaqpQTHtFHgZxkV7jCSCuZNFiofmE1EjWvVbgqtesIZQtTcye2KC
LOvn1zhurmjfpdlW5hXR+ID4rcSYfFLnZz0oHVSPsq1bWAWSk03tNOc5Bw7DJ0tY8OZr7SYdIjmPkdQ/TAOYZxIwhLlnS
soKfcGbj3jzXIY7bk1iZfOhV4bYOcJ4bUZQeSi3+74JHE17Y7+mfPrEvwTa4YwFRFONJsHjgWWoQhPDOwAiUtHsLtVM6K
fldqcGFGmI5dg3ZrMgffCRx4ieOvHYXmLhbUjywYk4p/uIxF0dPIqHjw6ei4lG1mWWJxcGcxhOqIN8bW9j/iHesJ7/dxz
3LR9dOffS8OET52q1movCTtanRAodPJo9i7X0x02947fdULaKq1018JqUBUvKxXJYhEHJ7qJLoDSwTXDXL40xftJI+174
QoWyDgWPCJ9BbJlKOIAPKxA/sNW5jvUF5KX4/zg5siwGLzJ1ljz3tAoICeUJOyOPVlEU/Oh8BvDniC1zi3uKvTowIOT+l
fHoazS6GnztaOm4U3CvH4LKJiv4TsUx7a6BI8opfRXtMZ7A8hsepuC8fyHU8HBI10nI362BlfDID6JzHDwsRkDsfWlXTt
p48sTGQzEakBYrwjrf0793/JHExzwN4ud4PivqHimCIKgsliFP2B4+ahww2X4lBmNZia/qhxyjSBk1WJUx73UNLzh5Vii
JmLg6VRessEBQjxo3uUHqVT8DM5KIW9LiNy5c9PZY0+h31qyi9vtfS6PoONclxGnqQui9dnDkL3qqEu0aPDxIUFrfibrT
TqVHgjMrTxsYVGo2j5n7RB2HBXAKsYGiBhjdvZrDsVIv1gstC5/jAjYDQMuFml/bKevCsJBnLo6sQdI58ocBWOk7avCrr
Hsqar7q+rbEPz8I2Mqv2oApg1oCU5b0vBg7Axa1vEisNup5/boU/SW+9BOpeAStltoU/AKWt582qKkGqFfe3Oljl/CbHg
L3YtFWe3HXBW83Bm8MvJ3Nx2VAH6QdHKUIF1w+dJTnUowGNM7fozRJyC1vEDfWYnHh1ZuxqxNkgMGdoD/nRzuDPUNhN8w
E8fxBSufOfEybEagHCdQjlbMUnXw0bwu5kJF/YAN4n6EPCzYK4O/OTtjNOm/XQYFFRONuuP9cEtd+6E8ZX3q3qccl4tqo
zK2BPj3YUvbDN4WtWcqefEP1CMrWLYAD+tZ6wKY9q7OiGQNoJimbe33ZU77q9rprVAafuiXxc1wMEfzGgQB9iDMrGZYXh
RwwHIELrtrCNLLoBI9kw41CuFO7lXG8rVo2fvPus4aSFPkAFZnMi37ZKigU55+R31aMA6YkBPcBsDkZ+kJGzozIwMDAuB
goGCCsGAQUFBwEaBCCwOV5Z95FM+o8jfI35xfUahc2M6vTSsknn6dlIL77HAzALBglghkgBZQMEAxEDggl1AExf8GWkYJ
n7X3sSrs1BMS4jZPa3ORzUU5xRqsWPpfUaVgs/rY/ZKbJV7r/Vu4zjGkhlDAOMqAcMCeFQWD6TCz/FHcRd/JGm35MkNwD
cIzh4M6s0dJfH2y5KoeL5sbEmV8dhTsyJVz0J+7u4OMFPJYGQBssnuV//iRjmy1qagNK7t/fDyc1gTnvoZJAUuBOtsseI
tTQyxdaZy01HKdjR9xWCta9t+w1ftJwfE4laRL3TxfFxXjj2awd2i3/m18OS2gY7IjbD1kQR8Jm/Xr3V85nWOJRpxLPDi
dT5cxzQkLi6hkbAH7/36cJIKhRuBSOYKtYHYskmbAlOqTQ98oHSXy6hLm6vogzLxTvGJjYYW2pbtKg/6dwaKBUCtSb44g
NQMuOrFfVB+NATJeXjSvPMQp1yGVLcNf3vs5lDkgph9W5M/ulRKfoObq9rL1+ftv5ztRHVRKgWGN5aWHzYT2oEjO9UreT
BJNFJReZ/s9NkI8ezBGgEqSUxy+dRQlIQ6AIL+LSu+9FsXMwakVv/MDAnAk9dLJyFa1iSZEH8x6ABULMlwqVDPUCaDB1W
WvykgNfh+U6hUkDOSIOtYX834P7pFE6iWhnKcmXSU8Xo8MOqmX+lmXnFg+NMqFrWCHGeN1UXbdh8Nj3kcFb5JJEW+UBlL
5wFJUuGXgbJw7zgCONbu8suFxdamOQos3mnw8gukIOJJAEvUjRRir4EDJD9PjZSfMMJ1of66ZiVOqCzX/+dK8Jpw0uO5B
37FZuQs2e+wvOHnTMH2gUTixYfGEDzgdELQQ1rhEFERxdvjgz8lFxSxnH9GuN/pXw2iDvtj26LeDeiAvsjpRfKpLYqoAx
gLIxjz3jE1jeSioBDLAHZfB7I+hD4OJMIJDJ6UgPYbX53Mv7gUKo9pN1rTDveFxyOG0M7WsNf3XL6O/G8LPXkIiw6Uni8
qT2lPPK01TH5Ss+J7UP37/DHzMHY5STjZcOYwB5TZEwNOYKUPEj364tpnVIs96NAJMjvVxIo/SdYtxWqvVoCoXPaBo2LM
T1YSO9UIgKY+3HL7v76nwbGf+2aZ616p795a1u+TvexccYaRRzjWGJ22kfMpwA/vxZ7URfUjN4TSuriJY4VQKwXKZVVnR
TUPdj8xflOwE2FMqyL1SV6Artnjy1BXKF7Qt92vCELyfkAnzxSMvTcO5fz2rRMLNmx5ebCg7ZVcIrzSi4D5IFAJbSwwaX
baKVL0nEdfGWyuYa/G8gSYxThfYWDDVXb9jUnUkqi/O/vXhsLYHjRqOlUkIFBO0+cCkdiPB9avJEeYxo7ItXO+Z0zkNOb
KLk4QmtGN7OJl10Out636bO2KL3ngb94XKFtfYNPHPJZRPMa4gI/1MAkatvf2nDZd3hvyP4+BprnYf2mrKfRZBqunPGMJ
0Np9oCjCUN/n+EVuGJ5rYzasbZpC/uvf8/ChJefRJV18uOZcm/Q7G8ob1UCnwQPuLDDoIPg0VHZc7rGL+cu2bmpNLBeVN
A8DDwd0lwq38e/HnGO8YaoWWRpb3CZOhb0QaI0VEV6489sxR6ZTFhLel2663Dh7tgg/Cl0iJidFkyhCRMhzBh5ook+Mkr
63LcKpLwPehIVME48HuG/sIC1X1xx/VPF1o3fbDONtLnkJQVzHBcug2p1ff7piI3gegm/ph2RyVKS73dak9xHwgowDoFa
t7OJGP27OqlpDAJgakzPQM4cCpDJvzNpTNNquHne8gbDHvA+NB2FRLFgX19HWUyHrQb7J4W7NGzyo9si2hUkNVNFORIdT
sCdGcJsXeXT4YsXdKd1bc2p8MjLREw9P74t/iw8K+/7sys8LPqYDJNWG4ux2msDq0mIyjIFPWEcVcxbcqcnhr/eLpCiQj
7ytHDlVdP54GXEU876eCJGNDk1XTCK61uVaq08BW9vEStYSCgff9z9pZXoyLvLz1Hpxb7E+J+KaZJbNfnx6/ApkQ/Etl5
SbBlA6Ne+sJ+nX5JQZf1Jb0MYohgU96/VMGOzW/Dv0HXXqhfdis7Buar6B2KEdwyUo2C9CdF9YHTJtdlzfv7uZGneooyk
K/sJbwn8hrpbSoGJ3Jwj1mFCHTZB2xUejat6V42DN8vGFlOG8t3GtvbUCO93F4c6XFrxTnVWhDNu4sMziS1xHhfLZQ5sx
kY9TFR6vvfO2kRyAmRNTJMpIYxo/YzkqUZHfORz3EEntpRHx4jVMxQXq68Q139/8SZ9BEGZrcXCl7xdMIYy4D+A9Z14kE
SsxZj0Vec/i9aEJOTagQ/llFZpvP90vu5MOCJdlholuIukVIMEGylPFCnZINEhDEPr8JejGNOVuTjLxXN2/0b1cGz79HL
GvbepZnwCPDQxhJXelKigovgL+M8OH3vrHLs57I63yVRJ5IGGoiLlo4Dk2/9lo2mMoTynidwa5rSkrgn7v0zdLDeXuj5F
aiJDkD5BgMsinau1RxXi9SK8Ocsi2eyNOT5XP+ukPWDegdwmimVOiwAauYAr3Uj6FXOKJYR3OL2VxYjvHrtHwPPjsmshl
wpEIF9/Hm5vlpFIeUDst/IG8ZhXzkbw5pbrZlPyIYnzLKr/1iF4v6p6WeRmxDElTjicFIsXNs3XFWXlQq0XGJRiZS7FvY
3SqsqFW9gMEOGvZYiO8GUAWoWtix6i/OKCmj/9tqQ4TlCIdKmQgUVQfDEaO9BW3zvoqCTDeWP1zXNEeqeUMZumFTO9nhB
YCOBZdQ/IFYO2Ox4jFEGsYAomflF921Mol4CHXuaYxbK6kowKK7+F86xRwgU8Js75n9d41GG+SxrgVzGrX+VypdQ4Y1XU
JFrxG32R2MrIOIW58pDrYfq//1BtpqRu2RNYyS4ZRIFQMdxuU5Oex765jwb7rmGMJuKKlyJ8FxM/yLz8+whYtrp7wVVmm
IGo43RTCB9ITjq7PCFkZS3a8f4ZVCu/rNJRDRE3Ram9gTXIDZMh0CQbkk8oSi7AdbtTY7eGyUzXBsRZKbxtDgQRJnU5o9
kDggtp8Pf4TnkV67nqeVqu4vUEnS2Adf7dIk8tnVOFSAgLtVPndDo4ripD9zQ+JDa7alW+A2/PGUrRCYp32Rn1r4JIX5t
hB/WgXqz4hCVRwGHmueWkTchmj6hVVlibh0SxBAUQKS0zNFpuf42cuMfn6gkMSlJTVWJtbp2eu8nL2+Dw9y5SWXF3fIOF
lLO+w9HoFiUyNkJETlVok5+mzwAAAAAAAAAAAAAAAAAAAAAAAAQIjA9
-----END CERTIFICATE-----
```

Listing A.7: Certificate with EC:P-256 (issued first).

```
-----BEGIN CERTIFICATE-----
MIIBJzCBzqADAgECAgh//////////jAKBggqhkjOPQQDAjAYMRYwFAYDVQQDDA0gVGVzdCBSb290IENBMCIYDzIxMDAwN
zIyMTIwMDAwWhgPMjEwMDA3MjIxNDAwMDBaMBgxFjAUBgNVBAMMDSBUZXN0IFJvb3QgQ0EwWTATBgcqhkjOPQIBBggqhk
jOPQMBBwNCAARCJUj4j7eC/7Xso3REUscqHlWPvW9zvl5I6TIyzEXFsWxMOQxMuNW4oXE56UiCyJklcpk0JfQUGat+kKQ
```

```
qSUJyMAoGCCqGSM49BAMCA0gAMEUCIDRIH4muIPsHqjvfysliRZJP481pEmnf2K82ZbGZxAQ1AiEA/l4Ojlcl5Ar68u0b
Nxy1f903Vxrmjf2miySmEKQzLns=
-----END CERTIFICATE-----
```

B

CBOR-Encoded Certificate Revocation Lists

The CBOR encoding of CRLs is based on a preliminary specification from IETF[1].

Listing B.1: CBOR-Encoded Certificate Revocation Lists CDDL Definition

```
C509CertificateRevocationList = [
   TBSCertificateRevocationList,
   issuerSignatureValue : any,
]

; The elements of the following group are used in a CBOR Sequence:
TBSCertificateRevocationList = (
   C509CertificateRevocationListType: int,
   issuer: Name,
   thisUpdate: Time,
   nextUpdate: Time,
   revokedCertificates: RevokedCertificates,
   crlExtensions: Extensions,
   issuerSignatureAlgorithm: AlgorithmIdentifier,
)

RevokedCertificates = [
    userCertificate: CertificateSerialNumber,
    revocationDate: Time,
    crlEntryExtensions: Extensions,
]
```

[1] https://github.com/cose-wg/CBOR-certificates/blob/master/draft-cose-cbor-revocation-management.md