Efficient PKI Design for Secure Communication and Collaboration in Space Networks

David Koisser, Albert Schwarzkopf, Ferdinand Brasser
SANCTUARY Systems GmbH
Darmstadt, Germany
david.koisser@sanctuary.dev, albert.schwarzkopf@sanctuary.dev,
ferdinand.brasser@sanctuary.dev

Giacomo Da Broi
Systems Security Engineering Section
European Space Agency
Noordwijk, Netherlands
giacomo.dabroi@esa.int

***Abstract*—The increasing scale and heterogeneity of space systems and future interplanetary assets necessitate secure, interoperable communication under severe operational constraints, including limited processing capacity, intermittent connectivity, and large delays. Existing practices based on pre-shared symmetric keys are inherently unscalable and present operational and strategic challenges in dynamic, multi-operator environments. Public Key Infrastructures (PKIs), which have long addressed these challenges in terrestrial networks, are appealing candidates; however, their reliance on low-latency credential validation (e.g., OCSP, CRLs) renders them unsuitable for networks in which delays need to be tolerated.**

This work proposes a novel PKI architecture designed for space networks. The design leverages a delay- and disruption-tolerant credential validation layer based on peer-to-peer epidemic dissemination of compact, cryptographically verifiable revocation data. The PKI design supports multi-authority environments by enabling inter-party credential issuance with jurisdiction-compliance acknowledgment proofs, allowing validation of cross-domain trust without relying on a connection to ground. A custom simulator evaluating this design at constellation scale demonstrates that, for the targeted scenarios, the proposed mechanism propagates critical revocation updates orders of magnitude faster than CRLs or OCSP Stapling, while incurring significantly lower network overhead.

***Index Terms*—Public key, Public key cryptography, Satellite communications, Network security, Computer security, Security, Satellite networks, Space networks, PKI, Certificate revocation.**

I. Introduction

The rapid expansion of commercial and governmental activity in Earth's orbit, cislunar space, and beyond has transformed the demand for secure, interoperable communications between a variety of space assets. There is an increasingly important need for encryption and authentication of telemetry, telecommands, and payload data; identity proofs to authenticate participating nodes; and frictionless collaboration among heterogeneous operators sharing orbital resources and network infrastructure. These requirements must be satisfied under severe constraints: limited on-board processing power, intermittent connectivity, and propagation delays that can exceed several seconds. The predominant security practice today, i.e., shipping pre-shared symmetric keys with each platform [1], leads to an exponentially growing set of key pairs ($\mathcal{O}(n^2)$ for n nodes), which must be managed individually for every communicating node pair. As the number of spacecraft rises into the tens of thousands, individual key pair management becomes operationally infeasible.

On terrestrial networks, the limitations of static secrets were overcome decades ago through Public Key Infrastructures (PKIs). PKIs enable any two endpoints with no prior negotiation to establish confidential and authenticated connections on demand. A browser verifying an online-banking site illustrates the paradigm: the certificate chain confirms the banking server's identity, while a handshake protocol negotiates encrypted traffic keys on the fly. Further, any other security services relying on digital signatures require a PKI and its key management capability, for instance, to provide data provenance proofs or secure software updates. Translating these well-understood mechanisms into the space domain appears inevitable, yet naive adoption collides with environmental realities.

Foremost, traditional PKIs depend on near-instant credential validation to determine whether a credential is valid at the time of check, e.g., if it has been revoked. For assets milliseconds away in latency, fresh Online Certificate Status Protocol (OCSP) [2] responses remain practical; for satellites with multi-second round-trip times and intermittent connectivity, let alone future visions on lunar or even Mars missions, the paradigm breaks. Prefetched or inherent revocation checks (Certificate Revocation List (CRL) [3], OCSP Stapling [4], or short-lived certificates) come with a potentially long time window in which compromised credentials retain their authority. While these approaches can be configured with short validity periods for reduced vulnerability timeframe, this significantly increases the bandwidth consumption for the entire satellite network. Furthermore, governance complexities emerge when credentials are issued under different national or commercial jurisdictions. Historical incidents in the Internet—such as the 2011 fraudulent issuance of certificates for the "google.com" domain by a French certificate authority [5]—demonstrate that a single jurisdiction can unilaterally undermine global trust. However, terrestrial mitigations to prevent jurisdiction violations rely on central trust authorities [6] or on-demand checks with numerous authorities for every connection establishment [7], and thus are either politically or technically infeasible (further details are outlined in Section V).

To reconcile the security advantages of PKI with orbital

realities, this paper proposes a novel PKI design, enabling credential validation and inter-party trust negotiation for constrained, delayed, and disrupted networks. For revocation checks, our design leverages peer-to-peer epidemic-style dissemination of concise revocation information, instead of central, always-online responders or burdensome prefetched data. With the topology-independent on-contact distribution strategy, each space asset opportunistically transmits and forwards compact security updates, enabling fast propagation without relying on continuous ground contact. Moreover, we incorporate Post-Quantum Cryptography (PQC) to defend against emerging threats while our design remains compliant with established standards. The latter enables interoperability and allows our design to benefit from decades of protocol evolution, formal verification, and widely available hardware acceleration. Our space-ready design enables comprehensive PKI principles: ensuring timely security updates with minimal network demands, providing a sound technical framework for secure collaboration without political friction, and protecting communications against both classical and quantum threats.

Our main contributions include:

- We present the first generic PKI architecture specifically designed for space networks.
- We use a novel combination of primitives to enable efficient and secure revocation and inter-domain policy enforcement.
- Our design fully integrates modern PQC primitives while being conform with established standards.
- We developed an efficient network simulator for large constellations to evaluate our design with thousands of satellites. Our results show that our approach outperforms traditional terrestrial approaches.

II. System Model

For our system model, we aim to accommodate several key trends in the space sector, namely, the shift towards large constellations, the growing number of different actors joining the space sector, and the increasing collaboration between parties to establish complex services, e.g., in the Artemis program [8]. Further, the PKI design must be fit for future development, i.e., operate in settings that will grow in scale and functionality, yet also support novel use cases established by orchestrating space assets and services, e.g., today one can already rent ground station networks [9] and constellations-as-a-service [10].

To formalize this, we assume several co-existing parties in our system called *domains*. Each domain has a typical Certificate Authority (CA) hierarchy in place, with a root CA (the domain's trust anchor), potentially intermediate CAs, and issuing CAs. Any holder of an issued certificate in the system is called a *principal*. While we focus on satellite principles in this paper, other types of entities are not excluded, such as service consumers (e.g. satellite communications user terminals) or other ground entities. Each principal belongs to one of the domains—the satellite's operator—and has a unique identity, proven by its certificate[1]. Each domain operates a large number of principals, i.e., a constellation of satellites, and principals have connections to each other, i.e., via Inter-Satellite Links (ISL). Nevertheless, the CAs' connection to principals is assumed to be imperfect, i.e., experiences delays and disruptions such that a share of principals may miss updates by the CAs, e.g., a satellite being out of sight by the ground station network or simply hibernating).

Domains and their principals aim to establish a secure and scalable collaboration between each other leveraging the PKI, such that confidentiality, integrity, and authentication (regarding their identity) for their communication is ensured[2]. Nevertheless, domains and their CA hierarchy are governed independently from each other and do not fully trust each other[3].

A. Adversary Model

Our adversary model defines three escalating classes of attackers. *All* adversaries aim to impersonate legitimate nodes or forge signed data to undermine authentication guarantees. A *basic adversary* operates under the Dolev-Yao model [11] with full network control and can compromise intermediate/issuing CAs or principals, but cannot break cryptography or compromise root CAs. CAs from one domain may also mistakenly or maliciously issue certificates for another domain. Including the basic capabilities, an *advanced adversary* further has the capability to store encrypted data for future decryption via a quantum computer. An alternative to the advanced adversary is the *quantum-ready adversary* who can break classical asymmetric primitives via a quantum computer. We exclude denial-of-service attacks and assume that the adversary cannot break cryptographic primitives (aside from the addressed quantum threat).

B. Requirements

Designing a robust and efficient PKI for the constrained space environment remains a significant challenge, due to the potentially limited processing capabilities in a heterogeneous space network, large delays, and intermittent connectivity. This is primarily due to two key challenges in the space environment. (1) Revocation information is a critical part of certificate validation, yet in a space environment, it is hard to ensure fresh information without inducing large overheads on the network. (2) Recent incidents in terrestrial PKIs [5] have shown that another crucial aspect is to ensure that domains do not violate other domains' jurisdiction, e.g., issuance of a certificate for another domain's principal. For long-term applicability, it is crucial to develop a generic space PKI that is application- and mission-agnostic, enabling secure and reliable trust management under diverse operational conditions.

[1]We assume there is a Registration Authority validating identities in place, which we consider orthogonal for this paper.

[2]For brevity we focus on the main use case of communication, yet there are numerous additional applications for a PKI, such as data provenance or secure software updates.

[3]In particular, one domain should not be able to issue a certificate for impersonating the identity of another domain's principal.

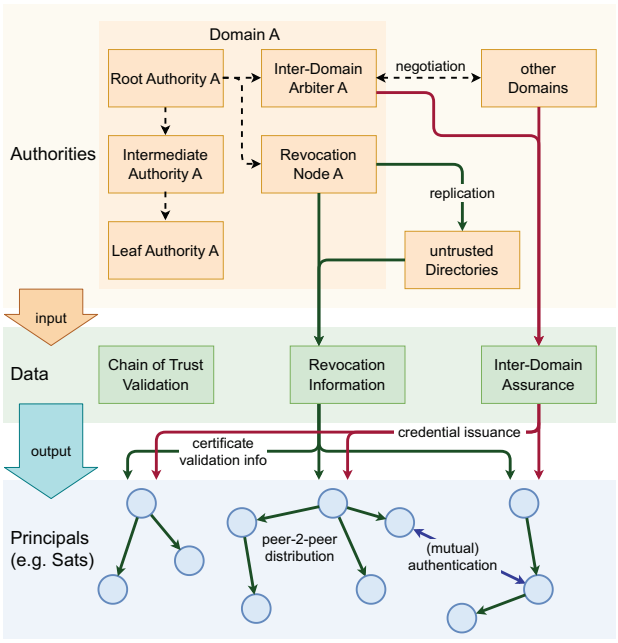

Fig. 1: SHED-SPKI Architecture Overview.

Accordingly, the PKI design should fulfill the following requirements:

R1 - Handle resource constraints: The scheme must impose low computational, storage, and communication overhead on constrained nodes.

R2 - Tolerate delays & disruptions: Communication with a terrestrial central authority (CA) may experience delays or disruptions; thus, the scheme must ensure security under such conditions.

R3 - Minimize update latency: Security demands access to fresh validation information; the scheme must support the timely and efficient dissemination of updates.

R4 - Support independent domains: The scheme must enable collaboration between multiple domains without relying on a central governing authority.

R5 - Flexible inter-domain policies: The scheme must support multiple domains with cross-domain policy enforcement, to prevent overreach between the domains.

R6 - Ensure future-proof interoperability: The scheme shall provide interoperability with existing systems and reduce the friction to integrate with future systems, especially considering the quantum threat.

III. Design

Figure 1 shows an overview of our PKI design, structured around distinct trust domains. Each domain contains a root Certificate Authority (CA) with a self-signed certificate, establishing the foundational trust anchor per domain. A hierarchical CA structure extends from this root, incorporating intermediate CAs and Leaf CAs responsible for certificate issuance for each domain (Figure 1 only shows this hierarchy for Domain A). This structure represents the typical architecture for most PKIs. To support cross-domain coordination and revocation governance, our design introduces two additional domain-specific entities. First, the *Inter-Domain Arbiter* (IDA), which ensures jurisdictional compliance of certificate operations across domains. Second, the *Revocation Node*, which maintains domain-wide revocation information and is responsible for network-wide distribution. These entities collectively construct high-level trust data used for certificate validation and inter-principal trust interactions, while revocation status is explicitly handled to extend traditional PKI validation mechanisms.

The issuance process begins when a Principal requests a certificate for its public key, based on the Certificate Signing Request (CSR) [12], requiring so-called *acknowledgements* from all relevant domains to confirm the absence of jurisdictional conflicts. Briefly, an acknowledgment is a domain's explicit recognition that another domain's certificate does not violate its jurisdiction without implying full trust. The concept of acknowledgments is defined in detail in Section III-A. This inter-domain exchange is mediated by the Inter-Domain Arbiters (IDAs), which each domain operates. Each IDA is performing checks, which complexity depends on the trust model. In environment with high trust between the different domains, basic policy checks may suffice. In contrast, dynamic and heterogeneous domain groups necessitate a more robust mechanism. To address this, we introduce an approach inspired by the current deployment of Certificate Transparency [7], the current terrestrial solution to prevent jurisdiction violations. It enables agreements by each domain on certificate issuance through cross-Arbiter communication, resulting in a cryptographic proof included with the issued certificate. This way, the Principal holding the certificate can simply provide the respective domain's proof to another, which allows the other Principal to validate that no jurisdiction violations occurred, without extending trust outside its domain.

Revocation is initiated by a CA notifying its domain's Revocation Node, which updates the domain's revocation state following a slightly modified version of the V'CER [13] revocation approach. Recognizing the challenges of intermittent connectivity in space environments, the design incorporates an epidemic revocation propagation model by leveraging the concept of a gossip protocol. This epidemic spread of revocation information enables up-to-date Principals to disseminate revocation data to outdated ones. This mitigates Time-of-Check-to-Time-of-Use (TOCTOU) inconsistencies without necessitating constant connectivity to the respective authorities, which is especially challenging in a multi-domain environment. To improve resilience, signed revocation information is redundantly hosted on untrusted third parties. During mutual authentication, Principals exchange proof of non-revocation for the full trust chain, leveraging the epidemic model to ensure freshness and eliminating the need for real-time authority queries or costly prefetched data.

A. Inter-Domain Arbiter

This section will detail our design for the Inter-Domain Arbiter (IDA) approach. First, we will define *acknowledg-*

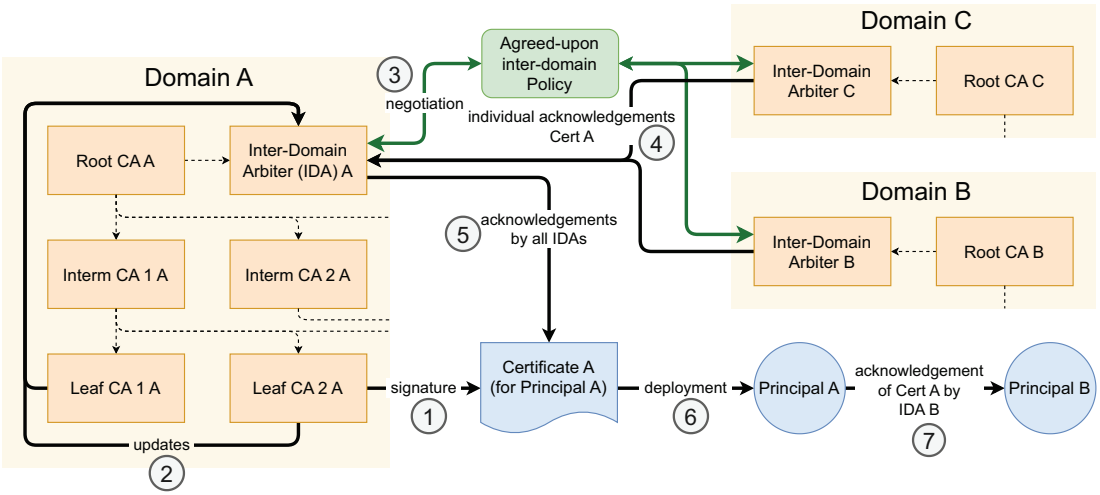

Fig. 2: Exemplary Flow of Certificate Issuance with Inter-Domain Arbiters.

ments, which are crucial for our IDA design. Afterward, we will present the procedure in place for certificate issuance and how Principals leverage the resulting acknowledgments when interacting with each other.

1) Acknowledgments: An important definition for IDA is the concept of acknowledgement. One domain giving an acknowledgement of a certificate issued by another domain is expressing the former domain's approval of the certificate, only at the time of issuance. To properly define this, four aspects are important. First, this acknowledgement is explicitly not expressing full trust in the certificate, as one would do with cross-signing. Second, the underlying policies governing the acknowledgement have to remain valid for the certificate permanently, such as the identity and domain association of the certificate holder (similar to the domain name for certificates in the Internet). Third, the acknowledgement cannot be revoked and is valid for the lifetime of the certificate (or until revocation of the certificate by the issuing authority). Fourth, the policies underlying acknowledgements can be distinct per domain, as acknowledgements will ultimately only be validated inside of their domain (this will be elaborated in Section III-A2). Nevertheless, for practical reasons, the different domain policies should at least roughly align for cohesion in the PKI.

In the context of this paper, the main goal of IDA is to prevent jurisdictional violations (or rather make them apparent). Thus, an acknowledgment is the recognition of the acknowledging domain that it cannot see any jurisdiction violations within an issued certificate. Again, it is important to emphasize that this only expresses the absence of a jurisdiction violation and is in no way a full endorsement of a foreign certificate. As such, this behavior mimics the *practical* implementation of the established Certificate Transparency (CT) [7] approach. While an evidence trail is theoretically available in CT for the browser to validate, in practice, certificates come with a mere promise by several third parties given on issuance to validate the certificate eventually (i.e., the Signed Certificate Timestamp, SCT given by CT logs[4]). Thus, in practice, a modern browser supporting CT (e.g., Chrome, Firefox, Safari) validates something akin to our definition of acknowledgements. Yet, in our model (cf. Section II), we define clear bounds and interactions of domains while the Internet's PKI is more obscure.

2) IDA Definition: For simple PKI systems with only a handful of domains that have strong trust relations between each other, the acknowledgement can be informed by simple agreed-upon policies. For example, a network involving multiple space agencies could agree that no party issues certificates outside their own domain. Thus, each domain could do a simple sanity check on any certificate issuance and check if its own jurisdiction was violated by mistake. If it finds such a violation, it could inform the issuing domain about the accident, trusting it will remedy the mistake, e.g., by revoking the certificate in question. However, the verification mechanism can be independent for each domain. Note that the negotiation between domains to agree on common policies and the potential policies themselves will be highly individual per domain group and use case, and thus these aspects are outside the scope of this paper.

The general idea of IDA is to decouple the validation process from individual principals and delegate the responsibility to the respective domain to vouch for the validity of their own principals. Each domain has a single Inter-Domain Arbiter (IDA) that handles all IDA operations for the entirety of the domain. Figure 2 shows an exemplary flow of IDA when a new certificate is issued, involving the following steps:

1) The CA from domain A will issue (or rather sign, in the context of a CSR) the new certificate.
2) The CA will inform the IDA in its own domain.
3) The IDA will inform other domains' IDAs of the new certificate.

[4]For informal details, see https://certificate.transparency.dev/howctworks/

4) The other IDAs will check the certificate according to the agreed-upon policy and, if successful, will send their individual acknowledgements for the certificate to the original IDA. The acknowledgement is constructed by signing the hash of the certificate.
5) All acknowledgements by all domains will be aggregated.
6) The signed certificate and all acknowledgements (separate from the certificate) are sent to the principal who will own the certificate.
7) A principal from another domain will receive the specific acknowledgement of the IDA of its own domain during the handshake process.

In simple terms, anytime a certificate is issued, all domains will see the certificate and issue signatures to the issuing domain, confirming it does not see any violation concerning its own domain (i.e., acknowledgements). During a handshake, e.g., a principal can then send the acknowledgement of the communication partner's domain along, such that the partner can validate this without direct contact with its own domain. This way, a domain's authority issues a proof that is specifically validated solely by its principals, even though another domain may distribute the proof, thus allowing independent policies per domain. This also means that no inter-party governance is required, yet it could also be used complementary to a strong trust model, such as bridge CAs. Moreover, for an open system supporting a dynamic group of domains with stronger cohesion, we also outline an alternative in Section A-B.

B. Revocation Scheme

Revocation is an important aspect of any PKI, yet the terrestrial approaches typically used to address revocation are impractical for space networks. We will elaborate on these approaches in detail in Section V. As a revocation scheme specifically designed for constrained networks, such as space networks, we chose to use V'CER [13] as the revocation approach for our PKI design. Moreover, we modified some aspects of V'CER to specifically fit our design and increase efficiency. In the following, we will outline how V'CER works and emphasize our modifications.

V'CER employs Sparse Merkle Trees (SMTs) to represent the set of active certificates in a deterministic and cryptographically verifiable structure. Each certificate is mapped to a specific leaf in the SMT using its hash, while the tree root hash—signed by the CA—acts as the root of trust. A cryptographic proof can be constructed by composing hashes along the certificate's leaf hash up to the tree root, which can be provided for this certificate's non-revocation proof. The proof only requires a $\mathcal{O}(\log n)$ number of hashes per proof, and thus per principal, fulfilling requirement **R1**. Thus, a prover can send the verifier its certificate plus its proof, which allows the verifier to reconstruct the root hash and matching it against the signed root disseminated by the CA.

Nevertheless, when the set of active certificates changes (e.g., on a revocation), the SMT will change, and thus all proofs will become invalid. Just like other revocation schemes, the CA can then distribute a delta update with the changes. Yet, unlike other schemes, V'CER covers peers that missed such updates with the help of algorithms allowing updated peers to help outdated peers to become updated as well, addressing requirement **R2**. Note that these algorithms are probabilistic in nature and there is a low chance of failure (increasing with the number of updates missed) that requires principals to request an individual update. The *aggregator* is a minimal data structure that peers exchange on contact and allows them to identify outdated tree roots (i.e., belonging to the different domains' SMTs). We modified this mechanism by removing the epochs used by V'CER and simply using a single SMT per domain, which allows us to simplify the aggregator to a UNIX-timestamp (4 Bytes) and 2 Bytes of parity information per domain. For example, a system with 5 domains requires 14 Bytes of additional data exchanged on contact between peers.

Another modification to simplify V'CER further is the proposed level cache carried only by a *share* of peers. We simply assume all nodes store and maintain their own domain's level cache. Note that these modifications simplify the revocation scheme, yet may sacrifice the efficiency gains from the original proposal. To further improve efficiency for specific use cases, one may particularly consider our simplifications.

C. Extension of Standards & PQC Support

To address requirement **R6**—and specifically PQC—this section will discuss our PKI design's integration into existing standards. Support for PQC can be split into two categories, digital signatures and key encapsulation. The NIST agency has already selected primitives for both categories [14], [15], [16]. Using PQC digital signatures prevents an adversary with a quantum computer from forging signatures, while PQC key encapsulation additionally protects against an adversary that will have a quantum computer in the future to eventually decrypt intercepted messages. The key to support PQC in a PKI thus lies in use of two keys per certificate for the transition phase; both for the traditional cryptography and PQC primitives.

1) X.509 Certificates: To maximize interoperability, our PKI design uses the widespread X.509 standard [17]. While the use of a single key is straightforward, the transition towards a PQC-based deployment requires storing two keys in a certificate. To achieve this, we follow the ITU-T recommendation for multiple cryptographic algorithms in public-key certificates leveraging X.509 extension field; concretely, the `subjectAltPublicKeyInfo`, `altSignatureAlgorithm`, and `altSignatureValue` fields [18].

Using this approach protects against the *quantum-ready adversary* (cf. Section II-A) while remaining compatible with traditional cryptography.

2) Transport Layer Security 1.3: As our PKI design leverage a revocation scheme that requires the prover to send a non-revocation proof similar to OCSP Stapling, we leverage a similar mechanism to achieve this in a standard-conform way.[5]

[5]It is possible to use different transport security protocols with our PKI design, which would require equivalent adaption and extensions.

For this, sending acknowledgments by the IDA and the V'CER non-revocation proofs via the TLS extension fields [19]. Aside from support for the respective primitives in the negotiation, there are no further adjustments needed to support PQC-based key encapsulation, if one generates ephemeral key pairs as required in the standard [19]. However, the use of both traditional and PQC for a hybrid key exchange [20] protects against the *advanced adversary* (cf. Section II-A).

IV. Evaluation

To demonstrate the effectiveness of our PKI design, this section will describe the scenario that we defined to reflect our system model (cf. Section II), then outline the design of our custom satellite network simulator, and finally show the results of our evaluation. Note that for the evaluation, we exclusively focus on the revocation aspect, as this induces the primary overheads, as opposed to the IDA's marginal overhead during handshake.

A. Network Scenario

A high-level motivation for our PKI design is to make it future-proof. Thus, for our evaluation, we constructed a challenging network scenario that reflects our system model in Section II and stress tests our design. We assume five domains in the network (inspired by the main partners in the Artemis program [8]), each with their own constellation in varying sizes reflecting real-world constellations; (1) Amazon's Project Kuiper with ∼3200 satellites [21] (at ∼600 km altitude), (2) SSST's Qianfan with ∼1300 [22] (at ∼1000 km altitude), (3&4) two times Eutelsat's OneWeb with ∼700 [23] (at ∼1200 km altitude), and (5) European Union's IRIS² with ∼300 [24] (at ∼1200 km altitude). Note that such constellations were taken only as representative references for altitude and number of spacecrafts for satellite communications systems. The configuration of the constellations and the connectivity scenario between them do not correspond to any real-world scenario. All constellations are modeled as walker pattern constellations with an inclination of 87°. Satellite can communicate with up to three other satellites (ISL) simultaneously. Furthermore, for a representative ground station network, we use both ESA's Estrack [25] and the AWS Ground Station [9] networks. Although the scenario employs a non-optimised ground network, it offers a unified basis for evaluating various PKI solutions under the intended conditions. To reflect the imperfection for the ground-to-space connectivity mentioned in Section II, 3% of satellites miss the initial CA update. Therefore, our network consists of 33 ground stations as the backbone distribution to deliver CA updates to 6200 satellites in different constellations. We simulate four virtual weeks, over which 28 revocations happen at random times, each affecting 0.5% of the affected domain. For the digital signatures used for the messages, we use the PQC primitive ML-DSA87 to cover the strongest *quantum-ready adversary* (cf. Section II-A) for our evaluation.

B. Simulator

Our target scenario involves a large number of satellites operated by multiple domains. Thus, to evaluate our PKI design, we need a network simulator able to simulate such scenarios. While there are numerous existing and open network simulators, they either do not support space scenarios [26], [27], [28] or they are designed for space, yet cannot scale to thousands of nodes [29], [30], [31]. Therefore, we implemented our own discrete-event network simulator for Earth-centered space networks in Python and C++, consisting of three components. At its core, the simulator implements a highly efficient line-of-sight calculator, which uses an SGP-4 propagator (based on the Skyfield library [32]) and supports both satellite-to-satellite connections and ground-to-satellite (and vice versa) connections, able to calculate them for thousands of nodes in a reasonable timeframe. The second component calculates a representative network delay for communications between nodes, which includes free space path loss, atmospheric attenuation following the ITU-R P.676-10 standard [33], and error-correction codes. The third component is a benchmarking tool, which implements and benchmarks cryptographic primitives (e.g., ML-DSA, SHA256, or Let's Revoke) and was run natively on hardware that represents modern satellite systems, specifically the Zynq UltraScale+ MPSoC ZU3EG (Cortex-A53 1.5 GHz x4). In Appendix B, we will present these measurements. The run times from this script are imported into the simulator to emulate the respective processing of the cryptographic operations.

C. Large-Scale Network Analysis

For the evaluation, we executed the same scenario (including a fixed random seed) with four different revocation schemes. Two represent the traditional revocation schemes with CRLs [3] and OCSP Stapling [4]. CRLs simply distribute verbose lists of all revocations[6], while OCSP Stapling distributes a unique staple for each principal in the network that is valid until expiry (there is no separate revocation mechanism. We also use Let's Revoke [34], a significantly more efficient alternative for CRLs (more details in Section V). Finally, our scheme uses the modified version of V'CER [13] as described in Section III-B. As is typical with revocation information, we also simulate the expiration of all revocation schemes. This is done to eventually let principals assume they are outdated, even in the case of total network occlusion. While we assume this expiration happens after 24 hours for all schemes, one exception is OCSP Stapling with only 6 hours, as OCSP Stapling has no other revocation mechanism[7], and thus to minimize the vulnerability time window, the only way is to reduce the lifetime for each staple.

[6]Note that we explicitly refrain from using delta CRLs, as such a strategy would necessitate either assuming a reliable broadcast, which is impractical, or a mechanism to identify the specific deltas that a node missed and their retransmission, which is absent in the standards and we consider this complex issue as an orthogonal topic.

[7]A stapled OCSP response's signature is always valid until its expiry.

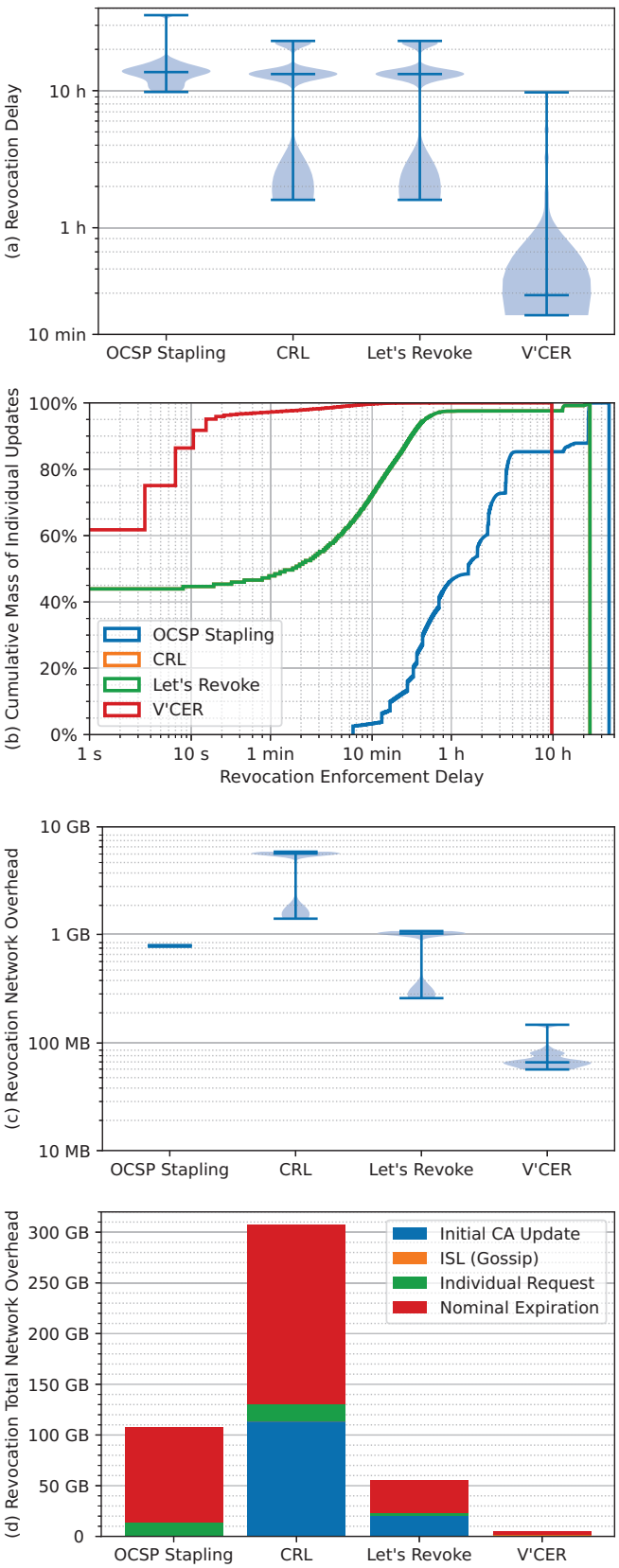

Fig. 3: Results of our large-scale simulations. Note that graphs (a) and (c) show results of individual revocations, and (b) and (d) aggregate all revocations in the simulation.

1) Revocation Delay: Figure 3 (a) shows the results of our simulation in terms of the revocation delay for each revocation event for all revocation approaches. For each approach, we show the minimum (lower line), maximum (upper line), median (middle line), and the distribution of all values (the filled area). Indeed, OCSP Stapling has the longest revocation delay, as principals need to wait for the expiry of their staples and then also need to wait for their distribution. The range emerges as the chronological distance from the time of revocation to the next expiry of revocation information varies. CRL and Let's Revoke look virtually equivalent, as only the transmission delays are different and are insignificant. Both are either very quick (less than an hour) or quite slow (24 hours) in their revocation enforcement. As V'CER leverages gossip-based dissemination, it is quick in its enforcement in the median ($\sim$20 min).

To elaborate on these results, consider Figure 3 (b), which shows the cumulative mass of the revocation delay of all individual principals over all revocations. The horizontal lines indicate the longest delay measured. For CRL and Let's Revoke (both appearing green due to overlap), one can see the holistic delay of the initial CA update of revocation information. Typically, after $\sim$50 minutes, they reach 97% of the network, after which principals only realize they are outdated on expiry. V'CER's gossip approach reaches 96% of the network after $\sim$12 seconds, yet all principals are made aware of the missing information and start to request it. As connectivity to the ground network is difficult in rare individual cases, a handful of principals may take relatively long times to become updated.

2) Revocation Network Overhead: Figure 3 (c) shows the measured network-wide overhead of the different revocation approaches per revocation. The resulting range (or lack thereof) depends on the amount of distributed data, as the longer the revocation process the more data needs to be exchanged while OCSP Stapling solely relies on individual requests. The overall effect of the network overheads in Figure 3 (c) can be seen in (d), which shows the accumulated network overheads over the entire simulated four weeks. The verbose nature of CRL shows, especially compared to Let's Revoke. For OCSP Stapling, keep in mind that this data is more complex to distribute as OCSP staples are unique for each principal, as opposed to the general data used by the other revocation approaches.

V. Related Work

To give an overview of the related work to this paper, we will focus on three large topics; relevant PKI designs for space, state-of-the-art revocation approaches, and schemes to address jurisdiction violations.

A. PKI in Space

There is a plethora of research addressing PKI in space; however, only a few papers address revocation information. Nevertheless, revocation is a crucial and particularly challeng-

ing aspect of PKI in space. Thus, we will solely focus on papers addressing this issue and will further focus on it.

An early paper by Cruickshank [35] proposed standard CRL-based revocation, yet without addressing dissemination challenges specific to space environments. Another popular approach suggested in many works is to simply limit the lifetime of credentials [36], [37], [38], [39]. Yet, such an approach naturally leads to regular re-issuances for the entire network, leading to impractical overheads for large networks.

There are schemes leveraging a form of gossip-based revocation. However, one scheme assumes a reliable adversary detection scheme between nodes [40], which is an impractical assumption (similar to another approach based on a group-established CA [41]). Another scheme assumes an inherently authenticated communication channel between all nodes [42], yet it is not properly defined how this can be achieved in practice. A Web of Trust-like approach is proposed by another scheme [43] in which direct neighbors inherently trust each other. Nevertheless, this only protects against an impersonator and not against an adversary simply issuing a new certificate.

Another approach [44] leverages NOVOMODO [45] to build a space PKI. Briefly, in NOVOMODO, a hash chain is constructed in which each hash represents a different time period, which is then gradually released by the CA. For revocation, CAs publish a hash table with all revoked nodes in the respective time period hash. Another approach shifts the distribution of a CRL to a blockchain [46], yet establishing a reliable access to it is not addressed.

B. Revocation

While Section V-A has a strong focus on revocation in the context of other space PKI proposals, this section focuses on sole revocation schemes. The previous section covers short-lived certificates, while the introduction covers CRLs [3] and OCSP [2]. OCSP Stapling [4] works by principals prefetching OCSP responses for themselves and providing them dynamically to a prover (e.g., during a handshake). However, a valid OCSP staple remains valid during its lifetime, as there is no revocation mechanism for individual staples. Thus, there is a trade-off between the inherent vulnerability time window of staples and the network overheads of re-issuing them. Further, in contrast to other schemes with universal update data, these staples are unique per principal, which, in a practical deployment, significantly complicates their distribution. While CRLs are impractical for space networks due to their large network overhead, CRLite [47] and Let's Revoke [34] leverage efficient data structures to significantly reduce the network overhead. The former, leverages a number of multi-layer bloom filters in which one layer's output is used as the input for the next layer to eliminate the false positives for the revocation check. Let's Revoke adds a unique incremental number n to all certificates and establishes a bitvector that shows each certificate's revocation state on the n-th position. While both schemes are efficient, Let's Revoke is more efficient, especially in context of delta updates, which why we chose it as a main scheme to compare against in our evaluation. We already described V'CER [13] in Section III-B. The main motivation for choosing it for our scheme is its ability to distribute revocation information even if nodes miss updates, which is expected in a space network and is the main concern when considering Let's Revoke, which does not cover such cases.

C. Independent Multi-Domain PKI Support

While we focus on jurisdiction violations between different domains in this paper and have already discussed Certificate Transparency in Section III-A, we additionally want to outline general approaches to handle multiple domains aiming to collaborate in the context of a PKI.

Cross-signing and Bridge CAs are two related strategies for managing trust across multiple domains. Cross-signing enables one CA to sign another's certificate, effectively merging trust hierarchies, while a Bridge CA serves as a centralized intermediary that issues cross-certificates to participating CAs, offering a scalable alternative to pairwise cross-signing. However, both approaches suffer from the need for complex revocation mechanisms, increased trust path complexity, and governance challenges that arise from ambiguous trust scopes and policy misalignment. Moreover, cross-signing implies full transitive trust in external domains, reducing the overall system's security to that of its weakest member, as all certificates are treated equally valid at the cryptographic level. In practice, Bridge CAs are rare and limited to intra-jurisdictional deployments, such as within the US or EU, and no known deployments span multiple, independently governed jurisdictions, as all domains must trust the Bridge CA's governance. The current suggestion (orange book) by the CCSDS for the Intergovernmental Certification Authority [48] (IGCA) states this as well: "The IGCA reduces the years of negotiation between nation states and corporations that currently takes place by establishing a centralized organization [...]" Thus, domains have to agree to the central organization's policies and further accept it as an external single point of failure.

Another proposal is to shift trust among the domains to a Byzantine Fault Tolerance consensus protocol [49], essentially democratizing decision-making and establishing a common root CA and adding failure (and compromise) tolerance. However, this requires agreement among the consensus group about the underlying policies and changes in the consensus group, which may be too complex to govern. Nevertheless, we also suggest a more open strategy allowing for individual policies in Section A-B.

VI. Conclusion

In this paper, we presented a design to enable the efficient use of PKI for collaborative space networks. We identified two core challenges for PKIs in this context; timely revocation enforcement and preventing jurisdiction enforcement in a multi-domain environment. For revocation, we use a slightly modified version of V'CER, which is designed for constrained networks in mind via its topology-independent epidemic distribution. To ensure secure multi-domain collaborations, we designed the Inter-Domain Arbiter scheme, which

exploits both the assumptions in a space network and the way Certificate Transparency works today to ensure no inter-domain policy violations occur in the system.

Acknowledgment

This work was supported by the European Space Agency under the "Secure, Hybrid, Efficient, and Delay-tolerant Space PKI (SHED-SPKI)" (ESA AO/1-11788/23/NL/AF) ARTES 4.0 Space Systems For Safety and Security (4S) activity.

Disclaimer

The views and opinions expressed in this paper are those of the authors and do not necessarily reflect the official policy or position of their respective organizations or affiliations.

References

[1] CCSDS, "Space data link security protocol," https://public.ccsds.org/Pubs/355x0b2.pdf, 2022.

[2] S. Santesson, M. Myers, R. Ankney, A. Malpani, S. Galperin, and D. C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP," RFC 6960, Jun. 2013. [Online]. Available: https://www.rfc-editor.org/info/rfc6960

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

[4] Y. N. Pettersen, "The Transport Layer Security (TLS) Multiple Certificate Status Request Extension," RFC 6961, Jun. 2013. [Online]. Available: https://www.rfc-editor.org/info/rfc6961

[5] The Hacker News, "France government used rogue google SSL digital certificates to spy on users," https://thehackernews.com/2013/12/fake-google-ssl-certificates-made-in.html, 2013.

[6] IdenTrust, "U.s. federal bridge cross-certification," https://www.identrust.com/us-federal-bridge-cross-certification, 2025.

[7] B. Laurie, A. Langley, and E. Kasper, "Certificate Transparency," RFC 6962, Jun. 2013. [Online]. Available: https://www.rfc-editor.org/info/rfc6962

[8] NASA, "Artemis partners," https://www.nasa.gov/artemis-partners/, 2025.

[9] Amazon AWS, "AWS ground station," https://aws.amazon.com/ground-station/, 2025.

[10] Satellogic, "Constellation-as-a-service," https://satellogic.com/products/constellation-as-a-service/, 2025.

[11] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[12] M. Nyström and B. Kaliski, "PKCS #10: Certification Request Syntax Specification Version 1.7," RFC 2986, Nov. 2000. [Online]. Available: https://www.rfc-editor.org/info/rfc2986

[13] D. Koisser, P. Jauernig, G. Tsudik, and A.-R. Sadeghi, "V'CER: Efficient certificate validation in constrained networks," in *31st USENIX Security Symposium (USENIX Security 22)*. Boston, MA: USENIX Association, Aug. 2022, pp. 4491–4508. [Online]. Available: https://www.usenix.org/conference/usenixsecurity22/presentation/koisser

[14] National Institute of Standards and Technology, "Module-lattice-based digital signature standard," U.S. Department of Commerce, Washington, D.C., Tech. Rep. Federal Information Processing Standards Publications (FIPS) 204, 2024.

[15] ——, "Stateless hash-based digital signature standard," U.S. Department of Commerce, Washington, D.C., Tech. Rep. Federal Information Processing Standards Publications (FIPS) 205, 2024.

[16] ——, "Module-lattice-based key-encapsulation mechanism standard," U.S. Department of Commerce, Washington, D.C., Tech. Rep. Federal Information Processing Standards Publications (FIPS) 203, 2024.

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

[18] International Telecommunication Union, "Recommendation X.509 (10/19)," International Telecommunication Union, Tech. Rep., 2019. [Online]. Available: https://www.itu.int/rec/T-REC-X.509-201910-I/en

[19] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3," RFC 8446, Aug. 2018. [Online]. Available: https://www.rfc-editor.org/info/rfc8446

[20] D. Stebila, S. Fluhrer, and S. Gueron, "Hybrid key exchange in TLS 1.3," Internet Engineering Task Force, Internet-Draft draft-ietf-tls-hybrid-design-13, Jun. 2025, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-tls-hybrid-design/13/

[21] Amazon, "Everything you need to know about project kuiper, amazon's satellite broadband network," https://www.aboutamazon.com/news/innovation-at-amazon/what-is-amazon-project-kuiper, 2025.

[22] SatNews, "China's thousand sails satellite plan at phase 2," https://news.satnews.com/2025/04/01/chinas-thousand-sails-satellite-plan-at-phase-2/, 2025.

[23] Eutelsat, "Our Network OneWeb," https://oneweb.net/our-network, 2025.

[24] European Commission, "Commission takes next step to deploy the IRIS² secure satellite system," https://defence-industry-space.ec.europa.eu/commission-takes-next-step-deploy-iris2-secure-satellite-system-2024-12-16_en, 2024.

[25] ESA, "Estrack: ESA's global ground station network," https://www.esa.int/Enabling_Support/Operations/ESA_Ground_Stations/Estrack_ESA_s_global_ground_station_network, 2025.

[26] nsnam, "ns-3," https://www.nsnam.org/, 2025.

[27] OpenSim Ltd., "OMNeT++," https://omnetpp.org/, 2025.

[28] Boeing Company, "Common Open Research Emulator (CORE)," https://www.nrl.navy.mil/Our-Work/Areas-of-Research/Information-Technology/NCS/CORE/, 2025.

[29] CNES, "SNS3," https://www.sns3.org/, 2025.

[30] ——, "OpenSAND," https://www.opensand.org/, 2025.

[31] OpenSim Ltd., "OS³, the Open Source Satellite Simulator," https://omnetpp.org/software/2013/08/14/os3-released.html, 2025.

[32] GitHub, "Skyfield," https://github.com/skyfielders/python-skyfield/, 2025.

[33] International Telecommunication Union, "P.676: Attenuation by atmospheric gases, recommendation p.676-10," International Telecommunication Union, Tech. Rep., 2013. [Online]. Available: https://www.itu.int/rec/R-REC-P.676-10-201309-S/en

[34] T. Smith, L. Dickinson, and K. Seamons, "Let's revoke: Scalable global certificate revocation," in *Network and Distributed Systems Security (NDSS) Symposium 2020*, 2020.

[35] H. Cruickshank, "A security system for satellite networks," in *Fifth International Conference on Satellite Systems for Mobile Communications and Navigation, 1996.* IET, 1996, pp. 187–190.

[36] A. Seth and S. Keshav, "Practical security for disconnected nodes," in *1st IEEE ICNP Workshop on Secure Network Protocols, 2005.(NPSec).* IEEE, 2005, pp. 31–36.

[37] A. Roy-Chowdhury, J. S. Baras, and M. Hadjitheodosiou, "An authentication framework for a hybrid satellite network with resource-constrained nodes," in *International Conference on Space Information Technology*, vol. 5985. SPIE, 2006, pp. 1094–1105.

[38] T.-H. Chen, W.-B. Lee, and H.-B. Chen, "A self-verification authentication mechanism for mobile satellite communication systems," *Computers & Electrical Engineering*, vol. 35, no. 1, pp. 41–48, 2009.

[39] W. Meng, K. Xue, J. Xu, J. Hong, and N. Yu, "Low-latency authentication against satellite compromising for space information network," in *2018 IEEE 15th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. IEEE, 2018, pp. 237–244.

[40] Y. Qian, B. Cao, X. Chen, and X. Du, "A certificate revocation scheme for space network," in *2009 5th International Conference on Wireless Communications, Networking and Mobile Computing*. IEEE, 2009, pp. 1–5.

[41] R. Fang and F. Jiulun, "An adaptive distributed certificate management scheme for space information network," *IET information security*, vol. 7, no. 4, pp. 318–326, 2013.

[42] Z. Jia, X. Lin, S.-H. Tan, L. Li, and Y. Yang, "Public key distribution scheme for delay tolerant networks based on two-channel cryptography," *Journal of Network and Computer Applications*, vol. 35, no. 3, pp. 905–913, 2012.

[43] C. I. Djamaludin, E. Foo, S. Camtepe, and P. Corke, "Revocation and update of trust in autonomous delay tolerant networks," *Computers & Security*, vol. 60, pp. 15–36, 2016.

[44] M. N. M. Bhutta, H. Cruickshank, and Z. Sun, "Public-key infrastructure validation and revocation mechanism suitable for delay/disruption

tolerant networks," *IET Information Security*, vol. 11, no. 1, pp. 16–22, 2017.
[45] S. Micali, "Scalable certificate validation and simplified pki management," in *1st Annual PKI research workshop*, vol. 15, 2002.
[46] J. Guan, Y. Wu, S. Yao, T. Zhang, X. Su, and C. Li, "Bsla: blockchain-assisted secure and lightweight authentication for sgin," *Computer Communications*, vol. 176, pp. 46–55, 2021.
[47] J. Larisch, D. Choffnes, D. Levin, B. M. Maggs, A. Mislove, and C. Wilson, "CRLite: A scalable system for pushing all tls revocations to all browsers," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 539–556.
[48] CCSDS, "Intergovernmental certification authority," https://ccsds.org/Pubs/357x1o1.pdf, 2024.
[49] D. Koisser, D. Fischer, M. Wallum, and A.-R. Sadeghi, "Trusat: Building cyber trust in collaborative spacecraft networks," in *2022 IEEE Aerospace Conference (AERO)*. IEEE, 2022, pp. 1–12.
[50] D. Mazieres, "The stellar consensus protocol: A federated model for internet-level consensus," *Stellar Development Foundation*, vol. 32, no. 4, pp. 1–45, 2015.

APPENDIX A
FUTURE WORK

A. Gossip-based Let's Revoke

Another option worth mentioning is a theoretical extension to Let's Revoke. The general idea is to extend Let's Revoke with a form of gossip, akin to the aggregator exchange in V'CER, and thus enabling the epidemic spread of the Let's Revoke data structure across the network. However, such a design is not straightforward, as V'CER's aggregator is tailored to its data structure and not directly applicable to Let's Revoke's data structure without security or efficiency issues. Nevertheless, we implemented a naive version of this idea into our simulator, and the preliminary results were promising. Thus, properly defining such a scheme seems appealing.

B. Stellar Consensus Protocol for IDA

Instead of the strict one-to-all interactions for certificate issuance in our definition of IDA in Section III-A2, another method allows for an open and dynamic domain space, akin to the Internet. For this, one could leverage the Stellar Consensus Protocol (SCP) [50], a Byzantine Fault Tolerant protocol that allows each node to pick its own set of trusted nodes, called its consensus slice. As long as there is an overlap of these individual consensus slices (unique for each node), SCP ensures a secure network-wide consensus. This is a very useful property in settings where it is hard to agree on common trust among the nodes in a network, such as a real-world deployment including numerous national and international space agencies as well as private companies. Thus, all IDAs could shift trust into a consensus-based model with open membership and remove single points of failure for inter-domain certificate issuance.

APPENDIX B
EVALUATION OF PQC PRIMITIVES

To generate representative run time numbers for use in our network simulator (cf. Section IV), we implemented a benchmark to measure the run times of different cryptographic primitives natively on different platforms. To achieve this, we leveraged the popular WolfSSL[8] and OpenSSL[9] libraries, which have PQC support for different algorithms. We repeated each experiment 100 times and averaged the results. We further assume 620 Bytes for a minimal X.509 certificate, plus the respective sizes for the public key and a signature. In the following, we separate the measurements between *Key Encapsulation* primitives and *Digital Signature* primitives.

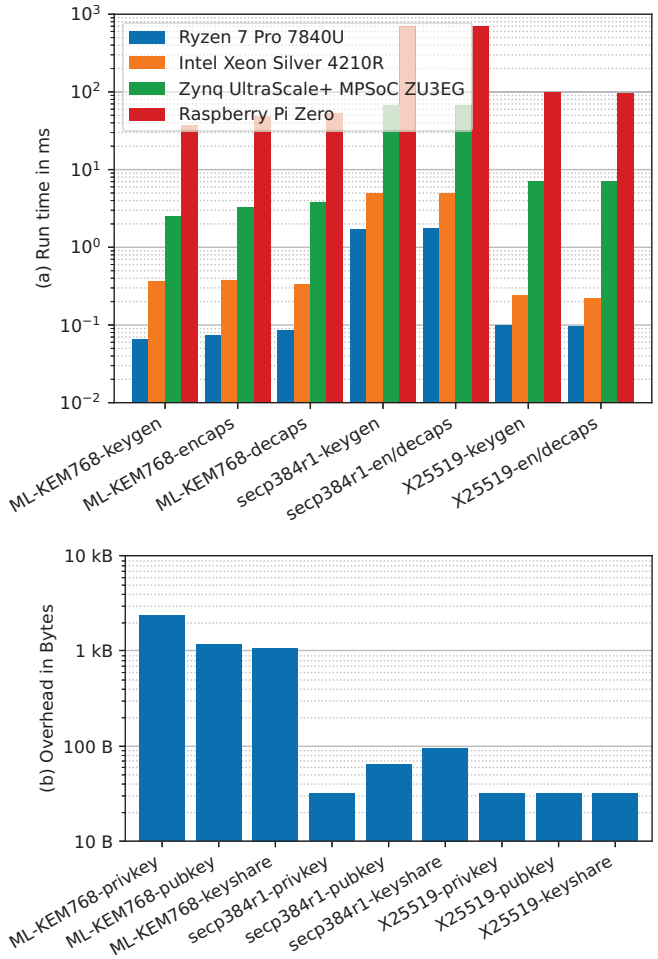

Fig. 4: Run time (a) and bytes overhead (b) measurements for different key encapsulation primitives.

We measured four different platforms that we attribute to different use cases:

- *Commercial:* Ryzen 7 Pro 7840U (2.4 GHz ×8)
- *Server:* 2× Intel Xeon Silver 4210R (2.4 GHz ×10)
- *Satellite:* Zynq UltraScale+ MPSoC ZU3EG (Cortex-A53 1.5 GHz ×4)
- *Limited Embedded System:* Raspberry Pi Zero (ARM1176JZF-S 1 GHz ×1)

Figure 4 shows our results for the different key encapsulation primitives, including `ML-KEM768`, `secp384r1`, and

[8]WolfSSL with native PQC support: https://www.wolfssl.com/products/wolfcrypt-post-quantum/
[9]OpenSSL with SPHINCS and Falcon support via the `oqs-provider` extension: https://github.com/open-quantum-safe/oqs-provider

X25519. Note that at the time of evaluation, we did not find a suitable implementation for HQC, which was only recently selected by NIST at the time of writing. For the run times in Figure 4 (a), it is noteworthy that the PQC primitives take significantly less time than the traditional counterparts. While such result is expected and in line with previously reported measurements, the run times for secp384r1 cannot be compared against X25519 due to their different key sizes. Our selection of secp384r1 aligns with current recommendations for minimum cryptographic key sizes for secp. Further, we use a prototypical reference implementation for ML-KEM, while secp is based on production-ready code. For the bytes overhead in Figure 4, we can see the expected, significant increase in bytes for ML-KEM.

Figure 5 shows our results for the different digital signature primitives, including ML-DSA87, SPHINCS-SHAKE256, Falcon1024, RSA4096, and secp384r1. On the one hand, Figure 5 (a) shows the run time results, and we see the previously mentioned anomaly again. Falcon needs a particularly large amount of entropy to generate. On the other hand, Figure 5 (b) shows the bytes overheads and no noteworthy measurements.

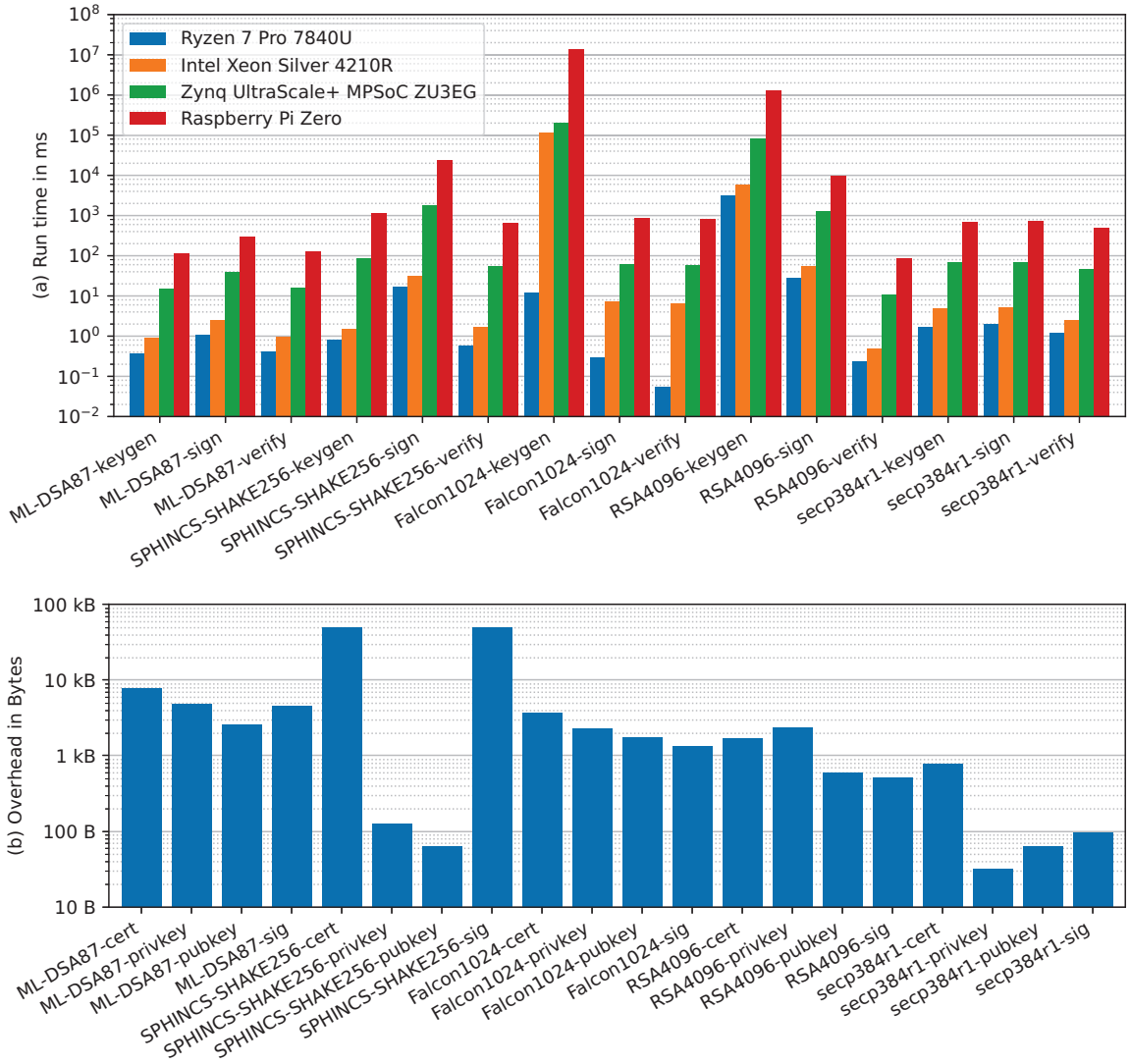

Fig. 5: Run time (a) and bytes overhead (b) measurements for different digital signature primitives.