PNNL-35760

Exploring the Adoption Challenges of Post-Quantum Cryptography in EV Charging Infrastructure

April 12, 2024

Thomas E. Carroll
Moran, Addy
Redington, Lindsey

U.S. DEPARTMENT OF ENERGY

Prepared for the U.S. Department of Energy
Under contract DE-AC05-76RL01830

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor Battelle Memorial Institute, nor any of their employees, **makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights**. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or Battelle Memorial Institute. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

**Available to DOE and DOE contractors from
the Office of Scientific and Technical
Information,
P.O. Box 62, Oak Ridge, TN 37831-0062**
www.osti.gov
**ph: (865) 576-8401
fox: (865) 576-5728
email: reports@osti.gov**

**Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
or (703) 605-6000
email: info@ntis.gov
Online ordering: http://www.ntis.gov**

PNNL-35760

Exploring the Adoption Challenges of Post-Quantum Cryptography in EV Charging Infrastructure

April 12, 2024

Thomas E. Carroll
Moran, Addy
Redington, Lindsey

Prepared for
the U.S. Department of Energy
Under Contract DE-AC05-76RL01830
Pacific Northwest National Laboratory
Richland, Washington 99354

Abstract

The rapid evolution of electric vehicle (EV) technology and the corresponding growth of the Electric Vehicle Charging Infrastructure (EVCI) brings to light significant cybersecurity concerns, notably in the context of emerging post-quantum computing capabilities. This report, prepared by Pacific Northwest National Laboratory (PNNL) under the U.S. Department of Energy contract, delves into the challenges associated with integrating Post-Quantum Cryptography (PQC) into EVCI to safeguard against potential quantum computing threats. Post-quantum computers will eventually be able to invalidate technologies secured through public key cryptography. As part of this effort, the primary gaps and challenges in the EVCI were investigated with a focus on comparing traditional algorithms against PQC algorithms. One of the notable findings was that the P-521 algorithm was frequently surpassed in performance by PQC algorithms.

This document provides a thorough examination of the hurdles the industry can expect when transitioning to PQC within the EVCI, such as interoperability concerns, the computational and memory demands of PQC algorithms, and the organizational readiness for such a transition. It emphasizes the necessity of a forward-thinking approach to cybersecurity, advocating for early and strategic engagement among EVCI stakeholders to ensure a seamless and cost-effective migration to quantum-resistant cryptographic standards. Through this report, the authors aim to catalyze awareness and action among policymakers, industry leaders, and cybersecurity professionals towards fortifying the EVCI against emerging quantum threats, thereby securing the infrastructure essential for the future of electric mobility.

Acronyms and Abbreviations

AES	Advanced Encryption Standard
CA	Certificate Authority
CNP	Charging Network Provider
CPU	Central Processing Unit
CRQC	Cryptograpically-Relevant Quantum Computer
DER	Abstract Syntax Notation One Distinguished Encoding Rules
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie-Hellman
ECDHE	Elliptic Curve Diffie-Hellman Ephemeral
ECDSA	Elliptic Curve Digital Signature Algorithm
ECU	Electronic Control Unit
EV	Electric Vehicle
EVCI	EV Charging Infrastructure
FALCON	Fast-Fourier Lattice-based Compact Signatures over NTRU
GDPR	General Data Protection Regulation
HKDF	Hash-Based Key Derivation Function
HSM	Hardware Security Module
mTLS	Mutual TLS
ISO	International Organization for Standardization
KEM	Key Encapsulation Mechanism
NIST	National Institute of Standards and Technology
NTRU	Nth Degree Truncated Polynomial Ring Units
OCPP	Open Charge Point Protocol
OCPI	Open Charge Point Interface
OCSP	Online Certificate Status Protocol
PCI DSS	Payment Card Industry Data Security Standard
PII	Personally Identifiable Information
PNNL	Pacific Northwest National Laboratory
PQ	Post Quantum
PQC	Post Quantum Cryptography
PQ/T	Post Quantum/Traditional
RSA	Rivest-Shamir-Adleman
SHA	Secure Hash Algorithm
SIS	Shortest Integer Solution
SVP	Shortest Vector Problem

TEE	Trusted Execution Environment
TLS	Transport Layer Security
V2G	Vehicle-to-Grid

Acknowledgments

We thank Lee Slezak from the Department of Energy's Vehicle Technologies Office (VTO) office for his assistance. This study was supported by the U.S. Department of Energy under Contract DE-AC05-76RLO1830.

Contents

Figures

Tables

Listings

1.0 Introduction

There is a drive to accelerate the adoption of electric vehicles (EVs) as a strategic measure to mitigate the emissions of greenhouse gases and other pollutants. Notably, the Biden Administration's "Investing in America Agenda" aims for EVs to constitute 50% of all new vehicle sales by 2030, necessitating concerted efforts from both the private and public sectors [1]. In 2017, global EV sales reached $1 million, surging to over $10 million in 2022 [2]. Specifically, the National Electric Vehicle Infrastructure (NEVI) formula program has earmarked $5 billion over five years to assist states in developing networks of fast charging stations along alternative fuel corridors, particularly targeting the Interstate Highway System [3]. This investment underscores the commitment to transforming the nation's transportation infrastructure. However, amidst this rapid deployment, there has been insufficient emphasis on cybersecurity measures.

The EV charging infrastructure (EVCI) is evolving to become more digitalized and intelligent. Advanced communications underlie the charging process, charging station management, smart charging, and grid ancillary services. As the EVCI relies on public key cryptography (PKC) for securing communications and transactions, it faces a significant future challenge: the advent of quantum computing. Post-quantum computing's potential to break current PKC systems necessitates a proactive shift toward post-quantum cryptography (PQC) to safeguard the authenticity and confidentiality of data and control within the EV charging ecosystem. Adopting PQC will be pivotal in maintaining the security of EV charging, protecting user data, and ensuring the reliability of the energy grid that supports this critical infrastructure. The journey toward quantum-resistant cryptography represents a crucial step in future-proofing EV charging and its related infrastructure against the next generation of cyber threats.

Traditional PKC is a method of encrypting or signing data with a key pair. Each key pair is comprised of a public key and a numerically-related private key. The public key is made available for anyone to use. A public key certificate is a digital document that cryptographically links the public key to the owner. Public key infrastructure (PKI) is then used to manage and distribute the certificates.

Although it's theoretically feasible to deduce the private key from its public counterpart, doing so is considered computationally impractical with today's computing technology. However, the emergence of quantum computing poses a significant challenge to this assumption. Quantum computing is a rapidly emerging technology that uses properties of quantum physics to compute and store data, providing the capacity to solve some complex problems more efficiently than traditional computers. A sufficiently large general quantum computer, known as a Cryptographically Relevant Quantum Computer (CRQC), will have the potential to quickly break existing traditional PKC. For public key cryptosystems that are widely utilized, CRQC will derive the private key in a matter of tens to hundreds of hours [4, 5, 6]. Once that epoch has been reached, encrypted data that is considered safe today may be rapidly decrypted. Digital signatures will be readily forged, degrading trust, authenticity, and source origination. In the case of PKI, the efforts to re-key a certificate authority (CA) and issue new certificates would exceed the time needed to attack the new signatures.

National Institute of Standards and Technology (NIST), an agency of the U.S. Department of Commerce is standardizing PQC systems to defend against traditional and quantum cryptanalytical advancements. PQC systems are substitutes for traditional asymmetric cryptosystems, serving the same purposes and goals, but they are resistant to CRQC. Efforts to accelerate PQC adoption are underway [7, 8, 9] even though the formal PQC standardization process has yet to be completed. Research continues to harden the PQC primitives against side-channel attacks, where the hardware that the algorithm runs is exploited to gather information that violates the security objectives [10, 11, 12]. While work is underway to guide the PQC transition [13], the push is driven by lengthy time requirements expected for the transition. Interoperability is paramount during the transition, with upgraded vehicles needing to maintain backward compatibility with existing chargers, and conversely, upgraded chargers accommodating older vehicle models. With respect to interoperability, NIST states that "[as] a general rule, cryptographic algorithms cannot be replaced until all components of a system are prepared to process the replacement" [14]. The National Cybersecurity Center of Excellence, a division of NIST, also states that "[a] truly significant challenge will be to maintain connectivity and interoperability among organizations and organizational elements during the transition from quantum-vulnerable algorithms to quantum-resistant algorithms" [15].

When contemplating the security implications of quantum computing, two primary threats emerge: the strategy of "harvest now, decrypt later" [16] and the potential for forging digital signatures.

- "Harvest now, decrypt later": This scenario entails adversaries collecting encrypted data transmitted over networks today with the intention of decrypting it in the future when quantum computers become available. The threat arises from the fact that quantum computers could break key exchange algorithms, making it possible to access the encrypted information. This scenario is concerning for data that needs to remain confidential over long periods, such as state secrets or personal data subject to privacy laws. The anticipation of future decryption capabilities necessitates the early adoption of quantum-resistant encryption methods to protect sensitive data from future threats.

- Forging digital signatures: Digital signatures are a crucial element of cybersecurity, providing authentication, integrity, and non-repudiation to digital communications and transactions. The challenge in the quantum era is to develop digital signature schemes that remain secure against quantum attacks, ensuring that digital signatures cannot be forged and that the integrity of signed data, firmware, and transactions remains intact.

Considering the nature of EVCI data and control, the threat posed by the "harvest now, decrypt later" scenario is less concerning compared to the risk of digital signature forgery. This is mainly due to the fact that sensitive data, like cardholder data, generally remains pertinent for a finite time span, such as seven years. Addressing the "harvest now, decrypt later" risk can be managed transparently, for example, by upgrading transport layer security (TLS) libraries on devices equipped with adequate computing and memory capacities. On the other hand, tackling the challenge of digital signature forgery, particularly in the form of digital certificates, involves complexities and logistics that require global coordination that unfolds over many years,[1] making it a concern of increasing urgency each passing year.

The PQC transition presents a unique challenge in the EV automotive and infrastructure sectors due to the extended life span of the systems, where product longevity is a significant consideration. Cars and infrastructure are designed to last for many years, often outliving the rapid technological advancements in cybersecurity. This durability means that vehicles on the road today might still be in use when quantum computing becomes a reality, potentially rendering their existing cryptographic protections obsolete. Long-lived assets require forward-thinking strategies to ensure that these vehicles and infrastructure can be updated to quantum-resistant standards.

To assist in the transition to PQC for the EVCI, Pacific Northwest National Laboratory (PNNL) has published a report that inventories traditional cryptography algorithms in the infrastructure, focusing on the most prominent protocols used in the EVCI and the risks and the consequences of traditional algorithms being exploited [18].

This paper explores factors and challenges associated with transitioning the EV charging industry to PQC. The research specifically focused on the technology and communication between the EV and the charger and the charger and the charging station management system (CSMS). The National Electric Vehicle Infrastructure (NEVI) program identifies three protocols prominent in those communications and were explored as part of this effort [19].

Safeguarding over-the-air software updates, firmware updates, and software validation from quantum computing is a critical need. In addition to NIST's PQC initiatives, NIST SP 800-208 [20] offers methodologies for firmware and software signing considered to be resilient against quantum attacks. Notably, the National Security Agency (NSA), through its Commercial National Security Algorithm Suite 2.0, has recommended the immediate adoption of these signature schemes [21]. However, a detailed examination of these protective measures exceeds the scope of this document. This paper examines the nuances of PQC adoption across the components and communications discussed above and are organized into five key areas: key, ciphertext, and signature size; computational resources; interoperability; upgradability; and organization. Although this paper identifies challenges in each of the aforementioned areas, the authors emphasize that with proactive and strategic planning, it's possible to mitigate these challenges effectively. By anticipating potential hurdles in computational and memory demands, key and data sizes, system interoperability, upgrade pathways, and organizational readiness, stakeholders can devise comprehensive strategies to minimize disruptions. Such forward-thinking approaches will not only streamline the transition process but also significantly reduce the associated costs and complexities. This underscores the importance of early engagement, thorough assessment, and adaptive planning in ensuring a smooth and cost-efficient shift to PQC for the EV charging sectors.

[1] As a point of comparison, the deployment of the existing PKI spanned nearly two decades [17].

This report is organized as follows: Section 2 provides essential background information on PQC and EVCI. Section 3 delves into the various factors and challenges associated with the integration of PQC into EVCI. The report culminates in Section 4, where the findings are summarized.

Note 1: This paper utilizes terminology introduced in “Terminology for Post-Quantum Traditional Hybrid Schemes” [22], an Internet Engineering Task Force (IETF) resource. The term `traditional` is utilized to denote systems and algorithms that operate in accordance to traditional physics, with `classical` or `conventional` serving as interchangeable synonyms.

Note 2: This paper seeks to explore the challenges associated with transitioning EVCI to PQC. It concentrates on identifying and understanding the transition hurdles, while not addressing specific algorithm implementation issues such as timing irregularities that could potentially diminish the effectiveness of the cryptographic measures.

2.0 Background

In this section, an introductory overview of post-quantum cryptography and the EV charging infrastructure is provided. While the authors aimed to provide sufficient information for readers to comprehend the core concepts, readers are encouraged to explore additional resources for a more thorough and comprehensive understanding of these topics.

2.1 Introduction to Post-Quantum Cryptography

Post-quantum cryptography (PQC) refers to cryptographic algorithms designed to secure communications, data, and digital identities against the potential capabilities of future large-scale quantum computers. Quantum computers represent a fundamentally different approach to computing compared to traditional computers. At the heart of traditional computing are bits, the smallest unit of data, which exist in one of two states: 0 or 1. These bits form the basis of all operations, with complex calculations broken down into binary operations that traditional computers can process. Quantum computers use quantum bits or `qubits`, where, unlike traditional bits, qubits can exist in a state of 0, 1, or both 0 and 1 simultaneously, thanks to a principle known as `superposition`. Superposition allows quantum computers to perform many calculations simultaneously, potentially solving complex problems much faster than traditional computers can.

The PQC effort is a result of the risk of quantum computing becoming readily available, a resource that has been proven to exploit the assumptions that keep PKC secure. Cryptographically-relevant quantum computers (CRQC) will be able to solve mathematical problems exponentially faster than the best-known algorithms running on traditional computers. This includes problems upon which much of today's PKC relies, such as factoring large numbers and solving discrete logarithms. For instance, it has been estimated that it would take thirty trillion years to brute force RSA-2048 using today's classical computers. The designers of the algorithm understand that RSA is in fact "breakable," but it would take an unrealistic amount of time and is therefore considered secure. It has been estimated that it would take a quantum computer a matter of hours to break a public key algorithm, now making those algorithms insecure [23].

As quantum computing technology advances, it poses a significant threat to the foundation of modern cryptography, particularly PKC algorithms like RSA and Elliptic Curve Cryptography (ECC), which protect sensitive data, provide unique digital identities for users, devices, and applications, as well as, secure end-to-end communications. The goal of post-quantum cryptography is to develop new cryptographic systems that are secure against both traditional and quantum computing threats. These systems aim to ensure the confidentiality, integrity, and authenticity of communications and data in a post-quantum world. Researchers in this field are exploring various mathematical approaches that are believed to be resistant to the computational power of quantum computers, including lattice-based cryptography, hash-based cryptography, code-based cryptography, and multivariate polynomial cryptography.

NIST initiated a process to select and standardize post-quantum cryptographic algorithms. The NIST Post-Quantum Cryptography algorithm selection process is designed to be transparent, open, and inclusive, involving multiple rounds of evaluation and public feedback. At the time of writing, the NIST PQC selection process completed the third round, announcing four cryptographic algorithms for standardization. These algorithms include one for key establishment (CRYSTALS-KYBER) and three digital signature schemes (CRYSTALS-Dilithium, FALCON, and SPHINCS+). Moreover, the candidate algorithm announcement [24] recommended CRYSTALS-KYBER and CRYSTALS-Dilithium for most use cases due to their strong security and performance. NIST has initiated the process of standardizing these algorithms, releasing draft standards for three [25, 26, 27] of the four selected algorithms in 2023, with the goal of completing the process in 2024. The draft standard for the fourth algorithm, FALCON, is expected to also be released in 2024 [28]. Furthermore, NIST has selected four additional algorithms to advance to the fourth round for further analysis, BIKE, Classic McEliece, HQC, and SIKE, with the intention of possibly standardizing one or more of these in the future [24].

Table 1. Definition of NIST Post-Quantum Security Levels

PQC Security Level	Hardness
1	At least as hard as AES-128 key search
2	At least as hard as SHA-256/SHA3-256 collision search
3	At least as hard as AES-192 key search
4	At least as hard as SHA-384/SHA3-384 collision search
5	At least as hard as AES-256 key search

A key component of the NIST PQC selection process was establishing criteria to evaluate security strength. These criteria drew on principles from existing symmetric cryptography, which is anticipated to offer significant resistance to quantum cryptanalysis [29]. The security level definitions are reproduced in Table 1.

Returning to the topic of algorithms, a summary overview of each is provided. For a more comprehensive treatment, readers are encouraged to consult the cited references.

Table 2. PQC key exchange mechanism algorithm(s) and their properties

PQC Security Level	Algorithm	Public Key Size (in bytes)	Private Key Size (in bytes)	Ciphertext Size (in bytes)
1	Kyber512	800	1632	768
3	Kyber768	1184	2400	1088
5	Kyber1024	1568	3168	1588

CRYSTALS-Kyber [30] (which is referred to as Kyber for conciseness in the rest of the paper) is one of the algorithms, along with CRYSTALS-Dilithium (simply referred to as Dilithium), available in CRYSTALS (Cryptographic Suite for Algebraic Lattices) [31]. The algorithm's security foundation is based on the hardness of lattice-based problems, specifically Module Learning With Errors (MLWE) and Module Short Integer Solution (MSIS) problems. Kyber is a key encapsulation mechanism (KEM), a cryptographic method used to securely exchange encryption keys between two parties. It involves one party generating a pair of keys: a public key, which can be shared openly, and a private key, which is kept secret. To securely send a message, the sender uses the recipient's public key to encapsulate, or encrypt, a session key. This encapsulated key is then sent to the recipient, who uses their private key to decapsulate, or decrypt, the session key. There are three parameter sets, named Kyber512, Kyber768, and Kyber1024. As shown in Table 2, they differ in security level, as well as their length of public key, private key, and ciphertext. Kyber is undergoing standardization as a Module-Lattice Key Encapsulation Mechanism (ML-KEM) within FIPS 203 [25].

Table 3. PQC digital signature algorithms and their properties

PQ Security Level	Algorithm	Public Key Size (in bytes)	Private Key Size (in bytes)	Signature Size (in bytes)
1	FALCON512	897	1281	666
2	Dilithium2	1312	2528	2420
3	Dilithium3	1952	4000	3293
5	FALCON1024	1793	2305	1280
5	Dilithium5	2592	4864	4595

The digital signature schemes' properties are tabulated in Table 3. Dilithium [32] shares the mathematical underpinning of Kyber. There are three parameter sets, Dilithium2, Dilithium3, and Dilithium5, differing in security strength, key size, and signature size. Dilithium is in the process of standardization as Module-Lattice-Based Digital Signature Algorithm (ML-DSA) under FIPS 204 [26].

At its core, FALCON (Fast-Fourier Lattice-based Compact Signatures over NTRU) is based on two main mathematical concepts [33]: NTRU and the Shortest Vector Problem (SVP) in lattice cryptography. FALCON has two variations, FALCON512 and FALCON1024. FALCON is unique in that it involves the use of complex numbers, which are approximated with IEEE-754 double-precision floating point numbers. FALCON sets itself apart from Dilithium through its ability to generate digital signatures that are notably more compact in size. This characteristic is particularly significant in the realm of cryptographic applications where bandwidth efficiency and storage constraints are critical considerations.

Lastly, SPHINCS+ is a hash-based cryptographic system, which means it relies on the security properties of hash functions, mathematical operations that convert input data into a fixed-size string of characters, regardless of the input's length. Finding collisions in hash functions is well studied and the problem is considered to be secure against both classical and quantum computing attacks. The signatures generated by SPHINCS+ are notably larger compared to other schemes. It is suited to environments where resistance to quantum-computing attacks is a top priority. SPHINCS+ is being standardized as a stateless hash-based digital signature algorithm (SLH-DSA) within FIPS 205 [27].

Utilizing PQC algorithms carries risks and challenges, such as unproven security, performance, and regulatory and compliance uncertainties. Furthermore, PQC algorithm implementations have not had extended testing and may have weaknesses, for instance, certain architectures allowed the encapsulated key (generated from Kyber) to be recovered [12]. A strategy to address these risks is a multi-algorithm construction, combining and jointly operating one or more traditional algorithms with one or more post-quantum algorithms. This construction, Post-Quantum/Traditional (PQ/T) hybrid scheme, confronts the risks of operating a new algorithm at the cost of additional performance and storage.

2.2 Introduction to the Electrical Vehicle Charging Infrastructure

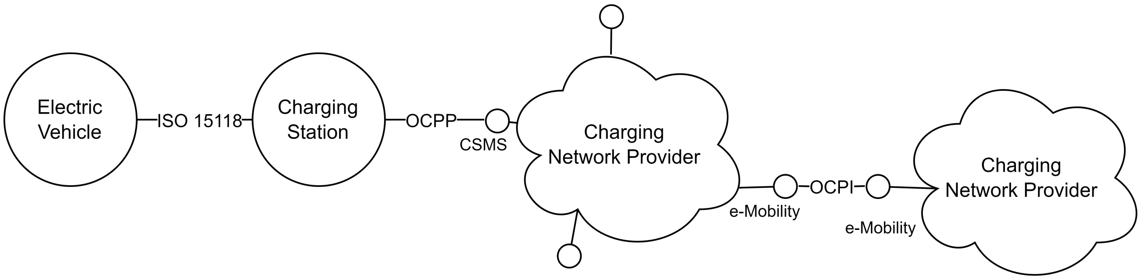

Figure 1. An architecture depiction of the EV ecosystem.

The EVCI refers to the charging stations and associated equipment needed to recharge the batteries of Electrical Vehicles (EVs). For this paper, a simplified model to represent EVCI is used, comprising just the charging station and the CSMS, illustrated in Figure 1. The EV charging station (CS), also known as an Electric Vehicle Supply Equipment (EVSE), is the device that transfers electricity from the local electrical supply to the EV. The CSMS, owned and operated by the Charging Network Provider (CNP), is designed to centrally control and monitor a network of EV charging stations. The CNP may also have an e-Mobility interface to support roaming, which allows EV drivers to access and use charging stations that are operated by different charging network providers using a single account or access method. Roaming enables EV drivers to conveniently charge their vehicles at various charging stations across different networks without needing separate memberships or accounts for each provider.

NEVI formula program identifies three key EVCI protocols, ISO 15118, the Open Charge Point Protocol (OCPP), and the Open Charge Point Interface (OCPI), that must be supported [19]. The EV and EVCS communications are governed by ISO 15118 [34, 35], a standard suite that covers the networking and signaling. The EVCS-to-CSMS communication uses the OCPP [36] protocol, which is designed to ensure interoperability between different brands and models of charging stations and management systems. This protocol allows for the seamless exchange of information, such as charging station status, transaction data, and remote control commands, enabling efficient operation and the remote monitoring and management of the EVCI. The OCPI [37] protocol aims to standardize and simplify the sharing of information across charging networks. This includes data related to charging station location, availability, pricing, and charging session initiation and billing.

ISO 15118, OCPP, and OCPI protocols employ common mechanisms for communication and message security. Each uses TLS, a cryptographic protocol designed to secure communication over untrusted networks. Moreover, they optionally employ digital signatures for purposes of authentication and identity operations. Throughout this effort, the team focused on the evaluation of TLS and digital signature applications identified in ISO 15118-20.

The EVCI consists of global players (suppliers, customers, legislation) and the innovation in the field has been explored by government-funded research,[2] environmental specialists,[3] and vehicle manufacturers.[4] For this effort, the PNNL team focused on the electrical vehicle component, the charging component, and the communication between the two. Specifically under test was an ARM Cortex-A7, ARM Cortex-A53, ARM Cortex-M4, IMX6Q (vSECC), and IMX8 chips, with ISO 11518-2, ISO 11518-20, Open Charge Point Protocol (OCPP), and Open Charge Point Interface (OCPI) protocols.

For more information on the components in the EVCI, the various industry players, or the applicable legislation, see any of the following [42] [43] [44].

[2] The Department of Energy's (DOE's) Vehicle Technology Office (VTO) has funded multiple government laboratories to investigate ways of lowering the cost of electric vehicles, the effectiveness of these vehicles (i.e. vehicle ranges), and future workforce training [38].

[3] The U.S. Environmental Protection Agency proposed various emissions standards to promote both vehicle manufacturers and consumers to transition to hybrid or fully-electric vehicles [39].

[4] Manufacturers such as Ford [40] and Toyota [41] have invested billions of dollars to design, manufacture, and promote electric versions of their vehicles.

3.0 Exploring the Challenge Areas

In this section, the key challenges, considerations, and factors that shape the PQC transition in the EV industry are explored. The exploration covers the effects of key sizes, ciphertext, and signature lengths on devices and network protocols, the demands placed on computing and memory resources, issues of interoperability, the feasibility of upgrades, organizational implications, and standardization hurdles.

3.1 Key, Ciphertext, and Signature Sizes

Table 4. Digital signature algorithm properties

PQC Security Level	Algorithm	Public Key Size (in bytes)	Private Key Size (in bytes)	Signature Size (in bytes)
Traditional	ECDSA P-256[a]	65	32	65
Traditional	ECDSA P-521[b]	133	66	133
Traditional	Ed448[b]	57	57	114
1	FALCON512	897	1281	666
2	Dilithium2	1312	2528	2410
3	Dilithium3	1952	4000	3293
5	FALCON1024	1793	2305	1280
5	Dilithium5	2592	4864	4595

[a] Algorithm specified in ISO 15118-2.

[b] Algorithm specified in ISO 15118-20.

The team investigated the impacts of the larger sizes generated from PQC algorithms compared to traditional algorithms. Table 4 provides an overview of the security levels, key sizes, and signature lengths associated with both traditional and PQC digital signature algorithms. It is evident from this comparison that PQC algorithms tend to produce significantly larger keys and signatures relative to their conventional counterparts. Specifically, when comparing PQC to ECDSA P-256—the default digital signature algorithm specified in ISO 15118-2, and ECDSA P-521—one of the defaults referenced in ISO 15118-20, it is noted that PQC public keys are approximately 14.7 to 44.5 times larger. Similarly, the sizes of signatures generated by PQC are observed to be 9.2 to 69.7

times larger, underscoring the marked increase in size associated with the adoption of PQC algorithms. The study of the impact of size variations is critical. According to a report published in the Computing Community Consortium Catalyst, “changes to key and hash output sizes, in practice, is highly impactful to widely deployed cryptography for data in motion and at rest and will require considerable engineering to make the transition” [45].

Table 5. The impact of the digital signature algorithm on the EVCC certificates

Algorithm	Leaf Certificate Size (in bytes)	Leaf + 1 CA Certificate Chain Size (in bytes)	Leaf + 2 CAs Certificate Chain Size (in bytes)
ECDSA P-256	502	1029	1562
ECDSA P-521	639	1301	1970
Ed448	514	1052	1596
FALCON512	1906	3839	5777
Dilithium2	4105	8234	12 369
Dilithium3	5618	11 260	16 908
FALCON1024	3416	6860	10 309
Dilithium5	7560	15 144	22 734
Dilithium2 (leaf) + FALCON1024 (CA)[a]	4105	6386	12 369

[a] Two digital signature algorithms are employed, Dilithium2 for the leaf and FALCON1024 for CA certificates.

Table 6. The impact of the digital signature algorithm on the SECC certificates

Algorithm	Leaf Certificate Size (in bytes)	Leaf+ 1 CA Certificate Chain Size (in bytes)	Leaf+ 2 CAs Certificate Chain Size (in bytes)	OCSP Response Size (in bytes)
ECDSA P-256	558	1085	1618	882
ECDSA P-521	692	1354	2023	1086
Ed448	568	1106	1650	931
FALCON512	1960	3893	5831	2870
Dilithium2	4159	8288	12 423	6838
Dilithium3	5672	11 314	16 962	9224
FALCON1024	3475	6919	10 368	4997
Dilithium5	7614	15 198	22 788	12 468
Dilithium2 (leaf) + FALCON1024 (CA)[a]	2999	6443	9894	5672

[a] Two digital signature algorithms are employed, Dilithium2 for the leaf and FALCON1024 for CA certificates.

A study was commenced to understand the implications of the larger sizes. For each digital signature algorithm detailed in Table 4, a corresponding PKI was established utilizing that specific algorithm. The PKI was configured in accordance to the criteria established in [35, Annex B]. The depth between the V2G Root CA and the leaf certificates is two, meaning that there are two intermediate CAs between the V2G Root CA and the leaf certificate. Across the PKIs, the certificates were identically issued, sharing the exact same metadata, except for variations in issue date and expiration date. The material differences in the certificates are then related to the differences in the public key and the issuer's signature. The certificate and certificate status sizes for each PKI are recorded in Table 5 and Table 6.

This notable increase in size necessitates a reevaluation and potential expansion of key and certificate stores. Key and certificate stores, which are essential components in securing private keys and verifying identities, must be adapted to handle the increased size requirements imposed by PQC. This involves not only expanding the storage capacity but also ensuring that the systems responsible for managing these stores can efficiently handle the larger data without compromising performance or security.

Moreover, the infrastructure supporting these stores may require enhancements to maintain quick access and retrieval times, vital for seamless authentication and encryption processes. This includes upgrading database systems, data structure optimization, and implementing more efficient algorithms for data management and retrieval. Additionally, considerations around data transmission and bandwidth usage become increasingly critical, as the larger key and certificate sizes impact network efficiency and data transfer costs.

Adapting key and certificate stores for PQC also entails revisiting backup, recovery, and archival strategies to ensure that the larger data volumes do not hinder the ability to recover from data loss or system failures.

Key, certificate, and signature sizes also play into plug and charge (PnC) and metering receipts. PnC, renamed park and charge in ISO 15118-20, is a feature within the ISO 15118 standard that simplifies the process of charging EVs. Imagine driving your EV to a charging station and simply plugging it in, without needing to use a card, a smartphone app, or any other form of manual payment and authentication. The charging station and your vehicle automatically recognize each other, authenticate, and take care of billing seamlessly in the background. This not only makes the charging process more convenient but also enhances security and efficiency, creating a smooth and user-friendly experience for EV owners. Metering receipts provide a record of the amount of electricity transferred and the cost incurred during a charging session.

Table 7. Lengths of AuthorizationReq and MeteringConfirmationReq

Algorithm	AuthorizationReq (in bytes)	MeteringConfirmationReq (in bytes)
ECDSA P-256	1915	497
ECDSA P-521	2392	565
Ed448	1991	538
Dilithium2	15 070	2844
Dilithium3	20 482	3717
FALCON512	6713	1081
Dilithium2+ FALCON1024	12 550	2855

The Electric Vehicle Communication Controller (EVCC) is required to provide a valid contract certificate to obtain charging authorization within the PnC framework. This is accomplished by sending a AuthorizationReq message to the Supply Equipment Communication Controller (SECC), which includes a digital signature and the corresponding contract certificate chain that the EVCC used to sign the message. An example of such a message is seen in Appendix F, Listing 2. Critically, the lengths of the certificates listed in Table 5 exceed the 1600-byte storage capacity of the AuthorizationReq message. The issue is further explored in Subsection 3.3, which outlines a strategy to surmount this limit.

In addition to the AuthorizationReq, the EVCC also provides a digital signature within the MeteringReq message. Unlike certificates, there is no signature length limit specified for either the AuthorizationReq or MeteringReq messages. The influence of the digital signature algorithms on the size of these encoded messages is recorded in Table 7. As an illustration of the effect, an AuthorizationReq message using Dilithium2 is 7.9 times larger than one using ECDSA P-256, and the MeteringConfirmationReq message is 5.7 times larger. The addition of certificates accounts for a 2.2-fold increase in size. However, opting for FALCON512 can mitigate this, resulting in a message size that is still 3.5 times larger than an ECDSA P-256-based AuthorizationReq, thereby offering a more size-efficient alternative.

The authors maintain that the larger messages do not pose a challenge for the EV-EVSE communications, which are capable of reaching a maximum data transfer rate of $5\,\mathrm{Mbit\,s^{-1}}$. Even in the most extreme case, where a message is tenfold larger, the additional time to transmit the message is only 0.030 s longer. Given that the SECC expects the response in at least that or more seconds,[5] the transfer time doesn't present a significant issue.

The Open Charge Point Protocol (OCPP) 2.0.1 incorporates the use of certificates for a variety of purposes, such as external identification means authorization and firmware verification. It features certificate management capabilities to streamline these processes. Given the expanding size requirements of PQC certificates, it's crucial to assess and modify these data types to ensure compatibility and effective communication within the OCPP framework.

OCPP 2.0.1 imposes length limits on types that are insufficient for storing PQC-based certificates, certificate signing requests, Online Certificate Status Protocol (OCSP) responses, and certificate chains. Specifically, the maximum size for certificates, signing requests, and OCSP responses is set to 5500 B, and for certificate chains, it extends up to 10 000 B [36]. It's important to note that these data types employ PEM encoding, which is a base64 encoding format. This format inherently increases the length of the certificate data by over one-third compared to the original DER encoding. Consequently, even a Dilithium2-based certificate, which is 4149 B in its DER form, swells to 5689 B when encoded in PEM format, surpassing the current limits set for the type.

OCPP can flag and report errors in requests but not responses. OCPP operates on a remote procedure call (RPC) framework, allowing a caller to execute a function on a remote system, known as the "callee". The process involves the caller dispatching a message (the call) and awaiting a response, which is either a call result or call error. In instances where the call involves large certificate data that exceeds the limit, the callee

[5] ISO 15518-20 [35, Table 215] specifies that the EVCC has up to 40 s to transmit AuthorizationRes. The SECC communicates the timeout for MeteringConfirmationRes using the NotificationMaxDelay parameter, denoted in seconds from message reception.

may return a call error, indicating PropertyConstrainViolation or TypeConstraintViolation error [46]. However, for oversized data in responses, the caller cannot report the failure to the callee, and will likely drop the response, silently ignore it, and, potentially, try again.

Table 8. The impact of digital signature algorithms on the TLS handshake

	TLS		mTLS	
Algorithm	Client Transmit Size (median, in bytes)	Server Transmit Size (median, in bytes)	Client Transmit Size (median, in bytes)	Server Transmit Size (median, in bytes)
ECDSA P-256	373	3400	2081	4613
ECDSA P-521	373	4076	2557	5578
Ed448	373	3524	2158	4769
FALCON512	373	10 185	6880	14 214
Dilithium2	373	22 532	15 237	30 945
Dilithium3	373	30 330	20 671	41 783
FALCON1024	373	17 472	12 028	24 501
Dilithium5	373	40 724	27 799	56 049
Dilithium2 (leaf) + FALCON1024 (CA)[a]	373	18 815	12 705	24 908

[a] Two digital signature algorithms are employed, Dilithium2 for the leaf and FALCON1024 for CA certificates.

Table 9. The impact of digital signature algorithms on the TLS handshake, without the certificate status extension

	TLS		mTLS	
Algorithm	Client Transmit Size (median, in bytes)	Server Transmit Size (median, in bytes)	Client Transmit Size (median, in bytes)	Server Transit Size (median, in bytes)
ECDSA P-256	364	2510	2072	3723
ECDSA P-521	364	2983	2548	4484
Ed448	364	2585	2149	3830
FALCON512	364	7307	6871	11 336
Dilithium2	364	15 664	15 228	24 077
Dilithium3	364	21 098	20 662	32 551
FALCON1024	364	12 459	12 019	19 497
Dilithium5	364	28 226	27 790	43 551
Dilithium2 (leaf) + FALCON1024 (CA)[a]	364	13 135	12 696	19 229

[a] Two digital signature algorithms are employed, Dilithium2 for the leaf and FALCON1024 for CA certificates.

Shifting attention to TLS 1.3, the TLS handshake is the process of creating a secure connection between parties. During the handshake, messages are exchanged to confirm the identities of the parties involved, negotiate encryption algorithms, and agree on session keys. One or more certificates are conveyed during this process. Each certificate is bound to a size limit of just under 16 MiB. The certificate sizes presented in Table 6 are significantly below this threshold. The handshake also sends signatures that are required to be shorter than 16 KiB. The length of signatures shown in Table 4 satisfy this requirement. Lastly, the key exchange shares must be less than 16 KiB. Again, the length of the shares shown in Table 10 are under this threshold.

Table 8 and Table 9 detail how digital signature algorithms influence the volume of data exchanged during the TLS 1.3 handshake. During the handshake, the volume of data transmitted was measured by varying the digital signature algorithm, while keeping the key exchange algorithm, Elliptic-Curve DiffieHellman (ECDHE) P-256, constant. The key difference between the tables is whether the certificate status extension is present: Table 8 features it, whereas Table 9 omits it. The certificate status extension is a feature in TLS 1.3 that allows the client to check the current status of the server's certificate to confirm it's still valid and hasn't been revoked. Unlike OCSP, the certificate status extension allows the server (the charger within ISO 15118) to send the certificate's revocation information directly to the client (the EV) during the TLS handshake, without the need for separate OCSP requests.

Table 10. Key exchange algorithm attributes

PQ Security Level	Algorithm	Client Key Exchange Share Size (in bytes)	Server Key Exchange Share Size (in bytes)
Traditional	ECDHE P-256[a]	65	65
Traditional	ECDHE P-521[b]	133	133
Traditional	X25519[c]	32	32
Traditional	X448[b]	57	57
1	Kyber512	800	768
3	Kyber768	1184	1088
3	X25519Kyber768[d]	1216	1120
5	Kyber1024	1568	1588

[a] Algorithm specified in ISO 15118-2.

[b] Algorithm specified in ISO 15118-20.

[c] TLS 1.3 de facto default.

[d] A concatenate PQ/T hybrid scheme combining X25519 and Kyber768.

Table 11, which features the certificate status extension, and Table 12, which omits it, showcase how different key exchange algorithms affect the amount of data exchanged during the TLS 1.3 handshake. The tables were created by varying the key exchange algorithm while the digital signature algorithm, ECDSA P-256, remained fixed. Notably, there is a significant increase in the volume of data transmitted compared to conventional

Table 11. The impact of key exchange algorithm has on the TLS handshake

	TLS		mTLS	
Algorithm	Client Transmit Size (median) (in bytes)	Server Transmit Size (median) (in bytes)	Client Transmit Size (median) (in bytes)	Server Transmit Size (median) (in bytes)
ECDHE P-256	373	3400	2081	4613
ECDHE P-521	441	3468	2149	4681
X25519	340	3367	2048	4580
X448	364	3391	2072	4604
Kyber768	1492	4423	3200	5636

Table 12. Key Exchanges have on TLS Handshake, without OCSP

	TLS		mTLS	
Algorithm	Client Transmit Size (median) (in bytes)	Server Transmit Size (median) (in bytes)	Client Transmit Size (median) (in bytes)	Server Transmit Size (median) (in bytes)
ECDHE P-256	364	2510	2072	3723
ECDHE P-521	432	2578	2140	3791
X25519	331	2477	2039	3690
X448	355	2501	2063	3714
Kyber768	1483	3533	3191	4746

key exchange algorithms. For example, Kyber768 results in transmitting 16.5 times and 7.5 times more data than ECDHE P-256 and ECDHE P-521.

The increased sizes of the cryptographic elements can affect TLS by prolonging the setup phase of the TLS connection. Once the connection is securely established, TLS transitions to symmetric cryptography, allowing high-performance and efficient encryption and decryption of bulk data. This means that once the connection is established, the performance of a session that was setup using PQC algorithms is indistinguishable from one using conventional cryptographic algorithms.

Table 13. Number of data TCP data segments exchanged during the TLS handshake[a]

Digital Signature Algorithm	X25519	Kyber768
ECDSA P-256	7	8
Ed448	7	8
Dilithium2	27	28
FALCON512	14	15

[a] Data segment refers to any TCP segment with a payload length greater than zero. The total number of these segments includes all data packets, up to and including the segment carrying the client's TLS Finished packet. Details on Finished can be had in [47].

The delay in the TLS connection establishment is due to four causes. Firstly, the transmission of larger keys, ciphertexts, and signatures inherently requires more time. Secondly, as highlighted in Table 13, more packets are needed, thereby incurring additional processing overhead. Each TCP segment requires processing by both the sender and the receiver (e.g., calculating and verifying checksums, managing sequence numbers, and acknowledging receipt). Thirdly, TCP's congestion control mechanisms come into play [48]. The increased number of TCP segments can activate these mechanisms. The initial congestion window on Linux systems is typically ten times the TCP's maximum segment size, approximately equal to 14 KiB. Given that each certificate is comprised of a public key and an issuer signature, and considering the inclusion of a key exchange key share, the total data often surpasses the initial congestion window. This excess prompts the sender to pause transmissions, waiting for an acknowledgment from the receiver before proceeding. Lastly, there is an increased probability of packet loss. Lost packets need to be retransmitted, and TCP's reliable delivery mechanism ensures that any lost segment is resent.

Besides causing delays, the increased size of the elements could lead to higher costs on metered connections, such as cellular links. However, the authors contend that the extra costs linked to PQC algorithms would be relatively minor. This assertion is reinforced by an analysis of OCPP, which is the most likely protocol to be used over metered connections. OCPP connections are typically long lived, extending over hours or days. The incremental costs attributed to PQC are mitigated by two key factors. Firstly, since handshakes occur infrequently, the larger initial handshake overhead can be amortized over the connection's lifespan. Secondly, the relative effect of the increased handshake size becomes less significant as the amount of data transmitted over the connection grows.

Efforts are ongoing to minimize the performance overhead associated with implementing PQC in TLS. Research conducted by Cloudflare has revealed that the size of the cryptographic elements significantly impacts the duration of the TLS handshake. Specifically, a 9 kB increase in size can result in a 15 percent slowdown, while a size of 10 kB or more can lead to a 60 percent reduction in completing the TLS handshake. This poses potential challenges for the performance of HTTP-based consumer services, including APIs and portals.

To address these concerns, NIST is actively exploring alternative signature schemes that could offer more efficient performance [49]. Many of these new schemes are promising because they produce shorter signatures [50], thereby potentially alleviating the network bottleneck.

Further innovations include strategies to eliminate the need for intermediate certificates, which can add unnecessary complexity and data overhead to the TLS process [51, 52]. Additionally, there are proposals to decrease the number of signatures required in certificates, such as incorporating Merkle tree-based approaches, which could further streamline the authentication process and enhance efficiency [53]. Another strategy being explored involves reducing the number of certificates housed in root certificate stores [52].

Next, the computing and memory requirements of PQC are evaluated.

3.2 Computational Resources

Table 14. System Attributes

System	Processor	Architecture	RAM	Internal Storage
Raspberry Pi 3 B+	4 x 1.4 GHz ARM Cortex-A53	64-bit	1 GiB	—
Beagle Bone	1 x 1 GHz ARM Cortex-A8	32-bit	512 MiB	—
NXP i.MX 6Quad	4 x 1.2GHZ ARM Cortex-A9	32-bit	2 GiB	7264 MiB
NXP i.MX 8M Nano	4 x 1.5 GHz ARM Cortex-A53	64-bit	2 GiB	29 820 MiB

One of the most prominent concerns about PQC algorithms is that their execution times are slower and require more computational resources such as memory, CPU, and storage, which is particularly concerning when these algorithms need to run on already constrained devices [45, 54, 55]. To assess the impact on resource-constrained systems, the team evaluated the algorithms running on four distinct system-on-modules (SoMs)[6] in three different exercises. The four SoMs utilized are ARM Cortex-A8, ARM Cortex-A9, NXP iMX 6Q, and NXP iMX 8M Nano. The attributes for each are listed in Table 14. The NXP i.MX 6Quad and NXP i.MX 8M Nano were identified as key components of commercially available charger control units, intended to be incorporated into high-power (greater than or equal to 100 kW) chargers compatible with the ISO 15118 standard. The charger control unit is the main controller, which handles the communication and management functions of a charging station. The BeagleBone's ARM Cortex-A8 serves as a representative example of another charger control unit accessible to the authors. Lastly, the Raspberry Pi 3 Model B+, with its Cortex-A9 processor, is frequently utilized in academic research settings.

Three tests were devised to evaluate the SoMs' ability to run PQC algorithms with respect to signature generation, signature verification, and TLS handshake. The TLS handshake is a composite operation, involving a key exchange, along with numerous signing and verifying operations. The timing and memory was recorded for each instance.

The tests were conducted using primitives and TLS foundations available in the OpenSSL (version 1.1.1u) and Open Quantum Safe (version 2023-07) cryptographic libraries. OpenSSL is an open-source library widely used for secure communication over networks while Open Quantum Safe provides a framework for the development and

[6] A SoM is a compact, fully integrated computer or electronic subsystem packaged into a single module. Think of it as a mini-computer that includes all the essential components, such as a processor, memory (RAM), and storage space on a small circuit board.

integration of PQC algorithms. The tests were programmed to the OpenSSL API, allowing the flexible runtime selection of cryptographic schemes. The tests were executed multiple times; appropriate runtime configurations were issued to select traditional or PQC routines for the specific instance of the test.

3.2.1 Algorithm Performance Using Standard Testing Libraries

Table 15. iMX6Q OpenSSL Speed Results

System	Algorithm	sign	verify	sign/s	verify/s
iMX6Q	P-256	0.0006s	0.0018s	1625.6	571
iMX6Q	P-521	0.0489s	0.0354s	20.5	28.2
iMX6Q	Ed448	0.0032s	0.0075s	312.1	133.3
iMX6Q	Dilithium2	0.0071s	0.0023s	141.5	442.3
iMX6Q	Dilithium3	0.0118s	0.0038s	84.9	263
iMX6Q	Dilithium5	0.0149s	0.0065s	66.9	152.8
iMX6Q	FALCON512	0.0424s	0.0007s	23.6	1502.9
iMX6Q	P-256 + Dilithium2	0.0080s	0.0041s	125.7	244.6
iMX6Q	P-256 + FALCON512	0.0431s	0.0025s	23.2	406
iMX6Q	P-384 + Dilithium3	0.0330s	0.0200s	30.3	49.9
iMX6Q	P-521 + Dilithium5	0.0633s	0.0432s	15.8	23.2

As part of the team's investigation, `openssl speed` was used to evaluate the performance of various classical and PQC algorithms on multiple platforms. `openssl speed` is part of the `openssl` package and is considered a performance test library. The team executed this library on the M1 Mac (bare metal), M1 Mac (in a docker container), Cortex A53, Cortex 8, iMX6Q, and iMX8M. As expected, the algorithms were most efficient on the M1 Mac (bare metal). Table A.1 contains the results across all of the SoMs, and Table 15 is a consolidated result set showing the comparison of the algorithms on the iMX6Q. The results highlight the negligible difference between P-256 and many of the PQC algorithms, and that many of the PQC algorithms are better performing than P-521. An additional observation resulting from this test was that the majority of the traditional algorithms performed the sign function more efficiently than the verify; however, with the PQC algorithms it's the opposite, where the verify function is more efficient than the sign. Commonly, the verify function is called more frequently than the sign function.

3.2.2 TLS Handshake

The TLS handshake test was programmed with a single thread of execution. Measurements collected during development showed that a single-thread execution meaningfully outperformed multi-threaded versions of the test. Examining the details of the TLS handshake, the authors reasoned that there was little opportunity to overlap compute and communication.

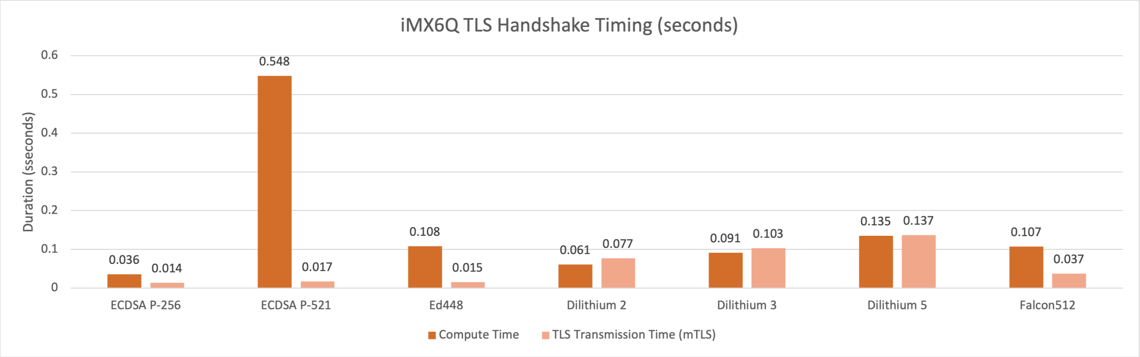

Figure 2. iMX6Q TLS Handshake Timing

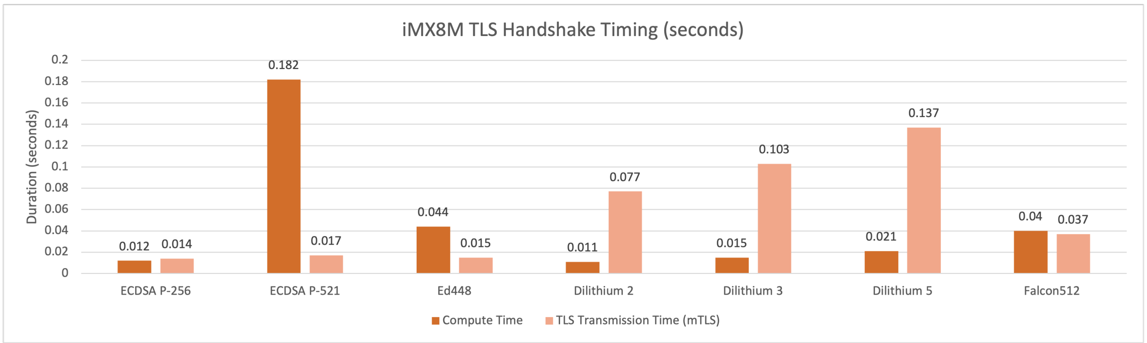

Figure 3. iMX8M TLS Handshake Timing

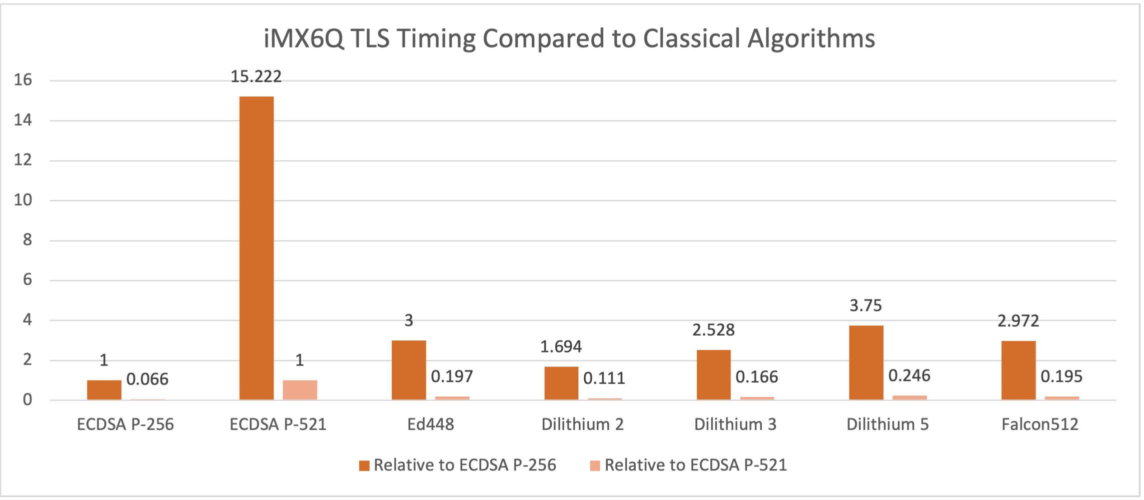

Figure 4. iMX6Q TLS Handshake Timing Compared to Classical Algorithms

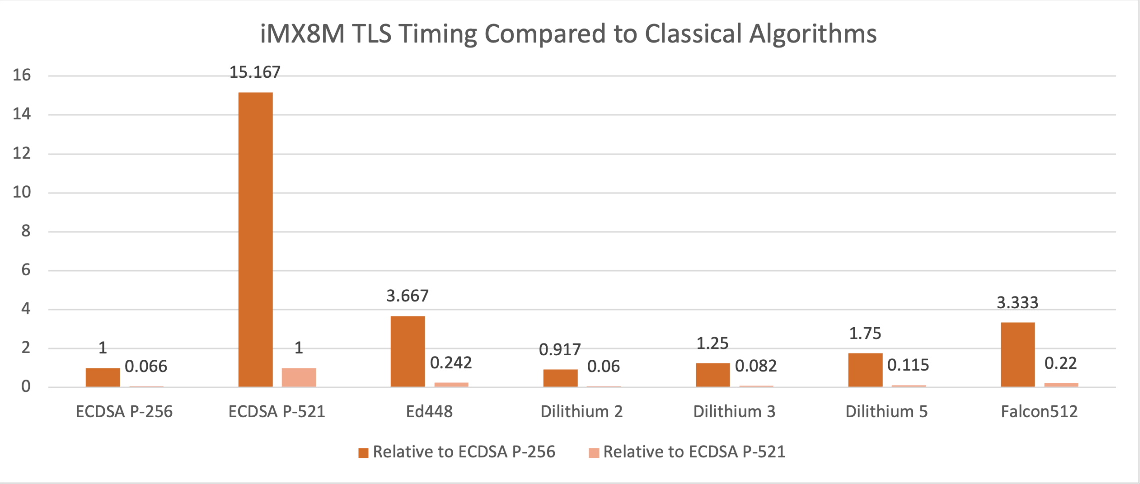

Figure 5. iMX8M TLS Handshake Timing Compared to Classical Algorithms

The thread repeatedly calls `SSL_do_handshake()` [56], alternating calls between the client and server contexts, until two successive calls return the value of 1, which signals the success of the handshake and the TLS connection establishment. The process includes not just the network communication, but also checking the structure, validity, the trustworthiness of the exchanged certificates, the ephemeral key material generation, and so on. The duration of the handshake is the time it takes until the series of

`SSL_do_handshake()` calls result in a connection establishment. Based off of the team's finding (shown in Figure 2 and Figure 3, see Appendices B, C, and D for full results), the PQC algorithms are generally slower than the traditional cryptographic algorithm P-256, but faster than P-521. Given a 5 Mbit s^{-1} communication link (the maximum transfer rate specified for power line communications used in charging scenarios), the time to transfer the cryptographic elements are slower than either P-256 and P-521. Interestingly, the time required for data transfer is roughly equivalent to the computation time. Figure 4 and Figure 5 compare the TLS timing of PQC algorithms against P-256 and P-521.

Table 16. Total Resident Memory Used During TLS Handshake

Algorithm	ARM Cortex A53 (in kB)	ARM Cortex-8 (in kB)	iMX6Q (in kB)	iMX8M (in kB)
ECDSA P-256	9756	8456	9000	9872
Ed448	9644	8412	8936	9644
Dilithium2	11 104	9964	10 540	11 260
Dilithium3	11 772	10 576	10 980	11 824
Dilithium5	12 176	11 008	11 500	12 304
FALCON512	10 648	9320	9844	10 652

Table 16 demonstrates that PQC cryptosystems required slightly more memory during the TLS handshake tests. However, the increase was minimal and stayed within the manageable and acceptable performance range for the evaluated SoMs.

In summary, the findings discussed in this section affirm the capability for the evaluated EVCI SoMs to effectively run PQC algorithms. In the TLS protocol, PQC algorithms are primarily utilized during the connection setup. For charging stations, TLS connections are kept active for extended periods of time, spanning several minutes to days, resulting in infrequent execution of PQC algorithms. Since there were no significant delays encountered, the user experience should not be degraded. Conversely, cloud-based charging network provider applications face the challenge of managing a large number of ephemeral connections but possess the necessary resources to meet these higher demands. The findings provided are from the perspective of a high-power charging infrastructure. Vehicles have further resource restrictions that deserve their own study. In line with ISO 15118, vehicles bear the responsibility for the digital signing of messages. Contrary to traditional ECC, PQC message signing is more costly than message verification, and the absence of double-precision support could favor Dilithium over FALCON in this case.

3.3 Interoperability

One of the biggest challenges faced by all industries is determining how they should handle security on legacy systems. The EV industry is no different. One of the biggest challenges in the transition to post-quantum resistant algorithms is that many of the approaches to integrate the new algorithms could not be deployed on vehicles already out on the road. Exacerbating the problem, NIST mentions that "experience has shown that, in the best case, 5 to 15 or more years will elapse after the publication of cryptographic standards before a full implementation of those 3 standards is completed. Unfortunately, the implementation of post-quantum public-key standards is likely to be more problematic than the introduction of new classical cryptographic algorithms. In the absence of significant implementation planning, it may be decades before the community replaces most of the vulnerable public-key systems currently in use" [14]. This can be seen in simpler crytographic transitions: the finance and payment industry isn't expected to make a full transition to AES, a block cipher that was standardized in 2001 [57], until 2030; and the SHA-1-to-SHA-2 hash migration is still in progress even though SHA-1 was deprecated in 2011 [54]. Therefore, the difficulty associated with securing EVCI is not only limited to vehicles already on the road, but vehicles in production for the next 5–15 years as well.

This pushes the EV charging community to be more cryptographically agile and for migration to a post-quantum resistant solution to be an immediate priority. Most industries (including EVCI) will likely need to go through two migrations, one from traditional cryptography to PQ/T (executing both a traditional and PQC algorithm), and then one from PQ/T to exclusively PQC. The necessity for multiple migrations highlights the need for cryptographically agile (`crypto-agile`) hardware and software solutions.

There are two potential approaches to crypto-agility, a software solution and a hardware solution. Hardware would be an in-field replacement (hardware swap-out) and should be avoided, if possible, due to cost. The systems are ideally built modularly to support hardware upgrades, ensuring they can be secured against unknown threats if a software solution isn't possible (also called "hardware ready"). The ideal solution is using software or firmware upgrades to incorporate new algorithms or to change configurations to be more secure. It isn't feasible to completely remove and replace entire systems in a vehicle, and while possible to remove and replace entire charging stations, it should be seen as a worst case scenario.

3.3.1 TLS 1.3

TLS 1.3's extensive crypto-agility diminishes the necessity for modifications to support additional key exchange and digital signature algorithms and research has been conducted to better defining the process and requirements for implementing TLS 1.3 according to ISO 15118-20 [58]. During the TLS handshake process, the client communicates to the server which groups and signature algorithms it can support. The server then selects the appropriate algorithms to be utilized throughout the session for both signing and certificate authentication. Given TLS 1.3 is designed to be algorithm-agnostic, incorporating PQC and PQ/T hybrid algorithms into its list of supported algorithms does not present a challenge [59]. This integration has already been implemented in current versions of the Chrome web browser [60].

If the EVCC is equipped with PQC capabilities but the SECC is not, an extra round trip will be required to establish the suitable key exchange key shares. This will likely delay the establishment of the TLS connection.

After the TLS connection has been established, the EVCC and SECC proceed to negotiate session parameters. If PnC is elected, the EVCC supplies a digital signature along with its corresponding contract certificate chain. As previously mentioned, there isn't sufficient capacity to hold post-quantum certificates. Consequently, compatibility issues that may be triggered by the PQC adoption need to be explored.

The ISO 15118 protocols face two technical hurdles in adopting PQC: the lack of support for the PQC digital signature algorithms, and restrictions on certificate length, with the latter posing the greater challenge. To accommodate multiple Vehicle-to-Grid (V2G) root certificate authorities within a market, ISO 15118-20 mandates the provision for storing at least two certificate authorities, as outlined in V2G20-1806 and V2G20-2352 [35].

Sending an AuthorizationReq message with post-quantum certificates would likely exceed the certificate size restrictions established by ISO 15118. Despite the maximum certificate length increasing from 800 bytes in the ISO 15118-2 XML schema to 1600 bytes in ISO 15118-20, all current PQC schemes exceed this requirement. For example, a FALCON512 certificate, the smallest quantum secure leaf certificate given in Table 5, is 1906 bytes, surpassing the ISO 15118-20 limit by 306 bytes. In this case, the SECC responds with FAILED, leading to the immediate termination of the charging session (shown in Figure 6).

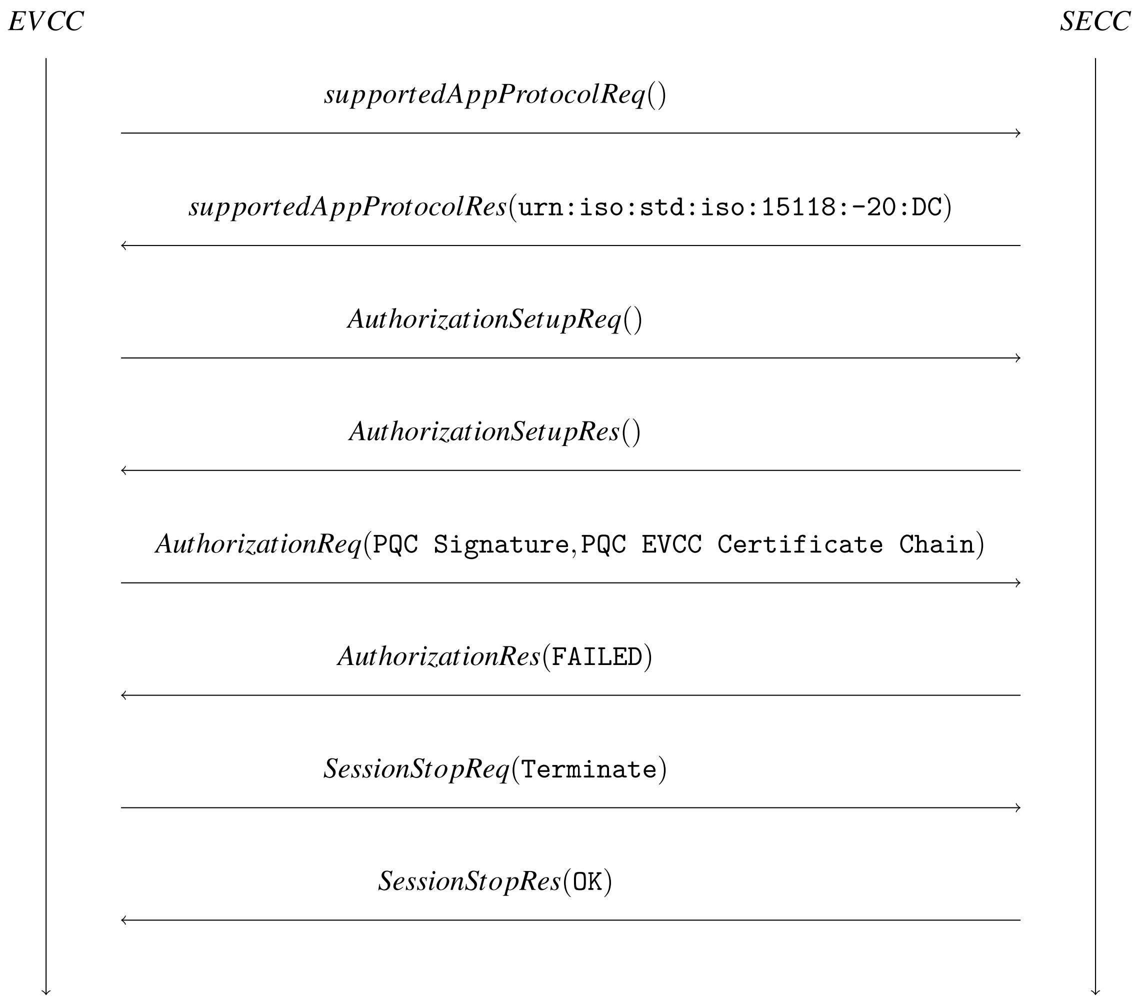
EVCC
SECC
supportedAppProtocolReq()
supportedAppProtocolRes(urn:iso:std:iso:15118:-20:DC)
AuthorizationSetupReq()
AuthorizationSetupRes()
AuthorizationReq(PQC Signature,PQC EVCC Certificate Chain)
AuthorizationRes(FAILED)
SessionStopReq(Terminate)
SessionStopRes(OK)

Figure 6. EVCC with PQC support, SECC without

A simple solution to address this error is to increase the certificateType maxLength to 12288 bytes[7] within the ISO 15118-20 XML schema. Both the invalid message (due to size) and the processing error (due to the algorithm) result in a FAILED response code that forces communication between the EVCC and SECC to be terminated. The message sequence for the failed messages can be found in Figure 6.

[7] 12288 was chosen to be larger than any identified certificates and to provide enough flexibility for additional metadata.

Listing 1. AppProtocolReq.xml example

```
<?xml version="1.0" encoding="UTF−8"?>
<supportedAppProtocolReq xmlns="urn:iso:15118:2:2010:AppProtocol" xmlns:xsi=
"http://www.w3.org/2001/XMLSchema−instance" xsi:schemaLocation=
"urn:iso:15118:2:2010:AppProtocol ./V2G_CI_AppProtocol.xsd">
    <AppProtocol>
        <ProtocolNamespace>urn:iso:std:iso:15118:−20:AC</ProtocolNamespace>
        <VersionNumberMajor>1</VersionNumberMajor>
        <VersionNumberMinor>0</VersionNumberMinor>
        <SchemaID>1</SchemaID>
        <Priority>3</Priority></AppProtocol>
    <AppProtocol>
        <ProtocolNamespace>urn:iso:std:iso:15118:−20:DC</ProtocolNamespace>
        <VersionNumberMajor>1</VersionNumberMajor>
        <VersionNumberMinor>0</VersionNumberMinor>
        <SchemaID>2</SchemaID>
        <Priority>2</Priority></AppProtocol>
    <AppProtocol>
        <ProtocolNamespace>
            urn:crypto:post−quantum::protocol#iso:std:iso:15118:−20:DC
        </ProtocolNamespace>
        <VersionNumberMajor>1</VersionNumberMajor>
        <VersionNumberMinor>1</VersionNumberMinor>
        <SchemaID>3</SchemaID>
        <Priority>1</Priority></AppProtocol>
    <AppProtocol>
        <ProtocolNamespace>urn:iso:15118:2:2013:MsgDef</ProtocolNamespace>
        <VersionNumberMajor>2</VersionNumberMajor>
        <VersionNumberMinor>0</VersionNumberMinor>
        <SchemaID>3</SchemaID>
        <Priority>4</Priority></AppProtocol>
</supportedAppProtocolReq>
```

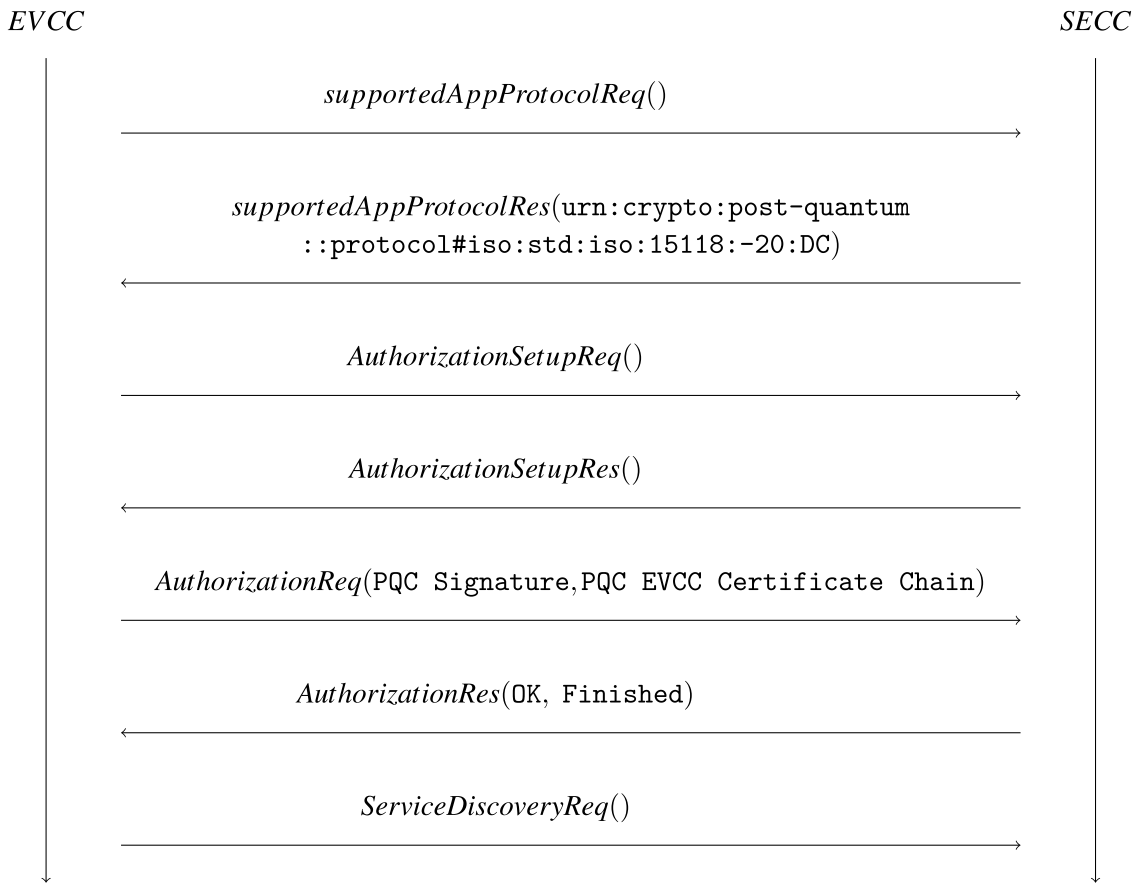
EVCC
SECC
supportedAppProtocolReq()
supportedAppProtocolRes(urn:crypto:post-quantum ::protocol#iso:std:iso:15118:-20:DC)
AuthorizationSetupReq()
AuthorizationSetupRes()
AuthorizationReq(PQC Signature,PQC EVCC Certificate Chain)
AuthorizationRes(OK, Finished)
ServiceDiscoveryReq()

Figure 7. EVCC and SECC with PQC support

Given the two solutions discussed, a new version or extension of ISO 15118 is needed to alter the supported algorithms list and XML schema. Creating an extension of ISO 15118-20 with PQC support would be the most straight-forward solution for transitioning with ease and scalability. At the beginning of a traditional session, a supportedAppProtocolReq/supportedAppProtocolRes is exchanged which allows the SECC to select a protocol from the EVCC's list of supported application protocols. The selected protocol will then be implemented for the rest of the session. Current ISO 15118-20 implementations include ISO 15118-20 Alternating Current (AC), ISO 15118-20 Direct Current (DC), and ISO 15118-2 messaging. When the extension is created, it can then be added to the supportedAppProtocolReq in instances where the EVCC supports the extension (PQC), as documented in Listing 1. If the SECC also supports PQC, it will send a response message selecting the Uniform Resource Name (URN) that represents the PQC extension (`urn:crypto:post-quantum::protocol#iso:std:iso:15118:-20:DC`) as shown in Figure 7.

Open Charge Point Protocol (OCPP) leverages the WebSocket subprotocol for version negotiation between charger and the CSMS. Through this subprotocol exchange, the charging station sends a list of supported OCPP versions to the CSMS, which then selects the most appropriate version and confirms its choice by echoing it back to the charging station. Similar to the ISO 15118, the process can be adapted to indicate support for PQC by appending specific tags like “ocpp2.0.1+pqc” to the subprotocol version identifier. Additionally, the charging station can introduce a specific variable to signal its capability that the CSMS can query. When this variable is present and set to true, it marks the station’s capable of PQC.

The PQC transition may require charging stations to support a range of certificate types to ensure broad vehicle compatibility. The OCPP certificate management capabilities may need to be revised to accommodate multiple certificates for a single use. For example, in the context of V2G communication, a charger maybe configure to offer three distinct certificate types: traditional, PQ, and PQ/T hybrid. Optimizing the management of these certificates through OCPP would be advantageous, streamlining the integration process.

3.4 Upgradability

The preceding sections demonstrate that the system-on-modules (SoMs) used in EVCI possess the necessary capabilities to effectively execute PQC. However, a potential obstacle to future upgrades lies in the hardware security modules (HSMs). These dedicated devices, designed to securely manage and protect cryptographic keys, might not be as easily adaptable to new standards or technologies, potentially limiting the system’s upgradability.

Generally, discrete HSMs are constrained devices, exhibiting limited flexibility to upgrade to new cryptographic systems. While FIPS 140-2 support is widespread [61], there is a notable lack of HSMs available in the market that are compliant with FIPS 204. This would be expected given that FIPS 204 is still in the draft stage and has not been finalized or widely adopted yet.

The industry might consider exploring alternatives such as SoftHSMs—software simulations of HSMs that run in Trusted Execution Environments (TEEs), or firmware-based Trusted Platform Modules (fTPM)—cryptoprocessors embedded within the system’s firmware. At the time of this paper, SoftHSM does not support FIPS 204 either, but these solutions could offer a more flexible pathway for upgrading cryptographic capabilities, adapting more readily to evolving standards and requirements, and offering a greater capacity to patch.

3.5 Organizational Implications

Amidst the large infrastructure deployments incentivized by the NEVI formula program, persuading EV sector stakeholders to adopt PQC poses significant obstacles. These challenges include a general shortage of personnel, particularly those with expertise in cryptographic algorithms, the fact that PQC algorithms are still undergoing development and testing, and the common preference for allocating investments toward new features that are more visible to the consumer compared to behind-the-scenes security enhancements. Although there has been a growing recognition of the importance of cybersecurity investments in recent years, the operational technology domain remains notably lagging.

This perspective is corroborated by research findings. A global study by DigiCert and the Ponemon Institute reveals that the main hurdles organizations face in transitioning any technology to PQC include limited resources (such as time, staff, and budget), uncertainty regarding the impact of quantum computing, a scarcity of experts in the field, ambiguous responsibilities and funding for the transition, the ongoing development and standardization of PQC algorithms, insufficient support from executive leadership, and a lack of adequate tools for facilitating the transition [62].

It was found in the Ponemon Institute's global study that only 52% of the participants' organizations kept an inventory of where their keys were stored and 58% of the participants said they didn't know how many keys or certificates they have [62]. This lack of inventory not only makes it difficult to renew and upgrade certificates and keys if they expire but also makes it nearly impossible to understand the full scope of what a migration to post-quantum entails and the resources that would be required. Exacerbating this problem, post-quantum algorithms contain many additional parameters and configuration options (compared to traditional algorithms), as well as new functionality such as state management and entropy [45], which makes the transition nontrivial.

Additionally, concerns about training needs and managing legacy systems add layers of complexity to the transition, emphasizing the multifaceted nature of shifting towards a quantum-safe EVCI.

4.0 Standardization

As the standardization of PQC algorithms progresses with the anticipation of final standards being released in 2024, several crucial standardization-related issues need to be resolved before widespread adoption can occur. Among these, the format of PQ/T hybrid certificates stands out as a key area requiring resolution. A PQ/T hybrid certificate contains public keys for two component algorithms, one being a traditional algorithm and the other for PQC [22]. The PQ/T hybrid approach is advantageous because it mitigates the uncertainties involved in adopting novel algorithms that lack a proven track record. Moreover, the approach reduces the risk of sudden, large-scale vulnerabilities.

As of this writing, five certificate formats are currently being considered, as outlined in [63]. Adopting a unified approach is crucial for the broad acceptance of PQ/T hybrid cryptosystems. Accepting a single certificate format would not only enhance interoperability and efficiency but also streamline the implementation of certificate-reliant software and certificate management practices.

Apart from NIST's effort to standardize post-quantum secure public-key cryptosystems, a recent call for proposals was launched to seek solutions aimed at expanding the range of algorithms available for general-purpose signature applications. While NIST, did not forbid the use of lattice-based solutions, they stated that "any structured lattice-based signature proposal would need to significantly outperform CRYSTALS-Dilithium and FALCON in relevant applications and/or ensure substantial additional security properties to be considered for standardization" [49]. The first round for proposals ended on June 1, 2023 and at the time of this writing, the submitted proposals are available for comment [50].

5.0 Conclusion

Based on the comprehensive exploration of the transition to post-quantum cryptography (PQC) within the electric vehicle charging infrastructure (EVCI) industry, it is evident that while significant strides have been made in advancing the adoption of electric vehicles to mitigate emissions and pollutants, there remains a critical gap in addressing cybersecurity measures.

The imminent threat posed by quantum computing to traditional public key cryptography (PKC) systems underscores the urgency of transitioning towards quantum-resistant cryptography. The risk of future decryption capabilities and digital signature forgery highlights the need for proactive measures to safeguard sensitive data, transactions, and the integrity of EVCI systems. Furthermore, the extended lifespan of EVs and infrastructure necessitates forward-thinking strategies to ensure that these systems remain secure against evolving cyber threats. The transition to PQC presents unique challenges, including interoperability, computational resources, and organizational readiness, which require careful consideration and planning. Efforts by organizations such as the National Institute of Standards and Technology (NIST) to standardize PQC systems and methodologies for firmware and software signing are crucial steps toward addressing these challenges. However, comprehensive strategies and early engagement from stakeholders are essential to minimize disruptions and ensure a smooth and cost-efficient shift to PQC for the EV charging sectors.

This effort concentrated on investigating anticipated challenges, both technical and organizational, and sought to comprehend the measures necessary to overcome these obstacles. According to the team's analysis, although PQC algorithms demand increased computational resources, they frequently exhibit performance levels comparable to P-256. Moreover, P-521 consistently lags behind PQC algorithms in performance. This leads the team to the conclusion that if the P-521 performance is accepted by the industry, the superior performance of PQC algorithms makes the discussion about their performance negligible. Based on these assessments, the team is confident in the feasibility of implementing PQC algorithms on EVCI hardware.

In conclusion, proactive measures must be taken to address the cybersecurity challenges posed by quantum computing and safeguard the integrity and security of EVCI systems. By adopting quantum-resistant cryptography and implementing comprehensive strategies, the EVCI can mitigate risks and ensure the resilience of its infrastructure against emerging cyber threats and technology.

References

[1] The White House. FACT SHEET: Biden-Harris Administration Announces New Private and Public Sector Investments for Affordable Electric Vehicles. Apr. 2023. URL: https://www.whitehouse.gov/briefing-room/statements-releases/2023/04/17/fact-sheet-biden-harris-administration-announces-new-private-and-public-sector-investments-for-affordable-electric-vehicles/?utm_source=link (visited on 12/04/2023).

[2] International Energy Agency. Global EV Outlook 2023. en-GB. Tech. rep. International Energy Agency, Apr. 2023, p. 140. URL: https://iea.blob.core.windows.net/assets/dacf14d2-eabc-498a-8263-9f97fd5dc327/GEVO2023.pdf (visited on 11/13/2023).

[3] Joint Office of Energy and Transportation. National Electric Vehicle Infrastructure Formula Program ANNUAL REPORT | Plan Year 20222023. Tech. rep. DOE/GO-102023-5905. US Department of Energy, July 2023, p. 77. URL: https://driveelectric.gov/files/nevi-annual-report-2022-2023.pdf (visited on 12/04/2023).

[4] Emily Grumbling and Mark Horowitz, eds. Quantum Computing: Progress and Prospects. Washington, D.C.: National Academies Press, Mar. 2019. ISBN: 978-0-309-47969-1. DOI: 10.17226/25196. URL: https://www.nap.edu/catalog/25196 (visited on 09/06/2023).

[5] Craig Gidney and Martin Ekerå. "How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits". en-GB. In: Quantum 5 (Apr. 2021). Publisher: Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften, p. 433. DOI: 10.22331/q-2021-04-15-433. URL: https://quantum-journal.org/papers/q-2021-04-15-433/ (visited on 09/06/2023).

[6] Mark Webber et al. The Impact of Hardware Specifications on Reaching Quantum Advantage in the Fault Tolerant Regime. arXiv:2108.12371 [quant-ph]. Nov. 2021. URL: http://arxiv.org/abs/2108.12371 (visited on 10/11/2023).

[7] Ro [D-CA-17 Rep. Khanna. H.R.7535 - 117th Congress (2021-2022): Quantum Computing Cybersecurity Preparedness Act. eng. legislation. Archive Location: 2022-04-18. Dec. 2022. URL: https://www.congress.gov/bill/117th-congress/house-bill/7535 (visited on 06/21/2023).

[8] The White House. Executive Order on Catalyzing Clean Energy Industries and Jobs Through Federal Sustainability. en-US. Dec. 2021. URL: https://www.whitehouse.gov/briefing-room/presidential-actions/2021/12/08/executive-order-on-catalyzing-clean-energy-industries-and-jobs-through-federal-sustainability/ (visited on 09/29/2023).

[9] US Department of Energy. DOE Invests $39 Million to Support a 21st Century Electric Grid. en. Aug. 2023. URL: https://www.energy.gov/gmi/articles/doe-invests-39-million-support-21st-century-electric-grid (visited on 11/10/2023).

[10] Yanning Ji et al. A Side-Channel Attack on a Hardware Implementation of CRYSTALS-Kyber. Publication info: Preprint. 2022. URL: https://eprint.iacr.org/2022/1452 (visited on 06/23/2023).

[11] Haocheng Ma et al. "Vulnerable PQC against Side Channel Analysis - A Case Study on Kyber". In: 2022 Asian Hardware Oriented Security and Trust Symposium (AsianHOST). Dec. 2022, pp. 1–6. DOI: 10.1109/AsianHOST56390.2022.10022165.

[12] Daniel J. Bernstein. KyberSlash: division timings depending on secrets in Kyber software. Jan. 7, 2024. URL: https://kyberslash.cr.yp.to/index.html (visited on 02/04/2024).

[13] National Institute of Standards and Technology. Post-Quantum Cryptography. EN-US. Jan. 2017. URL: https://csrc.nist.gov/projects/post-quantum-cryptography (visited on 11/13/2023).

[14] William Barker, William Polk, and Murugiah Souppaya. "Getting Ready for Post-Quantum Cryptography: Exploring Challenges Associated with Adopting and Using Post-Quantum Cryptographic Algorithms". eng. In: (Apr. 2021). DOI: 10.6028/NIST.CSWP.04282021. URL: https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.04282021.pdf.

[15] National Cybersecurity Center of Excellence. Migration to Post-Quantum Cryptography (PQC). eng. URL: https://www.nccoe.nist.gov/sites/default/files/2023-08/mpqc-fact-sheet.pdf.

[16] Ritik Bavdekar et al. "Post Quantum Cryptography: Techniques, Challenges, Standardization, and Directions for Future Research". In: CoRR abs/2202.02826 (2022). arXiv: 2202.02826. URL: https://arxiv.org/abs/2202.02826.

[17] NIST. Post-Quantum Cryptography (PQC). Feb. 15, 2024. URL: https://csrc.nist.gov/Projects/Post-Quantum-Cryptography (visited on 02/19/2024).

[18] Thomas E Carroll et al. Inventory of Public Key Cryptography in US Electric Vehicle Charging. eng. Tech. rep. PNNL-34843. Pacific Northwest National Laboratory (PNNL), Richland, WA (United States), Sept. 2023. URL: https://www.pnnl.gov/main/publications/external/technical_reports/PNNL-34843.pdf.

[19] National Electric Vehicle Infrastructure Standards and Requirements. 23 CFR Part 680. Feb. 2023. URL: https://www.ecfr.gov/current/title-23/chapter-I/subchapter-G/part-680.

[20] David Cooper et al. Recommendation for Stateful Hash-Based Signature Schemes. Tech. rep. NIST Special Publication (SP) 800-208. Oct. 2020. DOI: 10.6028/NIST.SP.800-208.

[21] National Security Agency. Announcing the Commercial National Security Algorithm Suite 2.0. Sept. 2022. URL: https://media.defense.gov/2022/Sep/07/2003071834/-1/-1/0/CSA_CNSA_2.0_ALGORITHMS_.PDF.

[22] Florence Driscoll. Terminology for Post-Quantum Traditional Hybrid Schemes. Internet-Draft draft-ietf-pquip-pqt-hybrid-terminology-02. Work in Progress. Internet Engineering Task Force, Feb. 2, 2024. 16 pp. URL: https://datatracker.ietf.org/doc/draft-ietf-pquip-pqt-hybrid-terminology/02/.

[23] Help Net Security. Q-Day: The problem with legacy public key encryption. en-US. July 2022. URL: https://www.helpnetsecurity.com/2022/07/15/legacy-public-key-encryption-problem/ (visited on 12/04/2023).

[24] NIST PQC Team. PQC Standardization Process: Announcing Four Candidates to be Standardized, Plus Fourth Round Candidates. July 5, 2022. URL: https://www.nist.gov/news-events/news/2022/07/pqc-standardization-process-announcing-four-candidates-be-standardized-plus (visited on 02/04/2024).

[25] NIST. Module-Lattice-based Key-Encapsulation Mechanism Standard. Tech. rep. FIPS 204. U.S. Department of Commerce, Aug. 24, 2023. DOI: 10.6028/NIST.FIPS.203.ipd.

[26] NIST. Module-Lattice-Based Digital Signature Standard. Tech. rep. FIPS 204. U.S. Department of Commerce, Aug. 24, 2023. DOI: 10.6028/NIST.FIPS.204.ipd.

[27] NIST. Stateless Hash-Based Digital Signature Standard. Tech. rep. FIPS 204. U.S. Department of Commerce, Aug. 24, 2023. DOI: 10.6028/NIST.FIPS.203.ipd.

[28] NIST. NIST to Standardize Encryption Algorithms That Can Resist Attack by Quantum Computers. Aug. 23, 2023. URL: https://www.nist.gov/news-events/news/2023/08/nist-standardize-encryption-algorithms-can-resist-attack-quantum-computers (visited on 02/04/2024).

[29] NIST. Post-Quantum Cryptography–Security (Evaluation Criteria). Jan. 3, 2017. URL: https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/evaluation-criteria/security-(evaluation-criteria) (visited on 02/04/2024).

[30] Roberto Avanzi et al. CRYSTALS-Kyber–Algorithm Specifications And Supporting Documentation (Version 3.02). Tech. rep. Aug. 4, 2021.

[31] Roberto Avanzi et al. CRYSTALS Cryptographic Suite for Algebraic Lattices. Feb. 2022. URL: https://pq-crystals.org/ (visited on 01/29/2024).

[32] Léo Ducas et al. CRYSTALS-Dilithium–Algorithm Specifications and Supporting Documentation (Version 3.1). Tech. rep. Feb. 8, 2021.

[33] Pierre-Alain Fouque et al. FALCON Fast-Fourier Lattice-based Compact Signatures over NTRU. URL: https://falcon-sign.info/ (visited on 01/29/2024).

[34] ISO 15118-2:2014. en. Apr. 2014. URL: https://www.iso.org/standard/55366.html (visited on 09/06/2023).

[35] ISO 15118-20:2022. en. Apr. 2022. URL: https://www.iso.org/standard/77845.html (visited on 09/06/2023).

[36] Open Charge Alliance. OCPP 2.0.1: Part 2—Specification (Edition 2 FINAL). Dec. 15, 2022.

[37] EVRoaming Foundation. OCPI 2.2.1: Open Charge Point Interface. Oct. 6, 2021.

[38] Batteries, Charging, and Electric Vehicles. en. URL: https://www.energy.gov/eere/vehicles/batteries-charging-and-electric-vehicles (visited on 12/04/2023).

[39] EPAs New Standards Will Accelerate Transition to Electric Vehicles | Article | EESI. URL: https://www.eesi.org/articles/view/epas-new-standards-will-accelerate-transition-to-electric-vehicles (visited on 12/04/2023).

[40] Ford Electric Vehicles | Ford Media Center. URL: https://media.ford.com/content/fordmedia/fna/us/en/media-kits/2021/electric-vehicles.html.html (visited on 12/04/2023).

[41] Nick Crisler. Toyota Expands Vehicle-to-Grid (V2G) Research with San Diego Gas & Electric Company Collaboration. en-US. Nov. 2023. URL: https://pressroom.toyota.com/toyota-expands-vehicle-to-grid-v2g-research-with-san-diego-gas-electric-company-collaboration/ (visited on 12/04/2023).

[42] US Environmental Protection Agency. Plug-in Electric Vehicle Charging: The Basics. en. Other Policies and Guidance. Sept. 2023. URL: https://www.epa.gov/greenvehicles/plug-electric-vehicle-charging-basics (visited on 01/29/2024).

[43] EVgo. EV201: How an Electric Vehicle Works! | EVgo. en. URL: https://www.evgo.com/ev-drivers/charging-basics/how-an-ev-works/ (visited on 01/29/2024).

[44] US Department of Energy. Federal and State Laws and Incentives. en. URL: https://afdc.energy.gov/laws/ (visited on 01/29/2024).

[45] David Ott, Christopher Peikert, and et al et. "Identifying Research Challenges in Post Quantum Cryptography Migration and Cryptographic Agility". In: CoRR abs/1909.07353 (2019). arXiv: 1909.07353. URL: http://arxiv.org/abs/1909.07353.

[46] Open Charge Alliance. OCPP 2.0.1: Part 4—JSON over WebSockets implementation guide (FINAL). Mar. 31, 2020.

[47] Eric Rescorla. The Transport Layer Security (TLS) Protocol Version 1.3. Tech. rep. Request for Comments (RFC) 8446. Internet Engineering Task Force, Aug. 2018.

[48] Bas Westerbaan. Sizing Up Post-Quantum Signatures. Nov. 8, 2021. URL: https://blog.cloudflare.com/sizing-up-post-quantum-signatures (visited on 02/19/2024).

[49] NIST. Call for Additional Digital Signature Schemes for the Post-Quantum Cryptography Standardization Process. Oct. 2022. URL: https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/call-for-proposals-dig-sig-sept-2022.pdf (visited on 04/08/2024).

[50] NIST. Post-Quantum Cryptography: Digital Signature Schemes. Feb. 22, 2024. URL: https://csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures (visited on 04/08/2024).

[51] Dimitrios Sikeridis et al. Intermediate Certificate Suppression in Post-Quantum TLS: An Approximate Membership Querying Approach. Publication info: Published elsewhere. 18th International Conference on emerging Networking EXperiments and Technologies (CoNEXT '22). 2022. URL: https://eprint.iacr.org/2022/1556 (visited on 12/21/2023).

[52] David Adrian. Mar. 22, 2024. URL: https://dadrian.io/blog/posts/pqc-signatures-2024/ (visited on 04/08/2024).

[53] David Benjamin, Devon O'Brien, and Bas Westerbaan. Merkle Tree Certificates for TLS. Internet-Draft draft-davidben-tls-merkle-tree-certs-02. Work in Progress. Internet Engineering Task Force, Mar. 4, 2024. 49 pp. URL: https://datatracker.ietf.org/doc/draft-davidben-tls-merkle-tree-certs/02/.

[54] Matt Campagna, Brian A. LaMacchia, and David Ott. "Post Quantum Cryptography: Readiness Challenges and the Approaching Storm". In: CoRR abs/2101.01269 (2021). arXiv: 2101.01269. URL: https://arxiv.org/abs/2101.01269.

[55] Jay Johnson et al. Cybersecurity for Electric Vehicle Charging Infrastructure. English. Tech. rep. SAND2022-9315. Sandia National Lab. (SNL-NM), Albuquerque, NM (United States), July 2022. DOI: 10.2172/1877784. URL: https://www.osti.gov/biblio/1877784 (visited on 11/13/2023).

[56] SSL_do_handshake. Feb. 9, 2024. URL: https://www.openssl.org/docs/manmaster/man3/SSL_do_handshake.html (visited on 02/12/2024).

[57] NXP. The Emergence of Post-Quantum Cryptography. eng. Feb. 2021. URL: https://www.nxp.com/docs/en/white-paper/POSTQCRYPTOWP.pdf (visited on 01/02/2024).

[58] Ahmet Kilic. "TLS-handshake for Plug and Charge in vehicular communications". In: Computer Networks 243 (2024), p. 110281. ISSN: 1389-1286. DOI: 10.1016/j.comnet.2024.110281.

[59] Douglas Stebila, Scott Fluhrer, and Shay Gueron. Hybrid key exchange in TLS 1.3. Sept. 7, 2023. URL: https://datatracker.ietf.org/doc/draft-ietf-tls-hybrid-design/ (visited on 02/14/2024).

[60] Devon O'Brien. Protecting Chrome Traffic with Hybrid Kyber KEM. Aug. 10, 2023. URL: https://blog.chromium.org/2023/08/protecting-chrome-traffic-with-hybrid.html (visited on 02/14/2024).

[61] Infineon Technologies AG. SLS37 V2X HSM. en. URL: https://www.infineon.com/cms/en/product/security-smart-card-solutions/security-controllers/security-controllers-for-automotive-applications/sls37-v2x-hsm/ (visited on 01/29/2024).

[62] Ponemon Institute. Preparing for a Safe Post Quantum Computing Future: A Global Study. eng. Oct. 2023. URL: https://www.digicert.com/news/digicert-global-study-preparing-for-a-safe-post-quantum-computing-future.

[63] NIST. Migration to Post-Quantum Cryptography Quantum Readiness: Testing Draft Standards—Volume C: Quantum-Resistant Cryptography Technology Interoperability and Performance Report. Tech. rep. NIST Special Publication (SP) 1800-38C, draft. US Department of Commerce, Dec. 2023.

Appendix A – OpenSSL Speed Results

Table A.1: OpenSSL Speed

system	algorithm	sign	verify	sign/s	verify/s
iMX6Q	P-256	0.0006s	0.0018s	1625.6	571
iMX8M	P-256	0.0001s	0.0005s	6712.2	2098.5
Mac M1	P-256	0.0000s	0.0001s	46 876.2	16 358.2
Mac M1 Docker	P-256	0.0000s	0.0001s	33 723	10 488.2
ARM Cortex 8	P-256	0.0000s	0.0001s	46 876.2	16 358.2
ARM Cortex A53	P-256	0.0002s	0.0005s	5417.7	1828.7
iMX6Q	P-521	0.0489s	0.0354s	20.5	28.2
iMX8M	P-521	0.0154s	0.0117s	65.1	85.7
Mac M1	P-521	0.0025s	0.0022s	406.9	463.5
Mac M1 Docker	P-521	0.0022s	0.0017s	462.6	593.4
ARM Cortex 8	P-521	0.0025s	0.0022s	406.9	463.5
ARM Cortex A53	P-521	0.0186s	0.0143s	53.8	69.8
iMX6Q	Ed448	0.0032s	0.0075s	312.1	133.3
iMX8M	Ed448	0.0020s	0.0029s	495.2	348.6
Mac M1	Ed448	0.0016s	0.0018s	626.4	541.7
Mac M1 Docker	Ed448	0.0002s	0.0002s	5308.8	4565
ARM Cortex 8	Ed448	0.0016s	0.0018s	626.4	541.7

Continued on next page

OpenSSL Speed (continued)

system	algorithm	sign	verify	sign/s	verify/s
ARM Cortex A53	Ed448	0.0026s	0.0037s	391	269.4
iMX6Q	Dilithium2	0.0071s	0.0023s	141.5	442.3
iMX8M	Dilithium2	0.0009s	0.0003s	1168	3485
Mac M1	Dilithium2	0.0001s	0.0000s	15 208	49 258.4
Mac M1 Docker	Dilithium2	0.0004s	0.0001s	2246.5	10 408.7
ARM Cortex 8	Dilithium2	0.0001s	0.0000s	15 208	49 258.4
ARM Cortex A53	Dilithium2	0.0011s	0.0004s	916	2742.8
iMX6Q	Dilithium3	0.0118s	0.0038s	84.9	263
iMX8M	Dilithium3	0.0014s	0.0005s	736.2	2098.4
Mac M1	Dilithium3	0.0001s	0.0000s	9846.4	31 843.3
Mac M1 Docker	Dilithium3	0.0007s	0.0002s	1380.3	6553.6
ARM Cortex 8	Dilithium3	0.0001s	0.0000s	9846.4	31 843.3
ARM Cortex A53	Dilithium3	0.0017s	0.0006s	584.4	1654.8
iMX6Q	Dilithium5	0.0149s	0.0065s	66.9	152.8
iMX8M	Dilithium5	0.0017s	0.0008s	573	1216.2
Mac M1	Dilithium5	0.0001s	0.0001s	8109.6	19 657.4
Mac M1 Docker	Dilithium5	0.0009s	0.0002s	1146.1	4103.9
ARM Cortex 8	Dilithium5	0.0001s	0.0001s	8109.6	19 657.4
ARM Cortex A53	Dilithium5	0.0022s	0.0010s	452.1	961.3

Continued on next page

OpenSSL Speed (continued)

system	algorithm	sign	verify	sign/s	verify/s
iMX6Q	FALCON512	0.0424s	0.0007s	23.6	1502.9
iMX8M	FALCON512	0.0153s	0.0003s	65.3	3472.9
Mac M1	FALCON512	0.0002s	0.0000s	6319.5	42 895.7
Mac M1 Docker	FALCON512	0.0044s	0.0000s	225.8	25 020.8
ARM Cortex 8	FALCON512	0.0002s	0.0000s	6319.5	42 895.7
ARM Cortex A53	FALCON512	0.0194s	0.0004s	51.6	2761.7
iMX6Q	P-256 + Dilithium2	0.0080s	0.0041s	125.7	244.6
iMX8M	P-256 + Dilithium2	0.0010s	0.0008s	989	1286.5
Mac M1	P-256 + Dilithium2	0.0001s	0.0001s	11 452.9	12 226.7
Mac M1 Docker	P-256 + Dilithium2	0.0005s	0.0002s	2115.3	5106
ARM Cortex 8	P-256 + Dilithium2	0.0001s	0.0001s	11 452.9	12 226.7
ARM Cortex A53	P-256 + Dilithium2	0.0013s	0.0010s	766.3	1019.7
iMX6Q	P-256 + FALCON512	0.0431s	0.0025s	23.2	406
iMX8M	P-256 + FALCON512	0.0155s	0.0008s	64.3	1295
Mac M1	P-256 + FALCON512	0.0002s	0.0001s	5535	11 787.4
Mac M1 Docker	P-256 + FALCON512	0.0045s	0.0001s	223.2	7318.4
ARM Cortex 8	P-256 + FALCON512	0.0002s	0.0001s	5535	11 787.4
ARM Cortex A53	P-256 + FALCON512	0.0196s	0.0010s	51.1	1025.3
iMX6Q	P-384 + Dilithium3	0.0330s	0.0200s	30.3	49.9

Continued on next page

OpenSSL Speed (continued)

system	algorithm	sign	verify	sign/s	verify/s
iMX8M	P-384 + Dilithium3	0.0076s	0.0055s	131.6	183
Mac M1	P-384 + Dilithium3	0.0013s	0.0011s	787.3	897.3
Mac M1 Docker	P-384 + Dilithium3	0.0016s	0.0009s	625.4	1139.2
ARM Cortex 8	P-384 + Dilithium3	0.0013s	0.0011s	787.3	897.3
ARM Cortex A53	P-384 + Dilithium3	0.0097s	0.0070s	103.3	143.8
iMX6Q	P-521 + Dilithium5	0.0633s	0.0432s	15.8	23.2
iMX8M	P-521 + Dilithium5	0.0172s	0.0127s	58	78.8
Mac M1	P-521 + Dilithium5	0.0026s	0.0023s	387.7	443.7
Mac M1 Docker	P-521 + Dilithium5	0.0030s	0.0019s	336.3	531.7
ARM Cortex 8	P-521 + Dilithium5	0.0026s	0.0023s	387.7	443.7
ARM Cortex A53	P-521 + Dilithium5	0.0218s	0.0161s	45.9	62.2

Appendix B – TLS Handshake Time Elapsed—mTLS Results

Table B.2. TLS Handshake Time Elapsed, mTLS Server Digital Signature

Algorithm	ARM Cortex A53 (sec)	ARM Cortex-8 (sec)	iMX6Q (sec)	iMX8M (sec)
ECDSA P-256	0.015	0.048	0.036	0.012
ECDSA P-521	0.229	0.685	0.548	0.182
Ed448	0.055	0.14	0.108	0.044
Dilithium2	0.017	0.08	0.061	0.011
Dilithium3	0.024	0.122	0.091	0.015
Dilithium5	0.033	0.184	0.135	0.021
FALCON512	0.051	0.11	0.107	0.04

Table B.3. TLS Handshake Time Elapsed, mTLS Client Digital Signature

Algorithm	ARM Cortex A53 (sec)	ARM Cortex-8 (sec)	iMX6Q (sec)	iMX8M (sec)
ECDSA P-256	0.011	0.036	0.026	0.009
ECDSA P-521	0.167	0.5	0.4	0.132
Ed448	0.04	0.1	0.076	0.032
Dilithium2	0.013	0.063	0.048	0.009
Dilithium3	0.018	0.095	0.071	0.012
Dilithium5	0.025	0.14	0.102	0.016
FALCON512	0.048	0.104	0.102	0.038

Table B.4. TLS Handshake Time Elapsed, mTLS Server Key Exchange

Algorithm	ARM Cortex A53 (sec)	ARM Cortex-8 (sec)	iMX6Q (sec)	iMX8M (sec)
ECDHE P-256	0.015	0.048	0.036	0.012
ECDHE P-521	0.071	0.22	0.173	0.056
Ed448	0.023	0.065	0.049	0.018
x25519	0.015	0.05	0.036	0.012
Kyber768	0.014	0.049	0.036	0.011

Table B.5. TLS Handshake Time Elapsed, mTLS Client Key Exchange

Algorithm	ARM Cortex A53 (sec)	ARM Cortex-8 (sec)	iMX6Q (sec)	iMX8M (sec)
ECDHE P-256	0.011	0.035	0.026	0.009
ECDHE P-521	0.085	0.265	0.21	0.067
Ed448	0.021	0.056	0.042	0.017
x25519	0.011	0.037	0.026	0.008
Kyber768	0.01	0.037	0.027	0.008

Appendix C – TLS Handshake Time Elapsed—NoMTLS Results

Table C.6. TLS Handshake Time Elapsed, No mTLS Server Digital Signature

Algorithm	ARM Cortex A53 (sec)	ARM Cortex-8 (sec)	iMX6Q (sec)	iMX8M (sec)
ECDSA P-256	0.01	0.075	0.025	0.008
ECDSA P-521	0.146	0.913	0.352	0.116
Ed448	0.036	0.1	0.072	0.029
Dilithium2	0.011	0.052	0.039	0.008
Dilithium3	0.015	0.078	0.059	0.01
Dilithium5	0.021	0.115	0.085	0.014
FALCON512	0.028	0.108	0.059	0.022

Table C.7. TLS Handshake Time Elapsed, No mTLS Client Digital Signature

Algorithm	ARM Cortex A53 (sec)	ARM Cortex-8 (sec)	iMX6Q (sec)	iMX8M (sec)
ECDSA P-256	0.01	0.076	0.025	0.008
ECDSA P-521	0.147	0.914	0.352	0.117
Ed448	0.037	0.1	0.072	0.029
Dilithium2	0.011	0.053	0.04	0.008
Dilithium3	0.015	0.078	0.059	0.01
Dilithium5	0.021	0.115	0.086	0.014
FALCON512	0.028	0.109	0.059	0.022

Table C.8. TLS Handshake Time Elapsed, No mTLS Server Key Exchange

Algorithm	ARM Cortex A53 (sec)	ARM Cortex-8 (sec)	iMX6Q (sec)	iMX8M (sec)
ECDHE P-256	0.01	0.033	0.025	0.008
ECDHE P-521	0.066	0.203	0.162	0.052
Ed448	0.018	0.05	0.039	0.015
x25519	0.01	0.035	0.025	0.008
Kyber768	0.01	0.033	0.025	0.008

Table C.9. TLS Handshake Time Elapsed, No mTLS Client Key Exchange

Algorithm	ARM Cortex A53 (sec)	ARM Cortex-8 (sec)	iMX6Q (sec)	iMX8M (sec)
ECDHE P-256	0.01	0.034	0.025	0.008
ECDHE P-521	0.084	0.261	0.208	0.067
Ed448	0.021	0.054	0.042	0.017
x25519	0.01	0.035	0.025	0.008
Kyber768	0.01	0.035	0.026	0.008

Appendix D – TLS Handshake Time Elapsed—Hybrid Results

Table D.10. TLS Handshake Time Elapsed, mTLS Server Digital Signature

Algorithm	ARM Cortex A53 (sec)	ARM Cortex-8 (sec)	iMX6Q (sec)	iMX8M (sec)
ECDSA P-256+FALCON512	0.061	0.15	0.134	0.048
ECDSA P-256+Dilithium2	0.12	0.418	0.335	0.092
ECDSA P-384+Dilithium3	0.12	0.418	0.335	0.092
ECDSA P-521+Dilithium5	0.257	0.863	0.676	0.2

Table D.11. TLS Handshake Time Elapsed, mTLS Client Digital Signature

Algorithm	ARM Cortex A53 (sec)	ARM Cortex-8 (sec)	iMX6Q (sec)	iMX8M (sec)
ECDSA P-256+FALCON512	0.055	0.13	0.12	0.043
ECDSA P-256+Dilithium2	0.087	0.309	0.248	0.067
ECDSA P-384+Dilithium3	0.087	0.309	0.248	0.067
ECDSA P-521+Dilithium5	0.187	0.633	0.494	0.146

Table D.12. TLS Handshake Time Elapsed, mTLS Server Key Exchange

Algorithm	ARM Cortex A53 (sec)	ARM Cortex-8 (sec)	iMX6Q (sec)	iMX8M (sec)
x25519+Kyber768	0.016	0.054	0.039	0.012
ECDHE P-256+Kyber768	0.016	0.053	0.039	0.013

Table D.13. TLS Handshake Time Elapsed, mTLS Client Key Exchange

Algorithm	ARM Cortex A53 (sec)	ARM Cortex-8 (sec)	iMX6Q (sec)	iMX8M (sec)
x25519+Kyber768	0.011	0.044	0.031	0.009
ECDHE P-256+Kyber768	0.012	0.043	0.031	0.01

Table D.14. TLS Handshake Time Elapsed, No mTLS Server Digital Signature

Algorithm	ARM Cortex A53 (sec)	ARM Cortex-8 (sec)	iMX6Q (sec)	iMX8M (sec)
ECDSA P-256+FALCON512	0.034	0.186	0.076	0.027
ECDSA P-256+Dilithium2	0.076	0.568	0.214	0.059
ECDSA P-384+Dilithium3	0.076	0.568	0.214	0.059
ECDSA P-521+Dilithium5	0.163	1.149	0.428	0.128

Table D.15. TLS Handshake Time Elapsed, No mTLS Client Digital Signature

Algorithm	ARM Cortex A53 (sec)	ARM Cortex-8 (sec)	iMX6Q (sec)	iMX8M (sec)
ECDSA P-256+FALCON512	0.034	0.187	0.077	0.027
ECDSA P-256+Dilithium2	0.076	0.568	0.214	0.059
ECDSA P-384+Dilithium3	0.076	0.568	0.214	0.059
ECDSA P-521+Dilithium5	0.163	1.15	0.428	0.128

Table D.16. TLS Handshake Time Elapsed, No mTLS Server Key Exchange

Algorithm	ARM Cortex A53 (sec)	ARM Cortex-8 (sec)	iMX6Q (sec)	iMX8M (sec)
x25519+Kyber768	0.011	0.04	0.028	0.008
ECDHE P-256+Kyber 68	0.011	0.039	0.028	0.009

Table D.17. TLS Handshake Time Elapsed, No mTLS Client Key Exchange

Algorithm	ARM Cortex A53 (sec)	ARM Cortex-8 (sec)	iMX6Q (sec)	iMX8M (sec)
x25519+Kyber768	0.011	0.043	0.03	0.009
ECDHE P-256+Kyber768	0.011	0.042	0.03	0.009

Appendix E – TLS Handshake—Resident Memory

Table E.18. Stack Resident Memory Used During TLS Handshake

Algorithm	ARM Cortex A53 (kB)	ARM Cortex-8 (kB)	iMX6Q (kB)	iMX8M (kB)
ECDSA P-256+Dilithium2	60	56	56	60
ECSDA P-256+FALCON512	52	48	48	52
ECDSA P-384+Dilithium3	88	84	84	88
ECDSA P-521	16	12	8	12
ECDSA P-521+Dilithium5	128	124	124	128

Appendix F – ISO 15118-20 Message Examples

Listing 2. AuthorizationReq example

```
<SelectedAuthorizationService>PnC</SelectedAuthorizationService>
  <PnC_AReqAuthorizationMode v2gct_cm:Id="ID1">
    <GenChallenge>U29tZSBSYW5kb20gRGF0YQ==</GenChallenge>
    <ContractCertificateChain>
        <Certificate>
MIIB8jCCAZmgAwIBAgIUepXpYMLexjk3A8KzyVlBKxEM5oAwCgYIKoZIzj0EAwIwODELMAkGA1UE
CwwCQ0ExKTAnBgNVBAMMIHFIQ1FaZzQ0M3pNWXViWmlLYVB2UURQU0p4Nnhlc05lMB4XDTIzMDkx
MTE5MjA1NFoXDTI0MDkxMDE5MjA1NFowOjENMAsGA1UECwwERVZDQzEpMCcGA1UEAwwgdkFpZVFV
ZUZOQzZMZ3ROQWFGSndLN2g4ZmIzRDFIbmUwWTATBgcqhkjOPQIBBggqhkjOPQMBBwNCAATmZ56F
L7dkczuxrKm5bxrtxB/kWupucOiqVWqgS3zOhpjufk0XauypLz5XmXhVCQ7Xn4G0ZliRsWbFUxTh
9ZU2o38wfTAMBgNVHRMBAf8EAjAAMB0GA1UdDgQWBBTbCs0O26Wl27Llez9rgCe4JnGm4TAfBgNV
HSMEGDAWgBSxGg9+BU5+A11wd1vkVlpZfQC1iDAOBgNVHQ8BAf8EBAMCB4AwHQYDVR0lBBYwFAYI
KwYBBQUHAwIGCCsGAQUFBwMBMAoGCCqGSM49BAMCA0cAMEQCIC3kSI1UOkxB7BJ8YLJexAZD4JBF
0j/tqMjqvZPJ6UJtAiAHkYivAulSyBG61Zv1DftmjoWM14H9L4rEjoE71q8bNA==</Certificate>
      <SubCertificates>
        <Certificate>
MIICCzCCAbGgAwIBAgIUVLryZ7GXYTSWteUrnyP9+vFzgrEwCgYIKoZIzj0EAwIwODELMAkGA1UE
CwwCQ0ExKTAnBgNVBAMMIDZzZUlaQVFISGR6cVBES3hpajl1N1M0T3FyaGd4cWd0MB4XDTIzMDkx
MTE5MjA1M1oXDTMzMDkwODE5MjA1M1owODELMAkGA1UECwwCQ0ExKTAnBgNVBAMMIHFIQ1FaZzQ0
M3pNWXViWmlLYVB2UURQU0p4Nnhlc05lMFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAE+zK13Se1
gKkFNeaAmEfTeEVT7f02uUkJkFOTdxWJTpmWjZjdyzINufWesY7cz9wuyUchIPn//V4cwiRX4Xg2
KKOBmDCBlTAdBgNVHQ4EFgQUsRoPfgVOfgNdcHdb5FZaWX0AtYgwHwYDVR0jBBgwFoAUM7XPobNj
IY7nRrpFPvmVU2RWef8wEgYDVR0TAQH/BAgwBgEB/wIBADALBgNVHQ8EBAMCAQYwMgYIKwYBBQUH
AQEEJjAkMCIGCCsGAQUFBzABhhZodHRwOi8vbG9jYWxob3N0Ojk5OTgvMAoGCCqGSM49BAMCA0gA
MEUCIQC9ELvukd103DyuA1E3CJUqkfmhoOGiMziTTd6mz51GRwIgfju92Df5P8nf79rWeOVbsgfQ
aQaDVITH1OPYQb0vOAA=</Certificate>
        <Certificate>
MIICCjCCAbGgAwIBAgIUUFx5KKWeCgMdumdASWezn6nrPVwwCgYIKoZIzj0EAwIwODELMAkGA1UE
CwwCQ0ExKTAnBgNVBAMMIHZpZFJING5hd243MTJJRDdzQmJxcUFUMkNIRnVIWXF2MB4XDTIzMDkx
MTE5MjA1M1oXDTMzMDkwODE5MjA1M1owODELMAkGA1UECwwCQ0ExKTAnBgNVBAMMIDZzZUlaQVFI
SGR6cVBES3hpajl1N1M0T3FyaGd4cWd0MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEueir2nvE
+NyT1yFUkMXB2mcwIHL3by7FqTRKwGPJYkjtXA87BjaHgMFSK0jT9V2fxxQYtJ2cGG3lw9hSC4F5
p6OBmDCBlTAdBgNVHQ4EFgQUM7XPobNjIY7nRrpFPvmVU2RWef8wHwYDVR0jBBgwFoAUd+jBLNAZ
NCRhNoYMkNPRjK3L54YwEgYDVR0TAQH/BAgwBgEB/wIBATALBgNVHQ8EBAMCAQYwMgYIKwYBBQUH
AQEEJjAkMCIGCCsGAQUFBzABhhZodHRwOi8vbG9jYWxob3N0Ojk5OTcvMAoGCCqGSM49BAMCA0cA
MEQCIBMhEFpMz0Jg/AVXlqhVxqlYnAeszYYLGDAkN7HowYjXAiAEI/ura5v+ghPAsuLbIYLk3MVZ
sNwaEAim2e28O8zMJg==</Certificate>
      </SubCertificates>
    </ContractCertificateChain>
  </PnC_AReqAuthorizationMode>
```

Pacific Northwest
National Laboratory

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99354
1-888-375-PNNL (7665)

www.pnnl.gov