Drive (Quantum) Safe! – Towards PQ Authentication for V2V Communications

Nina Bindel[1]
SandboxAQ
nina.bindel@sandboxaq.com

Geoff Twardokus[1]
Rochester Institute of Technology
geoff.twardokus@mail.rit.edu

Sarah McCarthy
IQC and University of Waterloo
sarah.mccarthy@uwaterloo.ca

Hanif Rahbari
Rochester Institute of Technology
rahbari@mail.rit.edu

Abstract

We tackle a challenging problem at the intersection of two emerging technologies: post-quantum cryptography (PQC) and vehicle-to-vehicle (V2V) communication with its strict requirements. We are the first to devise and evaluate a practical, provably secure design for integrating PQ authentication into the IEEE 1609.2 V2V security ecosystem. By theoretically and empirically analyzing the three PQ signature algorithms selected for standardization by NIST, as well as XMSS (RFC 8391), we propose a *Partially Hybrid* design—a tailored fusion of classical cryptography and PQC—for use during the nascent transition period to PQC. As opposed to a direct substitution of PQC for classical cryptography, our design meets the unique constraints of standardized V2V protocols.

1 Introduction

Connected vehicle (CV) technologies are among the safety requirements of advanced driver-assistance and future autonomous driving systems [48], and are further integral to emerging intelligent transportation systems. Those technologies are proliferating globally under the umbrella of vehicle-to-everything (V2X) communication, where vehicle-to-vehicle (V2V) communication is expected to have a large ($20 billion [75]) share of this market. V2V has the potential to drastically reduce roadway collisions [52] and, at the same time, increase transportation system efficiency (reduce travel time, pollution, etc.). It requires each vehicle to regularly broadcast safety messages containing travel information (location, heading, etc.), facilitating proactive movement coordination by other vehicles. Thousands of vehicles on the road today are already equipped with V2V modules [16] and their adoption rate continues to accelerate; e.g., Ford and other major automakers intend to begin an immediate, nationwide deployment of V2V on new vehicles sold in the U.S. as soon as regulatory approval is granted [26]. This matches the increasing pace of standardization efforts, as IEEE and 3GPP envision expanded use cases of V2V (platooning, sensor data sharing, 3D mapping, etc.) to increase safety, efficiency, and autonomy [8, 42, 65]. An overhaul of the IEEE protocol for V2V–Dedicated Short Range Communications (DSRC)–is being finalized [42] and its alternative–Cellular V2X (C-V2X)–continues to be a priority of 3GPP in its upcoming releases towards 6G [4]. Simultaneously, autonomous vehicles are improving [68]; by the 2040s to 2060s, autonomous driving systems are expected to be standard on most new vehicles [47].

As more vehicles and services are expected to count on V2V, the security of V2V communications will become more critical. The IEEE 1609.2 and 1609.2.1 standards for CV security [1, 39, 40] describe the protocols that are required for authenticating messages broadcast by vehicles (and potentially also Unmanned Aerial Systems [30]), including V2V certificates. Currently, these standards rely solely on elliptic curve cryptography (ECC) [39], and particularly on the Elliptic Curve Digital Signature Algorithm (ECDSA) for signing V2V messages (to protect against spoofing, alteration, and replay attacks). Notably, 1609.2 security protocols are used in both C-V2X and DSRC systems, providing uniform security—or, potentially, vulnerability—across the V2V domain.

Ensuring the security of a vehicle throughout its 12-15 year expected lifetime [17] is very challenging, in part because consumers are unlikely to respond to recalls for hardware security upgrades [54]. Moreover, many experts believe there is a 50% or greater chance that a quantum computer capable of breaking "classical" cryptography like ECDSA will be developed as soon as 2037 [51]—15 years from today. Thus, we are already pressing up against the point in time when we have to secure connected vehicles against the emerging quantum threat. If this threat is not addressed, the safety and security of drivers and passengers traveling in tens of millions of vehicles that use safety-critical V2V applications will be put at risk, as the ability of a quantum attacker to forge certificates and signatures will, e.g., enable an attacker to mislead vehicles, causing massive traffic gridlock, or even manipulate vehicle movements to cause severe, possibly fatal crashes.

It will be particularly difficult to mitigate the quantum

[1]Corresponding authors.

threat in V2V after the classical security primitives that vehicles support are eventually compromised by quantum attacks because the *hardware* security modules of vehicles on the road cannot simply be disenrolled from the V2V system, adapted to new cryptographic primitives using over-the-air updates, or successfully recalled as long as vehicles lack *crypto agility*. At the same time, simply rolling out new vehicles with only Post-Quantum (PQ) support (once possible), while most vehicles on the road support only classical cryptography, is likely impractical, even more so because, as we will show, the PQ algorithms recently selected for standardization by the National Institute of Standards and Technology (NIST) cannot be used in a plug-and-play manner with current V2V systems. All of this means it is imperative that V2V security standards employ *quantum-secure*, *crypto-agile*, and *backwards-compatible* designs that not only ensure interoperability with both existing and future vehicles, but also maintain the current (classical) security guarantees in case first-generation PQ algorithms turn out to be insecure. To this end, standardization agencies including NIST recommend transitioning to PQ security using *hybrid* (also called *composite* or *dual*) schemes [9, 33] that combine classical and PQ algorithms in one design [13].

Hybrid designs have been widely explored in related domains [5, 6, 18, 21, 27, 29, 31, 46, 61, 67, 70–72, 76], but such works do not generally comply with the stringent latency and message size requirements of V2V communications and often rely on bidirectional algorithm negotiation that is not possible with unidirectional V2V safety message broadcasts. Besides, existing works on PQ in wireless systems [6, 71] focus on data aggregation and confidentiality, and research into post-quantum cryptography (PQC) on embedded devices [18, 31, 70, 72, 76] is mostly considered only over a more reliable, wired channel. Most notably, existing PQ solutions for vehicles [5, 27, 61] generally consider only intra-vehicle communications, or specific properties like access control in V2V, and do not address the unique challenges of *authenticating* messages among vehicles, as follows.

In a V2V system, all vehicles must broadcast their safety messages both quickly and frequently (on the order of milliseconds) while maximizing the number of vehicles that can transmit over the shared and limited 5.9 GHz channel, strictly constraining *signing time* and *transmission length*. This means, the *size* of each message with its signatures and certificates must be minimized when possible and further must not exceed the limit of the underlying V2V communication protocol. Additionally, incoming messages must be verified and processed within a few milliseconds of arrival, constraining signature *verification time*. Taken together, such a practical hybrid design for V2V not only needs to support above requirements by (1) minimizing signing and verification run times and (2) complying with V2V protocol constraints on frame size, it must also (3) support a *generalizable* technique that is not bound to particular PQ algorithms and (4) facilitate backwards compatibility with legacy vehicles whose hardware cannot support PQC. Incorporating PQC—with its very large key sizes, lengthy digital signatures, or often significant signing/verification times—is a nontrivial and challenging problem that was not anticipated when V2V security standards were initially developed.

Contributions— We take steps toward future quantum-secure V2V (PQ-V2V) by proposing a practical, scalable solution for the first era, shown in Figure 1. Depending on technological advances in quantum computing and V2V communications, production times and user affluence, the switch to PQ-V2V may take years or even decades and consists of several stages we identify in this paper.

Starting today, era A_v lasts until ECDSA can be broken in fewer than v hours, where v is a variable that depends both on quantum and V2V technology advances. Based on exponential extrapolation from the 2017–2025 quantum computing power data provided by IBM [37] together with state-of-the-art estimates [73] of the number of necessary qubits, we expect ECDSA will be breakable within $v = 1$ hour circa 2036. The second era (era B), which may overlap with era A_v, will begin once the hardware of older vehicles on the road cannot yet support PQ authentication but new vehicles have become reliant on PQC. Although it is difficult to predict when PQ hardware will be widely available, companies are already beginning to develop PQ co-processors for embedded devices [59] and Hardware Security Modules [22], so we hypothesise era B will begin around the mid 2030s. The last transition will occur when ECDSA is disallowed (era C), which we estimate to be approximately 7 years after ECDSA is deemed broken[2].

Our new design for V2V authentication is carefully tailored and instantiated for the most urgent era A_v. As technological advances become very unpredictable further in the future, we refrain from presenting designs for the other eras as that would be premature. Our *key contribution* is to show that PQC, despite its apparent incompatibility with V2V communication (due to large signatures and keys), can in fact be integrated with IEEE 1609.2 for use during the A_v era. Specifically, we make the following contributions:

Hybrid Designs for V2V Authentication: Given the constraints above, we devise and instantiate a practical, 1609.2-compatible hybrid design (called *Partially Hybrid*), wherein we utilize PQ security only to secure the integrity of the ECDSA key so as to provide strong protection during era A_v. This protects against quantum adversaries who are not able to break ECDSA in the v hours that a certificate is valid, while also maintaining all current 1609.2 security guarantees.

Infeasibility of Alternative Designs: We show that simply replacing or combining ECDSA with any of the three recent NIST selections for PQ standardization [58] or the alternative eXtended Merkle Signature Scheme (XMSS) algorithm [35] would not meet V2V frame size constraints. Furthermore, we

[2]For instance, 3DES, the predecessor to AES [55], was broken by the Sweet32 attack [12] in 2016 and NIST will not disallow the algorithm until 2023 [56]. Hence we conjecture that ECDSA will be disallowed circa 2043.

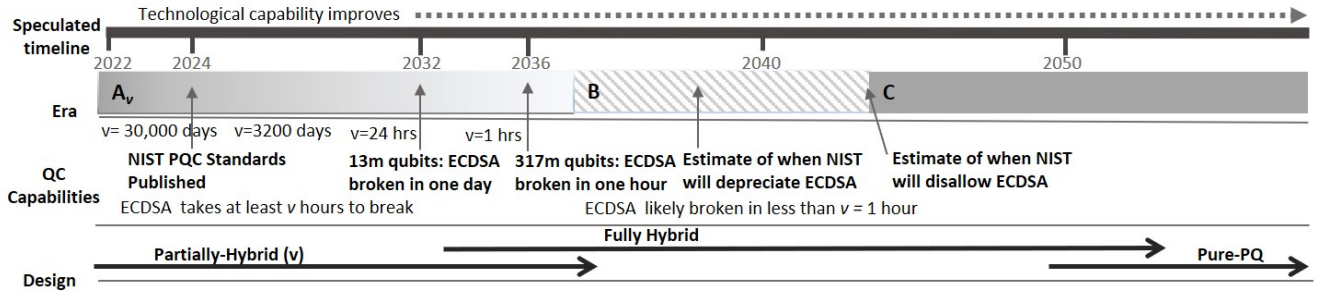

Figure 1: Estimated timeline of the PQ transition period and how our proposed eras and designs align.

outline and compare a *Fully Hybrid* design whose security relies on the security of both ECDSA and PQC. While such a design is desirable in the medium term (i.e., after 2032), we show that it will remain inviable as long as V2V hardware and communication protocols are not upgraded to more PQ-compliant versions and will need to be modified somewhat for integration with future V2V technology.

Paper Organization—After providing necessary background in Section 2, we compare key properties of our *Partially Hybrid* design with alternatives, showing the necessity of our tailored solution in Section 3. We present our *Partially Hybrid* design in Section 4, including an explanation of our threat model, description of suitable PQ instantiations, and informal discussion of its security. We conclude with related works in Sections 5.

2 A Primer on V2V Security

To achieve the safety benefits of V2V for proximity awareness, every vehicle broadcasts a digitally signed basic safety message (BSM) a minimum of once every 100 ms. Each BSM contains motion and position information to allow other vehicles to coordinate their movements and avoid collisions. Every BSM is signed and packed, along with the security information needed for verification, into a Secure Protocol Data Unit (SPDU) (see Figure 2). A frame containing the SPDU is then broadcast using either DSRC [38] or C-V2X [2], two similarly decentralized protocols based on different communication technologies. Below, we describe the V2V communication protocols and security mechanisms for BSMs.

2.1 V2V Security Standards

Security requirements and services for both DSRC and C-V2X are defined in IEEE 1609.2-2016 [39, 40] and IEEE 1609.2.1-2022 [43]. Among other things, 1609.2 specifies

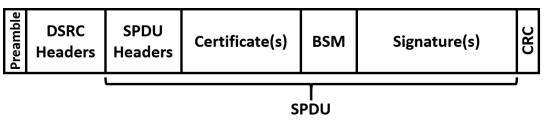

Figure 2: SPDU structure in a physical-layer frame of DSRC.

asymmetric cryptographic mechanisms and algorithms to securely exchange BSMs, while 1609.2.1 specifies certificate management and revocation requirements for vehicles. Figure 3 illustrates secure V2V communication under 1609.2. Of particular relevance, ECDSA is mandated to generate signatures (using either the `NIST P-256` or `brainpoolP256r1` elliptic curves) [39]. Beyond signatures, IEEE 1609.2 uses compact *pseudonym* certificates—in which the permanent user identity is replaced with a cryptographically unlinkable, ephemeral identifier—to protect the integrity of the public signature verification keys included in SPDUs. Pseudonym certificates are rotated every five minutes[3] while each one is (currently) valid for at most one week [15]– striking a balance between privacy and efficiency. For revocation, a single entry on a certificate revocation list can be used to efficiently revoke a large number of pseudonym certificates under 1609.2.1 [43]. It has been shown that this mechanism can be adapted to support PQ certificate revocation in V2X [10].

The 1609.2 standard defines both explicit and implicit certificates. Each *explicit* certificate contains a complete verification key and a signature over it by an issuer (e.g., a Certificate Authority (CA)), whereas an *implicit* certificate includes only a shorter *reconstruction value* from which the complete verification key can be derived using a trusted root certificate. Implicit pseudonym certificates generated using the classical scheme in IEEE 1609.2 (Elliptic Curve Qu-Vanstone (ECQV) [19]) help minimize SPDU size: under IEEE 1609.2, one SPDU is at most 226 bytes using implicit vs. 330 bytes using explicit certificates. Since PQ implicit certificates that are smaller than PQ explicit certificates have not yet been devised, we consider only explicit certificates in our design.

Under current industry standards [63], a vehicle generally includes its full pseudonym certificate only in every fifth SPDU and transmits a hash of that certificate in the other 80% of messages. This minimizes the number of large frames and consequently maximizes system capacity (see Section 3). From this implicit acceptance of ignoring up to 4 BSMs before one arrives that can be verified and accepted, we assume in our designs that a complete certificate must be transmitted over the course of every 500 ms interval (i.e., across the trans-

[3]IEEE 1609.2 does not define this, but five minutes is a common estimate [15].

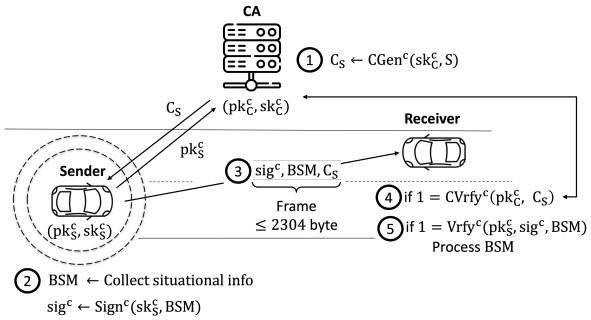

Figure 3: Current V2V security protocol: (1) Issuer generates a pseudonym certificate over the vehicle's ECDSA key pk_S^c. (2) The vehicle signs a BSM using ECDSA. (3) The frame containing that BSM is broadcast. (4) The receiving vehicle verifies the pseudonym certificate using a certificate chain (potentially with P2PCD). (5) If the signature is valid, the BSM is processed.

mission of 5 BSMs) to continue to meet current expectations. We call this period a *five-message cycle* and incorporate it as a critical element of our *fragmentation* method (in particular, fragmentation of large certificates that can then be transmitted using several SPDUs, see Section 4.2).

IEEE 1609.2 further defines the peer-to-peer certificate distribution (P2PCD) protocol to support verification of pseudonym certificates. When a pseudonym certificate is being verified, its issuer's certificate must also be verified, and so on until this *certificate chain* is verified all the way up to a self-signed trusted root certificate. During this process, a vehicle will generate a P2PCD *learning request* if it encounters an unknown certificate and attach that request to its next outgoing SPDU [39, Sec. 8.1]. Any vehicle receiving this request that has knowledge of the requested certificate then generates a *learning response*. After a wait time randomly chosen from the discrete uniform distribution between $0-250$ ms (see Section 4.4), if the vehicle has not heard at least 3 other vehicles broadcast a learning response containing that certificate [64], it broadcasts its learning response. This process works well for ECDSA, whose certificates are small enough that a learning response can fit within a single payload, but breaks down completely when certain PQ algorithms are used instead. As we discuss in Section 4.4, this alone excludes some PQ algorithms since P2PCD is mandatory under 1609.2.

2.2 V2V Communication Technologies

DSRC and C-V2X are the two major V2V protocols used around the world defined for the physical and Media Access Control (MAC) layers. DSRC, an IEEE 802.11 protocol tailored for the high-mobility V2V environment, is the dominant V2V protocol and de facto standard in Europe [32], more than 100,000 vehicles are on the roads in Japan [66], and, for the time being [25]), the majority of V2V-equipped vehicles in the U.S. use only DSRC. We focus in this paper on DSRC

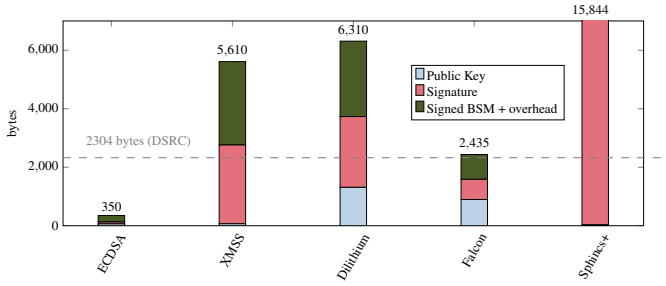

Figure 4: PQC sizes vs. DSRC payload constraint.

to develop our designs due to its current dominance in several places in the world and the incompatibility of the current C-V2X design with current PQC, as we elaborate below.

A significant obstacle to integrating PQC into V2V is the maximum size of frame payloads. Under DSRC, payloads are capped at 2,304 bytes regardless of data rate (i.e., modulation and coding scheme) or channel bandwidth [41, Table 9-25]. This is already a limiting constraint on any PQ-V2V design, but it is still one order of magnitude higher than the limit in C-V2X, a similarly decentralized protocol based on 4G and 5G cellular technology. In C-V2X, the maximum payload size (considering only *practical* data rates) in a standard 10 MHz channel is a mere 437 bytes [3, Table A.8.3-1]. As this is insufficient to contain even a PQ signature—let alone signature and a public key, typically in the orders of 100s of bytes for any of the three PQ algorithms recently standardized by NIST (see Section 3), C-V2X as the latest 3GPP specifications is unable to practically support post-quantum security. Adapting C-V2X to suit PQC, or vice versa, would require significant changes in the design of C-V2X, FCC allocations, and/or standard PQ algorithms, which are out of scope for this paper.

3 Elementary but Inviable PQ Designs

Due to V2V's strict size constraints alone, replacing ECDSA with post-quantum alternatives (in a purely PQ or classical-PQ hybrid solution) is not viable, as we explain in this section.

PQ Signature Algorithms. We consider the three candidate PQ signature algorithms recently selected for standardization by NIST—Falcon, Dilithium, and Sphincs$^+$—in addition to the stateful XMSS [35] scheme. While both XMSS and Leighton-Micali Hash-Based Signature Scheme (LMS) [44] are approved in the draft of FIPS 186-5 [62], we chose to evaluate only XMSS as its slightly smaller keys and signatures [45] make it more likely to be viable in V2V. We further give an analysis of the former alternate candidate Picnic in Appendix C, but we do not analyze the other former NIST candidates (Rainbow and GeMSS) due to recent and devastating attacks against them [11, 69].

We chose the following instantiations, corresponding to NIST's security Level 1: `Falcon-512` [60], `Dilithium-II` [49], and `Sphincs`$^+$`-128s` [36]. On the other

Table 1: PQC performance and resulting system capacity vs. ECDSA on Qualcomm chipset; $\bar{x}$ and σ denote mean and standard deviation, respectively, over 1,000 executions.

Sign (microseconds)					
Algorithm	$\bar{x}$	σ	vs. ECDSA	Hz	Acceptable?
ECDSA	123	26	1.00	8130	Yes
Falcon	31009	172	252.11	32	Yes
Sphincs$^+$	5.59e6	2914	45447.15	0.18	No
Dilithium	4076	2614	33.13	245	Yes
XMSS	5.0e11	4.0e9	4.1e9	1.9e-6	No
Verify (microseconds)					
Algorithm	$\bar{x}$	σ	vs. ECDSA	Hz	$v_{\max}$
ECDSA	272	13	1.00	3676	581
Falcon	555	23	2.04	1801	180
Sphincs$^+$	5585	206	20.53	179	17
Dilithium	1021	33	3.75	979	97
XMSS	6786	158	24.9	147	14

hand, we used `XMSS-SHA2_16_256`, a Level 5 instantiation, because this was necessary to ensure we can create at least 3,000 unforgeable signatures—the minimum required to securely send a BSM every 100 ms throughout each 5-minute period when we use one pseudonym certificate. While this instantiation gives higher security than the other schemes, it is the only instantiation that can securely generate this many signatures with the same key (a proposed Level 3 instantiation can only generate $2^{10} = 1024$ unforgeable signatures [57] and is therefore unacceptable).

Pure-PQ. We first discuss directly substituting ECDSA with PQC, which we call the *Pure-PQ* design. Figure 4 shows the sizes of public keys and signatures of each algorithm as well as the total frame size required to contain those elements if each algorithm were used to replace ECDSA. As the figure shows, directly replacing ECDSA with any of the PQ candidates would result in frame sizes that exceed the limit of 2,304 bytes. Using two frames per signature as a way to overcome this is inviable as it would incur unacceptable verification delay. Alternatively, only every other BSM could be signed using Falcon, leaving more than half of the BSMs unsigned–even more using one of the other PQ schemes. While allowed in the IEEE standard, this poses much opportunity for an attack and hence, we do not find this solution viable.

Moving on, a practical PQ-V2V design must be able to sign at least ten BSMs per second (10 Hz) and should be able to verify about 1,000 signatures per second (corresponding to an average of 100 nearby vehicles). The table shows that Falcon and Dilithium have acceptable performance while neither Sphincs$^+$ nor XMSS are even close to viable if every BSM carries a PQ signature. However, if these algorithms are only used to sign certificates (as in our *Partially Hybrid* design) rather than each individual BSM, then only 200 signature verifications are required per second. In this case, Sphincs$^+$ and XMSS are still slightly insufficient, but they are potentially viable.

Fully Hybrid. To keep classical security guarantees and add security against quantum adversaries at the same time, NIST recommends using dual signatures. In such an ideal design, certificates are a concatenation of an ECDSA and a PQ certificate, and BSMs are authenticated by ECDSA-PQ dual signatures over the same BSM. Under the current size constraints and hardware capability, however, this is not possible for every BSM, as discussed in *Pure-PQ* design above. Therefore we consider the following *Fully Hybrid* design. The first few SPDUs (with the exact number depending on the instantiation) contain BSMs signed using only ECDSA along with fragments of the sender's hybrid certificate. After a few messages, all fragments are received, allowing subsequent messages in the five-message cycle (if any) to be effectively protected with ECDSA-PQ dual signatures. Naturally, this additional security guarantee comes at the cost of having to send two certificates and two signatures, increasing the frame size to such an extent that *Fully Hybrid* PQ-V2V is only possible with significant overhead compared to our *Partially Hybrid* using Falcon. For all other instantiations, more than five BSMs need to be sent in order to transmit the entire certificate, exceeding the 5-message cycle. We provide a pseudo-code description, the details of the instantiation, resulting frame sizes, and discussion of inviability of the other PQ instantiations in Appendix B.

Table 2: PQ Security and viability of our *Partially Hybrid* compared to alternative designs.

Design	Certificate	BSM Auth.	Performance
Pure-ECDSA	○	○	Acceptable
Pure-PQ	◖	-/◖	Unacceptable
Fully Hybrid	●	○/●	Unacceptable
Partially Hybrid	●	○	Acceptable

- = none, ○ = ECDSA, ◖ = PQ, ● = ECDSA + PQ

Synopsis. Table 2 summarises the key properties of different designs detailed in this section: the *Pure-PQ* and *Fully Hybrid* designs offer PQ and ECDSA-PQ security, respectively, for both certificate and BSM authentication. However, the first BSMs are not or only ECDSA-authenticated due to the need of first communicating the large PQ/hybrid certificates. Unfortunately, both designs result in unacceptable performance. This motivates our *Partially Hybrid* design—also shown in the table for comparison.

4 Our Practical Partially Hybrid Design

During the PQ transition era A_{v} (see Figure 1), strict requirements (e.g., the upper bound of 2304 bytes on the payload size) are enforced to comply with the DSRC standard and facilitate sharing the limited bandwidth among many vehicles. Motivated by Section 3, we propose a *Partially Hybrid* design, proving it efficiently provides strong cryptographic protection against quantum adversaries during era A_{v}; specifically, it protects against adversaries who require at least time v to break ECDSA. We include parameter v in our design and assume that it can be gradually attenuated as attacks against ECDSA strengthen (e.g., $1 \leq \mathsf{v} \leq 30,000$ days—see Figure 1). This is based on current estimates of quantum computing [51,74] and

extrapolated from IBM's efforts [37]. Since we estimate such quantum computers will be built no earlier than ~ 2036, our design allows for a grace period to develop the hardware and V2V communication technology needed for a *Fully Hybrid* design.

Our core idea is to continue signing BSMs only with classical cryptography (i.e., ECDSA) while setting the validity period of the corresponding signature verification keys to v, significantly reducing it from the current 1 week. We then require the ECDSA verification key in the pseudonym certificate to be signed using ECDSA *and* PQ signatures. Put differently, this design protects the integrity of the ECDSA pseudonym verification key pk_S using dual ECDSA-PQ signatures, as the issuer's keys (and hence the signatures on the pseudonym certificates) are used over much longer time periods and need to be protected against quantum attacks. Our approach of carefully analyzing the quantum powers is inspired by the *quantum annoying* property of [23].

In this section, we define our threat model, explain how our fragmenting of certificates accommodates large PQ certificates, and then give an informal description of our viable (backwards-compatible) design and suggest different PQ instantiations.

4.1 Threat Model

The security goal defined by the IEEE 1609.2 standard is "to protect messages from attacks such as eavesdropping, spoofing, alteration, and replay" [39, Introduction]. Hence, we assume the attacker's goal is to make receivers accept fraudulent BSMs or certificates by launching the attacks mentioned above (excluding eavesdropping) in order to cause traffic delays or collisions, among other disruptive events.

Attacker's Capabilities. We assume the attacker can observe, drop, replay, or delay the sending or processing of legitimately generated and broadcast SPDUs; alter SPDUs, e.g., changing the BSM, changing/ dropping/ adding/ swapping the/a pseudonym certificate; enforce BSM contents that are then legitimately signed and broadcast by the targeted sender; and is unable to acquire more than one pseudonym certificate per pseudonym from CA (as is specified in [39]).

Assumptions. We assume that all computations (including storage of secret values) are secure, i.e., no side-channel or fault attacks can occur. Moreover, we assume that the certificate generation and verification is correct and secure. In particular, we assume that CAs are honest, the root certificate cannot be forged, certificates are only generated for legitimate pseudonyms, and invalid certificates are detectable. Furthermore, we assume that all honestly generated SPDUs are verified by the verifier in the *same order* that they have been sent by the signer (handled by lower layers or using signed BSM timestamps) as long as the receiver stays in the transmission range. Moreover, we assume that communication errors during transmission are handled by lower layers.

	Certificate fields	Value	Explanation
	version	4	Version of certificate format
	type	0	Implicit or explicit
	issuer	C	identifies issuer
toBeSigned	id	S	identifies holder (pseudonym)
toBeSigned	validityPeriod	start, duration v	validity period
toBeSigned	verifyKeyIndicator	R_S	reconstruction value
toBeSigned	PQsignatureAlg	PQ	PQ signature scheme
toBeSigned	others		E.g., crarId, crlSeries, region
	signature	$\mathsf{Sign}^{\mathsf{pq}}(\mathsf{sk}_S^{\mathsf{pq}}, \mathsf{C}_S)$	PQ signature generated by issuer

Table 3: Certificate fields of Partially PQ Hybrid certificate C_S; changes to classical certificate are shaded.

Quantum Powers. We assume that quantum computers cannot break ECDSA immediately. Instead, we assume for our *Partially Hybrid* design, that a quantum attacker needs $1 \leq \mathsf{v} \leq 24$ hours. This, based on recent findings [73], requires between 13×10^6 (for $\mathsf{v} = 24$ hours) and 317×10^6 (for $\mathsf{v} = 1$ hour) qubits. We refer to Section 1 for an estimated timeline on the arrival of such quantum computers. We discuss the resulting validity periods of the pseudonym certificates below.

4.2 Certificate Generation and Fragmentation

We formally define our certificate fragmentation scheme to be able to give a general description of our design, as follows.

Hybrid Certificate. Let C_S denote the hybrid certificate, of which there are several types, indicated by the `version` field. We propose a new `version` that combines an implicit ECDSA certificate with an explicit PQ-based certificate over the same ECDSA verification key (or reconstruction value, as the ECDSA certificate is implicit). This means that the issuer C holds two key pairs: ECDSA keys $(\mathsf{sk}_C^{\mathsf{c}}, \mathsf{pk}_C^{\mathsf{c}})$ and PQ keys $(\mathsf{sk}_C^{\mathsf{pq}}, \mathsf{pk}_C^{\mathsf{pq}})$. We depict our proposed certificate structure including the most important fields in Table 3.

Fragmentation. To satisfy size constraints on the frames, we will use *fragmentation*. More concretely, the first few SPDUs include fragments of the sender's hybrid certificate. After all fragments are received, the receiver can verify the integrity of the verification key using ECDSA and PQC, before using the ECDSA to verify any signatures.

Moreover, we define a function $\mathsf{CFrag}_\alpha : \mathsf{C} \to \{\mathsf{C}_1, ..., \mathsf{C}_\alpha\}$ which fragments a given certificate C into α parts. The number of fragments α is optimized based on the PQ algorithm used such that 1) α is minimal to transmit the entire hybrid certificate as soon as possible, 2) all frames are at most $2,304$ bytes, 3) the size of all frames used to transmit C_S is equal, to decrease the likelihood of frame loss due to large frames.

The inverse $\mathsf{CCons}_\alpha : \{\mathsf{C}_1, ..., \mathsf{C}_\alpha\} \to \mathsf{C}$ reconstructs a certificate from given fragments. We further define $\mathsf{H} : \{0,1\}^* \to \{0,1\}^{256}$ to be a hash function.

4.3 Informal Description

The pseudo-code description of our *Partially Hybrid* design in Figure 5 uses the following notation. We consider one run of the protocol, which re-occurs every five

messages, as explained in Section 2.1. Moreover, we define $\mathcal{S}_{\mathsf{c}} = (\mathsf{KGen}^{\mathsf{c}}, \mathsf{Sign}^{\mathsf{c}}, \mathsf{Vrfy}^{\mathsf{c}})$ to be ECDSA and $\mathcal{S}_{\mathsf{pq}} = (\mathsf{KGen}^{\mathsf{pq}}, \mathsf{Sign}^{\mathsf{pq}}, \mathsf{Vrfy}^{\mathsf{pq}})$ be a PQ scheme, see Appendix A for a definition of digital signature schemes and corresponding notation. In addition, we denote the function checking the validity of the pseudonym certificate using the CA's public key by $\mathsf{CVrfy}(\mathsf{pk}_C, \mathsf{C}_S) \in \{0,1\}$. The sender's keys are $(\mathsf{pk}_S^{\mathsf{c}}, \mathsf{sk}_S^{\mathsf{c}})$.

The proposed protocol can be divided into three stages, depending on the frame index $i \in [1,5]$, i.e., which of the five messages in the 5-message cycle is the current one. We denote the i-th message by BSM_i and its signature by sig_i, packed into spdu_i. The first stage (lines 3-11) and last stage (lines 25-31) are the same as for the ECDSA-based design conditioned on the Boolean value b_c. This value is 1 if the entire hybrid certificate has been received *and* verified by the receiver within the five minutes that the pseudonym certificate has been used, (i.e., b_c is set to 1 in line 18). If $b_c = 0$, processing of the BSM is delayed (using function $\mathsf{Delay}(\mathsf{BSM})$ in line 11) until the hybrid certificate has been verified. This is implicitly allowed under the IEEE 1609.2 standard and carried out similarly to the P2PCD mechanism in 1609.2-2016 [39], see Section 2.1. In our design, and as long as underlying communication protocols continue to impose current strict frame size limits, we can set a timer to 500 ms to obtain the hybrid certificate to verify the signature and process the BSM.

The first and second part (lines 3-24) are used to communicate all fragments of the hybrid certificate. Once this is done, the hybrid certificate is verified (line 16), which means that in the least the certificate validation period and the validity of the PQ signature is checked[4]. If the certificate is valid, the verification key $\mathsf{pk}_S^{\mathsf{c}}$ is reconstructed/extracted (line 17). It is important to note that if a signature verification fails (for the certificate chain or on the BSM itself), the BSM will be discarded. This is indicated by *Abort* in our pseudo-code, and follows the reasoning in [34].

Our design will cease to be cryptographically secure as soon as ECDSA can be broken within the validity period v (even though non-cryptographic methods can still be invoked to detect an abnormal BSM). Our design offers to adapt agile to a change of the state-of-the-art in quantum computers. Namely, the variable v can be decreased during certificate generation (therefore, more pseudonym certificates need to be generated as discussed below) and certificate verification. Both changes are in software only as the verifying vehicle can receive this update over-the-air. To keep the pseudo-code as simple as possible, we omit this detail in Figure 5.

Informal Security Analysis. As also supported by our formal analysis below, there are essentially two attack vectors: 1) A quantum adversary could *forge a signature on a BSM*. The attacker sees the respective key 'in use' for a total of only $2 \cdot 5 = 10$ to $14 \cdot 5 = 70$ minutes, depending on v. Moreover,

[4]For explicit certificates also the ECDSA signature is verified.

```
1 : Sender S                                   Receiver R
2 : h ← H( C_S )
3 : for i = 1,...,α−1 :
4 : sig_i ← Sign^c(sk_S^c, BSM_i)
5 : spdu_i ← (BSM_i, sig_i^c, h, C_i )
6 :                          spdu_i ⟶  if b_c = 1 :
7 :                                      if h = H( C_S )
8 :                                        if Vrfy^c(pk_S^c, sig_i, BSM_i) = 1 :
9 :                                          Process(BSM_i)
10 :                                       Abort
11 :                                     Delay(BSM_i)
12 : sig_α ← Sign^c(sk_S^c, BSM_α)
13 : spdu_α ← (BSM_α, sig_α, h, C_α )
14 :                         spdu_α ⟶  if b_c = 0 :
15 :                                     C_S ← CCons(C_1,...,C_α)
16 :                                     if CVrfy((pk_C^pq, pk_C^c), C_S ) = 1 :
17 :                                       pk_S^c ← C_S #reconstruct
18 :                                       b_c ← 1
19 :                                       Process(BSM_1,...,BSM_α)
20 :                                     Abort
21 :                                   if h = H( C_S ) :
22 :                                     if Vrfy^c(pk_S^c, sig_α, BSM_i) = 1 :
23 :                                       Process(BSM_α)
24 :                                   Abort
25 : for i = α+1,...,5 :
26 : sig_i ← Sign^c(BSM_i, sk_S^c)
27 : spdu_i ← (BSM_i, sig_i, h)
28 :                         spdu_i ⟶  if h = H( C_S )
29 :                                     if Vrfy^c(pk_S^c, sig_i, BSM_i) = 1 :
30 :                                       Process(BSM_i)
31 :                                   Abort
32 :
```

Figure 5: Pseudo-code description of the *Partially Hybrid* design to be repeated every five BSMs; $b_c \leftarrow 0$ at the beginning of a new 5-minute window; if $b_c = 1$, the receiver knows C_S; PQ values and operations are shaded

the attacker needs to generate the forgery within the respective validity period v. Since this would require quantum computers operating on a large number of qubits, quantum adversaries during era A_{v} (see Figure 1) are not more powerful than classical adversaries during such an attack. 2) A quantum adversary could attempt to *forge the pseudonym certificate* generated by a CA. To protect against this, we add a PQ signature generated over the ECDSA public key, resulting in the hybrid certificate C_S. Assuming that at least one–the PQ scheme or ECDSA– is unforgeable, this attack does not succeed.

It is worth noting that connected vehicles use several non-cryptographic safeguards to prevent forgeries of BSMs when certificates have not been verified yet. For example, threat detection services compare BSMs from a given vehicle against

its recent BSMs to detect anomalies, so a forgery attack would be limited in what it could claim without being identified as an anomaly (e.g., a BSM claiming a vehicle is stopped 100ms after its previous (legitimate) BSM reported moving at 60 km/hr would be clearly implausible). Further, on-board sensors like LiDAR or cameras act as a check on BSMs; e.g., a front-facing camera would reveal that no vehicle is actually braking ahead despite what a forged BSM might report.

Viability. Current practice is to rotate the use of 20 pseudonym certificates throughout one week, with each used at most 3,000 times per each five-minute rotation [53]. This means every certificate is used in at most 100 rotations (adding up to at most 300,000 signatures) for $\mathsf{v} = 1$ week. Since our *Partially Hybrid* design is only secure under the assumption that the pseudonym certificates are valid for at most one day, we advocate for increasing the number of certificates per week. The flexibility of our design and over-the-air updates allow decreasing the validity period v from one day to a few hours, likely even less. While decreasing the validity period toward the 5-minute (12-minute in ETSI standard [24]) absolute minimum would be even more conservative, using a certificate only once overall or keeping it valid for less than one hour might lead to practical complications [53], e.g., in defining the overlapping duration between rollovers (currently one hour) or related to managing a larger volume of certificates. If these limitations of current V2V protocols are resolved in the future (as ETSI is proposing to reduce the overlap to only a few minutes), v can be reduced further, allowing to extend the time our *Partially Hybrid* can be used securely.

Concretely, for $\mathsf{v} = 1$ day (resp., $\mathsf{v} = 2$ hours) the number of pseudonym certificates should be increased to at most 140 per week (resp., 2800), assuming[5] at most 20 different certificates used during v. While our approach requires a higher number of certificates per week, the butterfly key technique (used in 1609.2.1 [43]) allows a vehicle to obtain up to 2^{128} certificates per single request to a CA and efficient revocation when needed, which comfortably allows for shorter certificate validity time and makes our *Partially Hybrid* design viable.

Backwards-Compatibility. The *Partially Hybrid* design can be made backwards-compatible by allowing the receiver to ignore the PQ certificate. More concretely, this would change the certificate verification in line 16. Our notion of backwards-compatibility covers the scenario where vehicles' software and protocol fields can be updated (during maintenance or over-the-air) but the dedicated cryptographic unit (the hardware security module (HSM)) that is used (among others) to verify signatures, cannot. In order to prevent rollback attacks, we assume that all vehicles send and expect to receive the hybrid certificates, even if they do not possess the *hardware* capabilities to verify the PQ signature on the hybrid certificate. We note this enforcement only adds security for receivers who actually verify the PQ certificate. The advantage of this approach is that vehicles whose *software* but not their *hardware* has been updated will be able to continue to verify ECDSA signatures nevertheless. This approach also enables reverting back to ECDSA-only in a crypto-agile way in case the used PQC algorithms turn out to be insecure.

[5]This way, a certificate is only used in a total of at most 14 (resp., 2) rotations on average, potentially increasing the security and privacy since the certificates can be observed less often.

Table 4: Resulting sizes of frames F_i (in bytes) for the *Partially Hybrid* design; $|\mathsf{C}_S|$ is the size of the entire certificate.

PQ Scheme	$\lvert\mathsf{C}_S\rvert$	α	F_1	F_2	F_3	F_4	F_5	β
Pure ECDSA Design								
-	162	1	330	200	200	200	200	1
Partially Hybrid Design								
Falcon	858	1	1026	204	204	204	204	2
Dilithium	2588	2	1462	1462	204	204	204	4
Sphincs$^+$	8024	4	2174	2174	2174	2174	2048	
XMSS	2860	2	1598	1598	204	204	204	3

4.4 Instantiation

Table 4 presents the resulting frame sizes for each instantiation of the *Partially Hybrid* design for viable PQ schemes, calculated as follows. We compute the certificate size as $30 + |\mathsf{pk}| + |\mathsf{sig}_{\mathsf{pk}}|$, SPDU size as $|\text{SPDU}| = 24 + |\text{BSM}| + |\text{certificate}| + |\mathsf{sig}_{\text{BSM}}|$, and then the total MAC layer frame size as $|F| = 40 + |\text{SPDU}|$. Where $\mathsf{sig}_{\mathsf{pk}}$ and $\mathsf{sig}_{\text{BSM}}$ are the signatures over the sender's public key and over an BSM, respectively. For the existing ECDSA-based protocol using implicit certificates, the SPDU size is $|\text{SPDU}| = 24 + |\text{BSM}| + |\text{implicit cert}| + |\mathsf{sig}|$, where the implicit cert is comprised of a small reconstruction value $R_S = 38$ bytes. In addition, implementing each design requires a variable amount of overhead (on the order of 30-40 bytes).

The size of the ECDSA certificate $\mathsf{C}_S^{\mathsf{c}}$ is 162 bytes. Moreover, we compute the total frame size including the ECDSA-signed BSM and fragments of C_S with different instantiations of α (each frame also contains about 40 bytes of overhead for the encoded data structures). For our design to be viable, the number of frames needed to transmit C_S during the 5-message cycle as well as to share it via P2PCD learning response (see Section 2.1) is critical. Thus, Falcon, Dilithium, and XMSS are viable instantiations but Sphincs$^+$ is not.

5-Message Cycle. Falcon-, Dilithium-, XMSS-, and Sphincs$^+$-based certificates can be transmitted during the 5-message cycle as explained next. For Falcon, the size of $\mathsf{C}_S^{\mathsf{pq}}$ is 858 bytes as it includes a Falcon signature over an ECDSA key (see Table 4, column C_S). Therefore, it is not necessary to fragment C_S and one message is sufficient to communicate it (i.e., $\alpha = 1$). Hence, the payload size of the first frame F_1 is 1026 bytes and the payload size of the remaining frames F_2, F_3, F_4, F_5 are 204 bytes each. Dilithium, XMSS, and Sphincs$^+$ instantiations require larger values of α, as seen in Table 4, which translates to more messages being transmitted before the integrity of the ECDSA key can be

guaranteed by both classical and PQ signature schemes.

P2PCD. Currently, under ECDSA, a certificate can fit in a single P2PCD learning response; however, we have established that hybrid certificates must be fragmented as they exceed the DSRC payload size limit. Therefore, each P2PCD learning response in our design must also be fragmented. Moreover, before *each fragment* is transmitted the vehicle will wait for a period of time randomly selected (see Section 2.1). In Table 4, β indicates the number of frames required to completely convey a P2PCD learning response. The expected value of the uniformly distributed wait time is 125 ms, so *on average* we can expect Sphincs$^+$ will require 1000 ms to communicate all $\beta = 8$ fragments of a single learning response. For Dilithium, we would expect this to take about 500 ms, for XMSS, 375 ms, and 250 ms for Falcon. Based on this, we find Sphincs$^+$ to be unacceptable as it would almost always take longer than our 500 ms 5-message cycle to receive a learning response if any certificate has to be requested. Dilithium is on the edge of feasibility, but still viable, while XMSS and Falcon should generally be acceptable within the context of P2PCD requirements.

5 Related Work

Transitioning from classical cryptography to PQC by way of hybrid designs is specifically encouraged by NIST [50] and is well-established in the literature [7,18,21,28,29,31,46,67,70] for both hardware and network protocols. However, the majority of prior work on hybrid designs has considered very different systems than our V2V environment. For example, the multitude of work on integrating PQC into Transport Layer Security (TLS) (e.g., [18,21,29,31,46,67,70]) cannot be applied to V2V because such works do not consider the restricted payload size of DSRC, the latency requirements for safety-critical BSMs, or the inability of vehicles to negotiate algorithms or security parameters through unidirectional broadcast messages. Work on TLS and its kin (e.g. Security Protocol and Data Model (SPDM) [76]) further tends to assume a wired connection where the challenges of the wireless V2V environment (frame loss, fading, etc.) are not considered.

Prior works on PQC in embedded systems and wireless networks are more relevant to our work on V2V. Existing work on embedded systems (e.g., [18,31,70,72,76]) rarely considers the above constraints, and so it is of limited relevance. For wireless systems, PQC has mostly been considered only for key exchange or encryption (in contrast to our focus on authentication); e.g., [71] used PQC to protect the privacy of 5G subscriber identifiers, but only discussed key establishment protocols, while [20] investigated PQC in video streaming systems but looked only at encryption. In the vehicular communication domain, there is very limited work on integrating PQC. Most such work focuses on intra-vehicle device communication (e.g., [27,61]), a totally different problem than inter-vehicle V2V, and the few works that consider the 1609.2 standard at all focus on proposing alternatives (e.g., [61]) rather than a backwards-compatible transitional protocol like ours. To the best of our knowledge, we are the first to undertake the specific challenge of devising, implementing, and evaluating a PQC *authentication* scheme for V2V that can be easily integrated into the IEEE 1609.2 standard to kickstart the transition to a quantum-secure CV ecosystem.

Acknowledgments

We would like to thank Dan Brown and Matthew Campagna for clarifications on the ECDSA-ECQV security, Douglas Stebila and Fernando Virdia for fruitful discussions on the security of V2V protocols. Moreover, we are thankful to anonymous reviewers about suggestions how to extend earlier versions of this paper.

N.B. was supported by Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery grant RGPIN-2016-05146, NSERC Discovery Accelerator Supplement grant RGPIN-2016-05146, and Contract 2L 165-180499/001/sv, "PQC Analysis", funded by Public Works and Government Services Canada.

S.M was supported by Canada's NSERC Alliance Program and Public Works Government Services Canada (PWGSC).

This work was supported by the University of Waterloo Institute for Quantum Computing; IQC is supported in part by the Government of Canada through Innovation, Science and Economic Development Canada (ISED) and the Province of Ontario.

References

[1] *IEEE Standard for Wireless Access in Vehicular Environments (WAVE)–Certificate Management Interfaces for End Entities*, 2020.

[2] 3rd Generation Partnership Project (3GPP). *Summary of Rel-16 Work Items*, 2020.

[3] 3rd Generation Partnership Project (3GPP). *User Equipment (UE) radio transmission and reception*, 2021.

[4] 3rd Generation Partnership Project (3GPP). *Release 18*, 2022.

[5] Rémi Adelin, Cyrius Nugier, Éric Alata, Vincent Nicomette, Vincent Migliore, and Mohamed Kaâniche. Facing emerging challenges in connected vehicles: a formally proven, legislation compliant, and post-quantum ready security protocol. *Journal of Computer Virology and Hacking Techniques*, pages 1–28, 2022.

[6] Aarti Amod Agarkar, Mandar Karyakarte, and Himanshu Agrawal. Post quantum security solution for data aggregation in wireless sensor networks. In *Proc. IEEE*

Wireless Communications and Networking Conference (WCNC), pages 1–8, Seoul, Korea, 2020.

[7] Reza Azarderakhsh, Rami Elkhatib, Brian Koziel, and Brandon Langenberg. Hardware deployment of hybrid pqc: Sike+ ecdh. In *International Conference on Security and Privacy in Communication Systems*, pages 475–491. Springer, 2021.

[8] Hamidreza Bagheri, Md Noor-A-Rahim, Zilong Liu, Haeyoung Lee, Dirk Pesch, Klaus Moessner, and Pei Xiao. 5G NR-V2X: Toward connected and cooperative autonomous driving. *IEEE Communications Standards Magazine*, 5(1):48–54, 2021.

[9] William Barker, William Polk, and Murugiah Souppaya. Getting ready for post-quantum cryptography: Explore challenges associated with adoption and use of post-quantum cryptographic algorithms. *NIST CyberSecurity White Paper*, April 2021.

[10] Paulo S. L. M. Barreto, Jefferson E. Ricardini, Marcos A. Simplicio Jr., , and Harsh Kupwade Patil. qSCMS: Post-quantum certificate provisioning process for V2X. *Cryptology ePrint Archive*, February 2019.

[11] Ward Beullens. Breaking Rainbow takes a weekend on a laptop. In *Proc. 42nd Annual International Cryptology Conference (CRYPTO'22)*, pages 464–479, Santa Barbara, CA, August 2022. Springer.

[12] Karthikeyan Bhargavan and Gaëtan Leurent. On the practical (in-)security of 64-bit block ciphers: Collision attacks on HTTP over TLS and OpenVPN. In *Proc. ACM SIGSAC Conference on Computer and Communications Security*, pages 456–467, Vienna, Austria, October 2016.

[13] Nina Bindel, Udyani Herath, Matthew McKague, and Douglas Stebila. Transitioning to a quantum-resistant public key infrastructure. In Tanja Lange and Tsuyoshi Takagi, editors, *Post-Quantum Cryptography - 8th International Workshop, PQCrypto 2017, Utrecht, The Netherlands, June 26-28, 2017, Proceedings*, volume 10346 of *Lecture Notes in Computer Science*, pages 384–405. Springer, 2017.

[14] Alexandra Boldyreva, Marc Fischlin, Adriana Palacio, and Bogdan Warinschi. A closer look at PKI: Security and efficiency. In *Proc. International Workshop on Public Key Cryptography*, pages 458–475, 2007.

[15] Benedikt Brecht, Dean Therriault, Andre Weimerskirch, William Whyte, Virendra Kumar, Thorsten Hehn, and Roy Goudy. A security credential management system for V2X communications. *IEEE Trans. on Intelligent Transportation Systems*, 19(12):3850—3871, December 2018.

[16] Jon Brodkin. FCC takes spectrum from auto industry in plan to "supersize" Wi-Fi, November 2020. Accessed: July 15, 2021. Available: https://arstechnica.com/tech-policy/2020/11/fcc-adds-45mhz-to-wi-fi-promising-supersize-networks-on-5ghz-band/.

[17] Bureau of Transportation Statistics. Average age of automobiles and trucks in operation in the United States, 2022. Accessed: July 10, 2022. Available: https://www.bts.gov/content/average-age-automobiles-and-trucks-operation-united-states,.

[18] Kevin Bürstinghaus-Steinbach, Christoph Krauß, Ruben Niederhagen, and Michael Schneider. Post-quantum tls on embedded systems: Integrating and evaluating kyber and sphincs+ with mbed tls. In *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*, pages 841–852, 2020.

[19] Certicom Research. *SEC 4: Elliptic Curve Qu-Vanstone Implicit Certificate Scheme (ECQV)*, 2013.

[20] Alejandro Cohen, Rafael GL D'Oliveira, Salman Salamatian, and Muriel Médard. Network coding-based post-quantum cryptography. *IEEE Journal on Selected Areas in Information Theory*, 2(1):49–64, 2021.

[21] Eric Crockett, Christian Paquin, and Douglas Stebila. Prototyping post-quantum and hybrid key exchange and authentication in TLS and SSH. Cryptology ePrint Archive, Report 2019/858, 2019.

[22] Crypto4A. Crypto4A QASM™ achieves FIPS 140-2 level 3+ validation for the world's first PQC-capable hardware security module (HSM). https://crypto4a.com/press/crypto4a-qasm-achieves-fips-140-2-level-3-validation-for-the-worlds-first-pqc-capable-hardware-security-module-hsm/, June 2022. Accessed: July 26, 2022.

[23] Edward Eaton and Douglas Stebila. The "quantum annoying" property of password-authenticated key exchange protocols. In Jung Hee Cheon and Jean-Pierre Tillich, editors, *Post-Quantum Cryptography - 12th International Workshop, PQCrypto 2021, Daejeon, South Korea, July 20-22, 2021, Proceedings*, volume 12841 of *Lecture Notes in Computer Science*, pages 154–173. Springer, 2021.

[24] ETSI. *Intelligent Transport Systems (ITS); Security; Pre-standardization study on pseudonym change management*.

[25] Federal Communications Commission. In the matter of use of the 5.850-5.925 GHz band (ET Docket No. 19-138), November 2020. Accessed: February 10, 2021.

[26] Federal Communications Commission. The Federal Communications Commission: Seeks comment on a request for nationwide waiver of Intelligent Transportation System rules to use C-V2X technology in the 5.895-5.925 GHz band. Federal Register Vol. 87 No. 123, June 2022.

[27] Tim Fritzmann, Jonas Vith, and Johanna Sepulveda. Post-quantum key exchange mechanism for safety critical systems, 2019. Accessed: July 14, 2021. Available: https://hss-opus.ub.ruhr-uni-bochum.de/opus4/frontdoor/deliver/index/docId/6653/file/Kapitel2.pdf.

[28] Diana Ghinea, Fabian Kaczmarczyck, Jennifer Pullman, Julien Cretin, Rafael Misoczki, Stefan Kölbl, Luca Invernizzi, Elie Bursztein, and Jean-Michel Picod. Hybrid post-quantum signatures in hardware security keys. *Google Research*, 2022.

[29] Alexandre Augusto Giron, João Pedro Adami do Nascimento, Ricardo Custódio, and Lucas Pandolfo Perin. Post-quantum hybrid KEMTLS performance in simulated and real network environments. *Cryptology ePrint Archive*, 2022.

[30] Global UTM Association. Leveraging 3GPP cellular network mechanisms to support UAS operations, March 2022. Accessed: Feb. 7, 2023. Available: https://bit.ly/3OU1SrV.

[31] Ruben Gonzalez and Thom Wiggers. KEMTLS vs. post-quantum TLS: Performance on embedded systems. In *Proc. Int. Conf. on Security, Privacy, and Applied Cryptography Engineering (SPACE)*, pages 99–117, Jaipur, India, December 2022.

[32] Onn Haran and Ram Shallom. V2X technology trends and market evolution in Europe and China, August 2021. Accessed: July 29, 2022. Available: https://auto-talks.com/v2x-technology-trends-and-market-evolution-in-europe-and-china/.

[33] James Howe, Thomas Prest, and Daniel Apon. SoK: How (not) to design and implement post-quantum cryptography. In Kenneth G. Paterson, editor, *Topics in Cryptology - CT-RSA 2021 - Cryptographers' Track at the RSA Conference 2021, Virtual Event, May 17-20, 2021, Proceedings*, volume 12704 of *Lecture Notes in Computer Science*, pages 444–477. Springer, 2021.

[34] Shengtuo Hu, Qu Alfred Chen, Jiachen Sun, Yiheng Feng, Z Morley Mao, and Henry X Liu. Automated discovery of denial-of-service vulnerabilities in connected vehicle protocols. In *Proc. USENIX Security Symposium (USENIX Security)*, August 2021.

[35] A. Huelsing, D. Butin, S. Gazdag, J. Rijneveld, and A. Mohaisen. *XMSS: eXtended Merkle Signature Scheme*. Internet Engineering Task Force, May 2018.

[36] Andreas Hulsing, Daniel J. Bernstein, Christoph Dobraunig Maria Eichlseder, Scott Fluhrer, Stefan-Lukas Gazdag, Panos Kampanakis, Stefan Kolbl, Tanja Lange, Martin M, Lauridsen, Florian Mendel, Ruben Niederhagen, Christian Rechberger, Joost Rijneveld, Peter Schwabe, Jean-Philippe Aumasson, and Bas Westerbaan Ward Beullens. SPHINCS+, 2022. Accessed: July 28, 2022.

[37] IBM. IBM's roadmap for scaling quantum technology. https://research.ibm.com/blog/ibm-quantum-roadmap, 2022. Accessed: July 28, 2022.

[38] IEEE. *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments*, 2010.

[39] IEEE. *Wireless Access in Vehicular Environments–Security Services for Applications and Management Messages*, 2016.

[40] IEEE. *Wireless Access in Vehicular Environments–Security Services for Applications and Management Messages - Amendment 1*, 2017.

[41] IEEE. *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, 2020.

[42] IEEE. *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment: Enhancements for Next Generation V2X*, 2022.

[43] IEEE. *Wireless Access in Vehicular Environments (WAVE)–Certificate Management Interfaces for End Entities*, 2022.

[44] Internet Engineering Task Force. RFC 8554: Leighton-Micali hash-based signatures, 2019. Accessed: December 02, 2021.

[45] Panos Kampanakis and Scott R. Fluhrer. LMS vs XMSS: A comparison of the stateful hash-based signature proposed standards. *IACR Cryptol. ePrint Arch.*, page 349, 2017.

[46] Panos Kampanakis and Dimitrios Sikeridis. Two post-quantum signature use-cases: Non-issues, challenges and potential solutions. In *Proceedings of the 7th ETSI/IQC Quantum Safe Cryptography Workshop, Seattle, US*, November 2019.

[47] Todd Litman. *Autonomous vehicle implementation predictions*. Victoria Transport Policy Institute, Victoria, Canada, January 2023.

[48] Albert Chun-Chen Liu, Oscar Ming Kin Law, and Iain Law. *Understanding Artificial Intelligence: Fundamentals and Applications*, chapter 5, pages 39–52. Springer, 2022.

[49] Vadim Lyubashevsky, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Peter Schwabe, Gregor Seiler, Damien Stehlé, and Shi Bai. CRYSTALS-DILITHIUM. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

[50] Dustin Moody. Nist pqc: Looking into the future. NIST-Fourth PQC Standardization Conference, 2022.

[51] Michele Mosca and Marco Piani. Quantum Threat Timeline Report 2022. Global Risk Institute in Financial Services (GRI), https://globalriskinstitute.org/publications/2022-quantum-threat-timeline-report/, 2022. Accessed: December 27, 2022.

[52] National Highway Traffic Safety Administration. Fact sheet: Improving safety and mobility through vehicle-to-vehicle communication technology, 2014. Accessed: Feb. 20, 2020.

[53] NHTSA. Federal Motor Vehicle Safety Standards; V2V Communications: 49 CFR 571. https://www.federalregister.gov/d/2016-31059/p-870, April 2017.

[54] NHTSA. Report to Congress: Vehicle safety recall completion rates report, December 2018. Accessed: November 9, 2021.

[55] NIST. *Advanced Encryption Standard*, 2001.

[56] NIST. *Transitioning the Use of Cryptographic Algorithms and Key Lengths*. Washington, D.C., 2019.

[57] NIST. *Recommendation for Stateful Hash-Based Signature Schemes*. Washington, D.C., 2020.

[58] NIST. Status report on the third round of the NIST post-quantum cryptography standardization process. https://nvlpubs.nist.gov/nistpubs/ir/2022/NIST.IR.8413.pdf, 2022. Accessed: July 26, 2022.

[59] PQShield. PQSoC hardware for embedded devices. https://pqshield.com/solutions/pqsoc/, 2022. Accessed: July 26, 2022.

[60] Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. FALCON. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

[61] Prasanna Ravi, Vijaya Kumar Sundar, Anupam Chattopadhyay, Shivam Bhasin, and Arvind Easwaran. Authentication protocol for secure automotive systems: Benchmarking post-quantum cryptography. In *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, Seville, Spain, 2020.

[62] Wilbur L. Ross and Walter Copan. Digital signature standard (DSS), 2013. Accessed: December 02, 2021.

[63] SAE International. *Dedicated Short Range Communication (DSRC) Systems Engineering Process Guidance for SAE J2945/X Documents and Common Design Concepts*, December 2017.

[64] SAE International. *V2X Communications Message Set Dictionary*, 2020.

[65] Khabaz Sehla, Thi Mai Trang Nguyen, Guy Pujolle, and Pedro Braconnot Velloso. Resource allocation modes in C-V2X: From LTE-V2X to 5G-V2X. *IEEE Internet of Things Journal*, 9(11):8291–8314, June 2022.

[66] Murray Slovick. Toyota, Lexus commit to DSRC V2X starting in 2021, May 2018. Accessed: July 29, 2022.

[67] Douglas Stebila, Scott Fluhrer, and Shay Gueron. Hybrid key exchange in TLS 1.3. Internet-Draft draft-ietf-tls-hybrid-design-05, Internet Engineering Task Force, August 2022. Work in Progress.

[68] Steven Loveday. Mercedes-Benz to start selling level 3 drive pilot in Germany, May 2022. Accessed: July 13, 2022.

[69] Chengdong Tao, Albrecht Petzoldt, and Jintai Ding. Efficient key recovery for all HFE signature variants. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part I*, volume 12825 of *Lecture Notes in Computer Science*, pages 70–93. Springer, 2021.

[70] George Tasopoulos, Jinhui Li, Apostolos P Fournaris, Raymond K Zhao, Amin Sakzad, and Ron Steinfeld. Performance evaluation of post-quantum tls 1.3 on resource-constrained embedded systems. In *Information Security Practice and Experience: 17th International Conference, ISPEC 2022, Taipei, Taiwan, November 23–25, 2022, Proceedings*, pages 432–451. Springer, 2022.

[71] Vincent Quentin Ulitzsch, Shinjo Park, Soundes Marzougui, and Jean-Pierre Seifert. A post-quantum secure subscription concealed identifier for 6g. In *Proceedings of the 15th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pages 157–168, 2022.

[72] Wen Wang, Bernhard Jungk, Julian Wälde, Shuwen Deng, Naina Gupta, Jakub Szefer, and Ruben Niederhagen. Xmss and embedded systems. In *International Conference on Selected Areas in Cryptography*, pages 523–550. Springer, 2019.

[73] Mark Webber, Vincent Elfving, Sebastian Weidt, and Winfried K. Hensinger. The impact of hardware specifications on reaching quantum advantage in the fault tolerant regime. *AVS Quantum Science*, 4(1):013801, January 2022.

[74] Mark Webber, Vincent Elfving, Sebastian Weidt, and Winfried K Hensinger. The impact of hardware specifications on reaching quantum advantage in the fault tolerant regime. *AVS Quantum Science*, 4(1):013801, 2022.

[75] Yahoo! Finance. Vehicle-to-vehicle (V2V) communication global market report 2022, 2022. Accessed: May 2, 2022.

[76] Jiewen Yao, Krystian Matusiewicz, and Vincent Zimmer. Post quantum design in spdm for device authentication and key establishment. *Cryptography*, 6(4):48, 2022.

[77] Greg Zaverucha, Melissa Chase, David Derler, Steven Goldfeder, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, Jonathan Katz, Xiao Wang, Vladmir Kolesnikov, and Daniel Kales. Picnic. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

A Definitions of Cryptographic Primitives and Their Security

Digital Signature Schemes We first present formal definitions of a digital signature scheme and security concepts as utilised in our designs and security analyses.

Definition 1 (Digital Signature Scheme)**.** *A Digital Signature Scheme is defined as a tuple of algorithms* $\mathcal{S} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$*, which are defined as follows:*

KGen *returns a public key* pk *and secret key* sk.
Sign *returns a signature* sig *on a message* m *using* sk.
Vrfy *returns 0 or 1. Upon input of a message* m*, a signature* sig*, and the public key* pk*, this returns 1 if the signature is valid. Otherwise, 0 is returned.*

$\mathcal{S}$ is considered correct if

$$Pr[\mathsf{Vrfy}(\mathsf{pk}, \mathsf{Sign}(\mathsf{sk}, \mathsf{m}), \mathsf{m}) \mid (\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}()] = 1.$$

Definition 2 (eUF)**.** *Let* $\mathcal{S} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ *be a digital signature scheme. Let* $\mathcal{A}$ *be an efficient (classical/quantum) adversary. We define the advantage of* $\mathcal{A}$ *against the security experiment* $\mathrm{Expt}^{\mathsf{eUF}}_{\mathcal{S}}(\mathcal{A})$ *as*

$$Adv^{\mathsf{eUF}}_{\mathcal{S}}(\mathcal{A}) = \Pr\left[\mathrm{Expt}^{\mathsf{eUF}}_{\mathcal{S}}(\mathcal{A}) = 1\right].$$

We say that $\mathcal{S}$ *is secure against Existential Unforgeability under Chosen Message Attack (*eUF*) if* $Adv^{\mathsf{eUF}}_{\mathcal{S}}(\mathcal{A})$ *is negligible in the security parameter* λ*.* $\mathrm{Expt}^{\mathsf{eUF}}_{\mathcal{S}}$ *is defined as follows:*

At the beginning of the experiment, define $q_{\mathsf{sig}} \leftarrow 0$ *and* $Q_{\mathsf{sig}} \leftarrow \{\}$*. The challenger calls* $\mathsf{KGen}()$ *to return a public key* pk *and secret key* sk*, and passes* pk *to* $\mathcal{A}$*.* $\mathcal{A}$ *may query the signing oracle* O_{Sign} *on messages in the message space* $\mathcal{M}_{\mathcal{S}}$ *as shown in Figure* **??***. Eventually* $\mathcal{A}$ *outputs a message-signature pair* $(\mathsf{m}^*, \mathsf{sig}^*)$*. The experiment returns 1 if* $\mathsf{Vrfy}(\mathsf{pk}, \mathsf{sig}, \mathsf{m}) = 1$ *and such that* $(\mathsf{m}^*, \cdot) \notin Q_{\mathsf{sig}}$.

Pre-image Resistance We now define pre-image resistance for hash functions.

Definition 3 ($\mathsf{2PR}$)**.** *Let* $\mathsf{H} : \{0,1\}^* \rightarrow \{0,1\}^{256}$ *be a hash function. Let* $\mathcal{A}$ *be an efficient (classical/quantum) adversary. We define the advantage of* $\mathcal{A}$ *against the security experiment* $\mathrm{Expt}^{\mathsf{2PR}}_{\mathsf{H}}(\mathcal{A})$ *as*

$$Adv^{\mathsf{2PR}}_{\mathsf{H}}(\mathcal{A}) = \Pr\left[\mathrm{Expt}^{\mathsf{2PR}}_{\mathsf{H}(x)}(\mathcal{A}) = 1 | x \in \{0,1\}^*\right].$$

We say that H *is secure against second Pre-image Resistance attacks (*$\mathsf{2PR}$*) if* $Adv^{\mathsf{2PR}}_{\mathcal{S}}(\mathcal{A})$ *is negligible in the security parameter* λ*.* $\mathrm{Expt}^{\mathsf{2PR}}_{\mathsf{H}}$ *is defined as follows:*

At the beginning of the experiment, define $q_{\mathsf{H}} \leftarrow 0$ *and* $Q_{\mathsf{H}} \leftarrow \{\}$*. The challenger passes* $x' \in \{0,1\}^*$ *to* $\mathcal{A}$*.* $\mathcal{A}$ *may query the hashing oracle* O_{H} *on elements* $x \in \{0,1\}^*$ *as shown in Figure* **??***. Eventually* $\mathcal{A}$ *outputs an element* $x^* \in \{0,1\}^*$*. The experiment returns 1 if* $\mathsf{H}(x^*) = \mathsf{H}(x')$ *and* $x^* \neq x'$.

Certified Signature Scheme Next we give the definition of certified signature schemes [14]. As we aim at giving a generalized definition for explicit, implicit and hybrid certificates, our algorithms are defined very generically. Following [14], we assume that the pair (U, pk_S) is uniquely bound in the certificate C_S.

Definition 4 (Certified Signature Scheme)**.** *A certified signature scheme* $\mathcal{C} = (\mathsf{KGen}_C, \mathsf{CGen}(\mathsf{CGen}_C, \mathsf{CGen}_S), \mathsf{Sign}, \mathsf{Vrfy})$ *is defined via the following polynomial-time algorithms.*

KGen_C *returns a public key* pk_C *and secret key* sk_C *belonging to the certificate authority* C.

$\mathsf{CGen}(\mathsf{CGen}_C, \mathsf{CGen}_S)$ *is an interactive (two-party) public-key registration protocol, involving the sender S and the CA C running their (randomized) sub-protocols* CGen_C *and* CGen_S*, respectively.* CGen_C *takes input a secret key* sk_C*;* CGen_S *takes input the identity S of a sender and the public key* pk_C *corresponding to* sk_C*. As result of the interaction, the output of* CGen_C *is* $(S, \mathsf{pv}_S, \mathsf{C}_S)$*, where* pv_S *is a public key value* pv_S*, corresponding to a public key* pk_S*, and* C_S *is an issued certificate. If* C_S *is an explicit certificate,* $\mathsf{pv}_S = \mathsf{pk}_S$*; if it is implicit,* pv_S *is the reconstruction value; if it is a hybrid certificate combining two or more sub-certificates,* pv_S *is the concatenation of the corresponding public key values. The local output of* CGen_S *is* $(S, \mathsf{pk}_S, \mathsf{sk}_S, \mathsf{C}_S)$*, where* sk_S *is a secret key S will use to sign messages. The owner of* sk_C *should not learn* sk_S *during* CGen*. Either party can quit the execution prematurely, in which case the output of the party is set to* $\perp$*.*

Sign *is a (possibly) randomized signing algorithm. It takes input an identity S, a secret key* sk_S*, a certificate cert* C_S*, the CA's public key* pk_C *and a message* m*, and outputs a signature* sig*.*

Vrfy *is a deterministic verification algorithm. It takes input an identity S, a public key* pk_S*, a certificate* C_S*, a public key* pk_C*, a message* m*, and a signature* sig*, and outputs 0 or 1. In the latter case, we say that* sig *is a valid signature for* m *relative to* $(S, \mathsf{pk}_S, \mathsf{C}_S, \mathsf{pk}_C)$*. If* C_S *is an implicit certificate this also involves the reconstruction of the U's public key.*

[14, Definition 4.1] defines unforgeability of a certified signature schemes, which we will base our *i*-unforgeability definition on.

B *Fully Hybrid* Design

We provide its pseudo-code description in Figure 6 and discuss instantiations.

Let $\mathcal{P}$ be the *Fully hybrid* protocol using the two signature schemes $\mathcal{S}_{\mathsf{c}}$ and $\mathcal{S}_{\mathsf{pq}}$.

KGen_C returns $(\mathsf{pk}_C, \mathsf{sk}_C)$ as in the *Partially Hybrid* design.

CGen generates $\mathsf{C}_S = (\mathsf{C}_S^{\mathsf{c}} || \mathsf{C}_S^{\mathsf{pq}})$ with $\mathsf{C}_S^{\mathsf{c}}$ over $(S, \mathsf{pk}_S^{\mathsf{c}})$ and $\mathsf{C}_S^{\mathsf{pq}}$ over $(S, \mathsf{pk}_S^{\mathsf{pq}})$, with $(\mathsf{sk}_S^{\mathsf{c}}, \mathsf{pk}_S^{\mathsf{c}}) \leftarrow \mathsf{KGen}^{\mathsf{c}}()$, $(\mathsf{sk}_S^{\mathsf{pq}}, \mathsf{pk}_S^{\mathsf{pq}}) \leftarrow \mathsf{KGen}^{\mathsf{pq}}()$.

SPDUGen returns (with $\mathsf{sig}_i^{\mathsf{pq}} \leftarrow \mathsf{Sign}^{\mathsf{pq}}(\mathsf{sk}_S^{\mathsf{pq}}, \mathsf{BSM}_i)$, $\mathsf{sig}_i^{\mathsf{c}} \leftarrow \mathsf{Sign}^{\mathsf{c}}(\mathsf{sk}_S^{\mathsf{c}}, \mathsf{BSM}_i)$, and $\mathsf{sig}_i \leftarrow (\mathsf{sig}_i^{\mathsf{pq}} || \mathsf{sig}_i^{\mathsf{c}})$)

$$\mathsf{spdu}_i = \begin{cases} (\mathsf{BSM}_i, \mathsf{sig}_i^{\mathsf{c}}, \mathsf{C}_i), & \text{for } i = 1, \\ (\mathsf{BSM}_i, \mathsf{sig}_i^{\mathsf{c}}, h^{\mathsf{c}}, \mathsf{C}_i), & \text{for } i \in [2, \alpha - 1], \\ (\mathsf{BSM}_i, \mathsf{sig}_i^{\mathsf{c}}, h, \mathsf{C}_i), & \text{for } i = \alpha, \\ (\mathsf{BSM}_i, \mathsf{sig}_i, h), & \text{for } i \in [\alpha + 1, 5]. \end{cases}$$

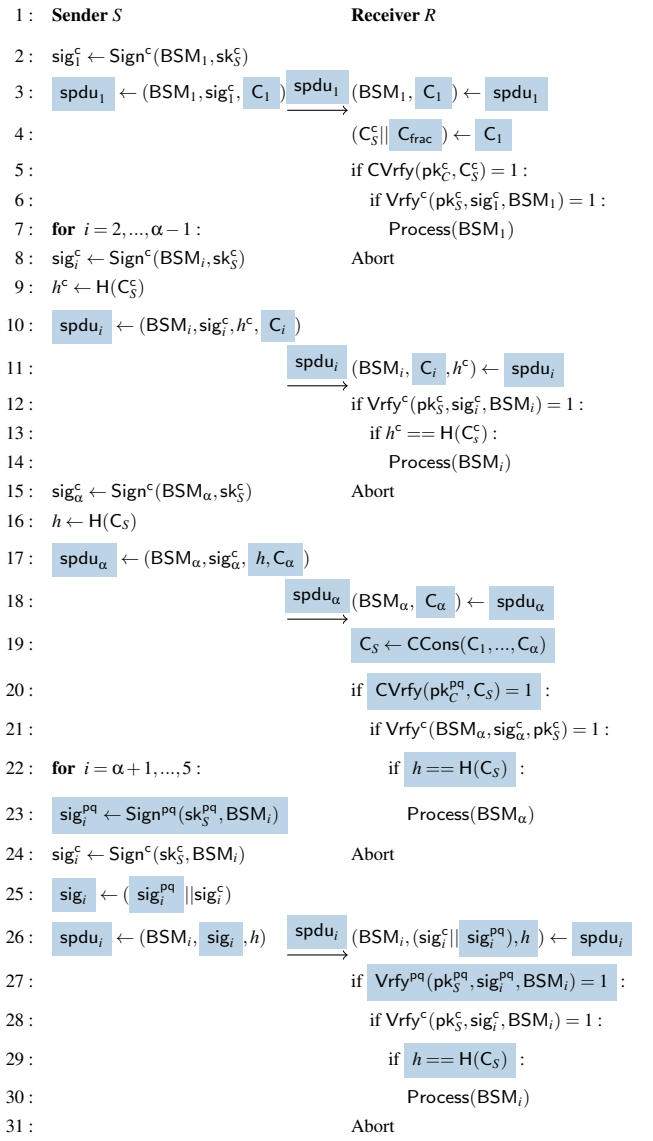

Figure 6: Pseudo-code description of the *Fully Hybrid* design to be repeated every five BSMs.

SPDUVerify is defined as follows, with $(\mathsf{sig}_i^{\mathsf{c}} || \mathsf{sig}_i^{\mathsf{pq}}) \leftarrow \mathsf{sig}_i$. For $i \in [1, \alpha - 1]$: as in the *Partially Hybrid* design. For $i = \alpha$: $\mathsf{C}_S \leftarrow \mathsf{CCons}(\mathsf{C}_1, ..., \mathsf{C}_\alpha)$. If $\mathsf{CVrfy}(\mathsf{pk}_C, \mathsf{C}_S, \mathsf{st}) = 1 \wedge h = \mathsf{H}(\mathsf{C}_S) \wedge \mathsf{Vrfy}^{\mathsf{c}}(\mathsf{pk}_S^{\mathsf{c}}, \mathsf{sig}_i^{\mathsf{c}}, \mathsf{BSM}_i) = 1 \wedge \mathsf{spdu}_i$ of correct form, then process BSM_i, update st with $\mathsf{H}(\mathsf{C}_S)$ and pk_S, and return 1. For $i \in [\alpha + 1, 5]$: if $h = \mathsf{H}(\mathsf{C}_S \wedge \mathsf{Vrfy}^{\mathsf{pq}}(\mathsf{pk}_S^{\mathsf{pq}}, \mathsf{sig}_i^{\mathsf{pq}}, \mathsf{BSM}_i) = 1 \wedge \mathsf{Vrfy}^{\mathsf{c}}(\mathsf{pk}_S^{\mathsf{c}}, \mathsf{sig}_i^{\mathsf{c}}, \mathsf{BSM}_i) = 1 \wedge \mathsf{spdu}_i$ of correct form, then process BSM_i and return 1. Else, return 0.

Discussion on PQ Security As we first need to transmit the entire hybrid certificate before we can sign and verify the hybrid PQ-ECDSA signatures, the first α message(s) of each five-message cycle are only protected using ECDSA. Ideally all messages should be authenticated by ECDSA *and*

Table 5: Resulting sizes of frames F_i (in bytes) for our *Fully Hybrid* design; $|C_S|$ is the size of the entire certificate (DSRC)

PQ Scheme	$\|C_S\|$	α	F_1	F_2	F_3	F_4	F_5	β
Pure ECDSA Design								
-	162	1	330	200	200	200	200	1
Fully/Backwards Compatible Hybrid Design								
Falcon	1723	1	1894	894	894	894	894	2

PQ signatures. However, embedding both the PQ certificate and signature in the first frame would incur a large frame size. Losing an important BSM due to its large frame size poses a more severe risk than a quantum adversary who would need to successfully run a very precise attack on the first SPDU in the five-message cycle.

Backwards-Compatibility As with the *Partially Hybrid* design, the *Fully hybrid* design can be extended to be backwards-compatible. The difference between the design described in Figure 6 and its backwards-compatible variant lies in whether the receiver runs verification on the PQ signature or not. More concretely in the handling of the $[\alpha, 5]$-th SPDUs in SPDUVerify. As before (see Section 4.3), we assume that all vehicles send and expect to receive the hybrid certificates, even if they do not possess the *hardware* capabilities to verify the PQ signature, in order to prevent rollback attacks. We also require each sender's certificate to indicate whether the sender has PQ-signing capabilities, so the verifier knows whether to run PQ-verification, and the fact this is signed by the CA prevents an adversary mounting a rollback attack. We note this is not included in our implementation, as we assume PQ capabilities for all vehicles.

Following [13], security by both schemes can only be guaranteed for honest sender/receivers and if both, Falcon and ECDSA, signatures are verified. Generating Falcon signatures does not give extra security for receivers without hardware updates compared to pure ECDSA V2V.

Instantiation and Resulting Frame Sizes. Only signature algorithms whose associated certificates can be sent in five or fewer fragments α can be used in the *Fully Hybrid* design, due to the five-message cyclic nature of the protocol. As mentioned in Section B, the only viable instantiation on current hardware and under the current size constraints is Falcon, with resulting frame sizes reported in Table 5. We chose to instantiate $\alpha = 1$. To be more concrete, using explicit ECDSA certificates the frame F_1 is 1894 bytes, as it contains the BSM, the ECDSA signature, and the entire certificate C_S (see Table 5, column "F_1"). As explained in Section 4.4, β indicates how many frames are necessary during P2PCD. Although the Dilithium and XMSS certificates can be split similarly to Falcon, the size of their signatures alone (let alone certificates) exceed the 2,304-byte payload limit; therefore, we do not instantiate our design using them.

C Analysis of Picnic

Picnic had been an alternate candidate in the 2nd round of NIST's PQC standardization but was not ultimately selected. As this decision was not made due to a security issue in Picnic (as was the case for Rainbow and GeMSS), we present a short analysis of Picnic here. As in Section 3, we chose a Picnic instantiation yielding a minimum security level of NIST Level 1 from the literature, namely *Picnic-L1-FS* [77]. Figure 7 shows that *Picnic* would require frame sizes that are completely unsustainable in the V2V environment. When considering Picnic for our *Partially Hybrid* design, we deduced that even fragmentation into five parts is not small enough to communicate the certificate within five messages. Hence, Picnic cannot be used in our *Partially Hybrid* design. Likewise, we also needed to rule out Picnic for its large signature sizes for our *Fully Hybrid* and *Pure PQ* design.

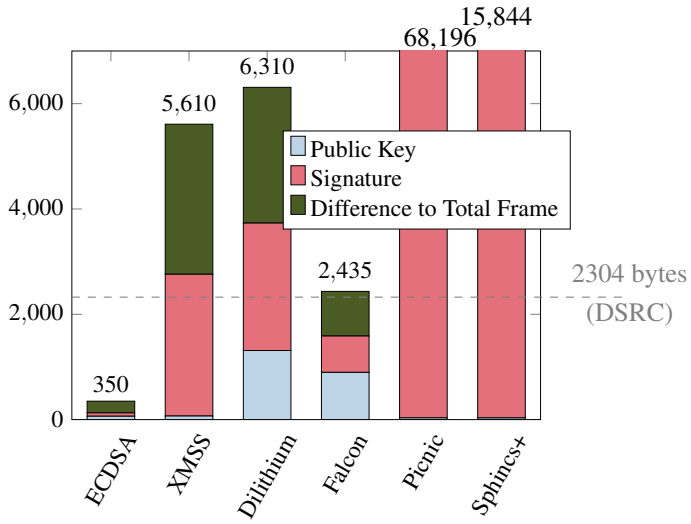

Figure 7: PQC public key and signature sizes versus frame size constraints of DSRC (shown by dashed line).