Post-Quantum Cryptography (PQC): Pioneering the Future of Automotive Security

Post-Quantum Cryptography (PQC): Pioneering the Future of Automotive Security

Santosh Kumar FIP, CISSP, PMP, CISA, CHFI, AIGP
Cybersecurity & Data Protection Leader | CISO & DPO | GenAI Architect | Fellow of Information Privacy (FIP) 🏫 IIT…

July 25, 2024

Introduction

The automotive industry is at the brink of a revolutionary transformation, propelled by the integration of advanced software and enhanced connectivity features. These advancements offer unparalleled flexibility, continuous updates, and enriched user experiences. However, the increased digitalization and connectivity of vehicles also open new avenues for cyber threats. Enter Post-Quantum Cryptography (PQC)—a cutting-edge approach designed to safeguard communications and data against the looming threat posed by quantum computers, which have the potential to break traditional cryptographic algorithms.

The Security Challenge in Modern Vehicles

As vehicles become more software-centric and connected, they increasingly become targets for cyber-attacks. Cybercriminals could exploit vulnerabilities to seize control of critical vehicle functions, leading to catastrophic consequences. While current cryptographic measures are effective, they may falter against the formidable capabilities of quantum computers, which can solve complex mathematical problems at unprecedented speeds.

Understanding Post-Quantum Cryptography (PQC)

What is PQC?

Since quantum computers represent a ***distinct and potentially more powerful paradigm for computing*** than the classical computers in use today, cryptographic security needs to be reassessed for a world where quantum computers may proliferate, enabling new cryptographic attacks that would not be possible using classical computers.

These attacks may occur in future on data that is being transmitted or stored now known as ***harvest now***, ***decrypt later***, so it is not sufficient to wait until the required systems are available, but to make changes now.

The field of ***quantum-safe cryptography (QSC)*** encompasses efforts to ***identify and develop cryptographic schemes*** that can withstand

attacks both from quantum and classical computing systems. This is also sometimes called ***quantum-resistant, or post-quantum cryptography.***

The development of general-purpose quantum computers poses risks to a number of traditional cryptographic primitives such as symmetric key algorithms, cryptographic hash functions, and asymmetric key algorithms.

The most significant impacts of quantum algorithms occur in the context of asymmetric key cryptography, where ***Shor's algorithm*** offers a polynomial-time solution to the ***prime factoring*** and ***discrete logarithm problems***. Therefore, asymmetric cryptosystems based on factoring and discrete logarithms need to be replaced by new quantum-safe cryptography schemes.

This is in contrast to the symmetric key and cryptographic hashing protocols impacted by the Grover and **BHT** algorithms, where the quantum speedups are not super-polynomial. Therefore, in this latter case, existing algorithms such as AES and SHA-256 can be fortified at least in the medium term by ensuring sufficiently long key and hash lengths.

Current Cryptographic Landscape

Symmetric Cryptography: Breaking AES-256: Classical vs. Quantum Perspectives

Classical Perspective: AES-256, the Advanced Encryption Standard utilizing a 256-bit key, is celebrated for its robustness in safeguarding sensitive data. Theoretically, AES-256 provides 2^{256} possible keys, rendering brute-force attacks virtually impractical using classical computational resources. For a brute-force attack scenario where a classical computer tests approximately 1 million keys per second, the estimated time to exhaustively search through all possible keys would amount to approximately 3.67× 10^{63} years. This protracted duration underscores the security AES-256 affords against contemporary computational power, making it a cornerstone of modern cryptographic protection.

Quantum Perspective: Quantum computing introduces the potential to leverage Grover's algorithm, which offers a quadratic speedup for searching through unsorted databases. Applied to AES-256, Grover's algorithm reduces the effective search space from 2^{256} to 2^{128} operations. Despite this substantial reduction, even under optimistic conditions where a quantum computer performs one trillion operations per second, the estimated time required to break AES-256 would still span approximately 1.08×10^{19} years. This quantum advantage, while significant, suggests that AES-256 r**emains secure against quantum threats in the foreseeable future**, providing a considerable buffer against potential quantum attacks.

Resources:

- **NIST on AES** - Provides the official AES specification and in-depth background information.
- Grover's Algorithm Explained - Offers insights into how Grover's algorithm impacts symmetric encryption.

Asymmetric Cryptography: Breaking RSA-2048: Classical Vulnerabilities and Quantum Resilience

Classical Vulnerabilities: RSA-2048, based on the difficulty of factoring large composite numbers, remains a key cryptographic mechanism. The most efficient classical factoring algorithm, the General Number Field Sieve (GNFS), would necessitate approximately 8.2×10^{22} years to factorize a 2048-bit RSA key. This time complexity highlights RSA-2048's

resilience against classical attacks, assuming no substantial breakthroughs in factoring algorithms.

Quantum Advantage: The advent of Shor's algorithm, a quantum algorithm designed for efficient integer factorization, presents a formidable challenge to RSA-2048. Shor's algorithm operates in polynomial time, significantly reducing the problem complexity. On a sufficiently advanced quantum computer, **RSA-2048 could potentially be compromised within a few hours**. This capability underscores the need for a transition to quantum-resistant cryptographic algorithms.

Resources:

- **RSA Algorithm** - A comprehensive overview of RSA encryption and its underlying mathematics.
- Shor's Algorithm - Detailed explanation of Shor's algorithm and its implications for cryptographic security.

Hash Function: Breaking SHA-256: Hash Function Vulnerabilities

Classical Security: SHA-256, a cryptographic hash function producing a 256-bit hash value, offers 2^{256} possible outputs. This vast space makes brute-force attacks against SHA-256 infeasible with current classical computing capabilities, ensuring robust protection for hashed data.

Quantum Considerations: Grover's algorithm also applies to hash functions, such as SHA-256, providing a quadratic speedup. This reduces the time complexity for brute-forcing SHA-256 from 2^{256} to 2^{128} operations. Despite this speedup, breaking SHA-256 with quantum computing would still require approximately 1.08×10^{25} years, reflecting the hash function's enduring security.

Resources:

- **SHA-256 Overview** - Detailed description of SHA-256's properties and applications.
- **Grover's Algorithm for Hash Functions** - Research paper discussing Grover's algorithm's application to hash function attacks.

Quantum algorithms and impacts to cryptography

Quantum Algorithm	Functionality	Security Strength (n = number of bits)	Impacted Cryptographic Protocols	Mitigation
Shor	factoring	poly(n)	RSA	Migrate to QSC
Shor	discrete logarithm	poly(n)	Diffie-Hellman, DSA, Elliptic Curve Cryptography	Migrate to QSC
Grover	key search	$2^{n/2}$	Symmetric key algorithms (e.g., AES)	Sufficient key length
Grover	pre-image attack	$2^{n/2}$	Hash functions (e.g., SHA-256)	Sufficient hash length
BHT	collision attack	$2^{n/3}$ or $2^{2n/5}$	Hash functions (e.g., SHA-256)	Sufficient hash length

Security Strength

Security Strength : As defined by the **NIST**: *A number characterizing the amount of work that is expected to suffice to "defeat" an implemented cryptographic mechanism (e.g., by compromising its functionality and/or circumventing the protection that its use was inSecurity StrengthAs defined by the NIST: A number characterizing the amount of work that is expected to suffice to "defeat" an implemented cryptographic mechanism (e.g., by compromising its functionality and/or circumventing the protection that its use was intended to facilitate). Security strength is often expressed in bits. If the security strength of a particular implementation of a cryptographic mechanism is s bits, it is expected that the equivalent of (roughly) 2s basic operations of some sort will be sufficient to defeat it in some way. tended to facilitate). Security strength is often expressed in bits. If the security strength of a particular implementation of a cryptographic mechanism is s*

bits, it is expected that the equivalent of (roughly) 2s basic operations of some sort will be sufficient to defeat it in some way.

The Need for PQC or Quantum-safe Cryptography(QSC)

With quantum computers anticipated to eventually surpass classical computers, there is a pressing need to transition to quantum-resistant cryptographic algorithms. PQC aims to develop algorithms that remain secure even against quantum computational capabilities.

Basic Principle of QSC

Current prime factorization-based crypto are affected by Shor's algorithm

Various efforts to find "harder" scheme — NP-hard problems

QSC takes advantage of different area of mathematics

- Lattice-based cryptography — rely of hardness of shortest vector problem (SVP) and closest vector problem (CVP) in a lattice structure
- Code-based cryptography — based on difficulty of decoding general linear code
- Multivariate polynomial cryptography — Hidden Field Equation field
- Hash-based cryptography — Merkle signature
- Isogeny-based cryptography — takes advantage of algebraic properties of elliptic curves

NP-hard problems

While there are many known NP-hard problems, not every such problem is suitable as a basis for cryptographic security. In this context, the notion of ***average-case hardness*** is useful for cryptography. A problem is **average-case hard** if most instances of the problem drawn randomly from some distribution are hard, whereas a problem is **worst-case hard** if it is hard only on some isolated *worst-case* instances. Quantum-safe cryptologists therefore search for mathematical problems that satisfy the assumption of average-case hardness and employ theoretical tools such as worst-case to average-case ***reductions*** to identify suitable protocols whose security and efficiency can be guaranteed.

Computational complexity

- In cryptography, the *computational complexity class* ***NP (non-deterministic polynomial time)*** plays an important role. This class consists of decision problems for which proposed solutions can be verified in polynomial time using a ***Deterministic Turing Machine (DTM).*** The importance of NP stems from the fact that it is conjectured to consist of many computational problems that cannot be solved efficiently by both classical and quantum computers.
- The first generation of successful asymmetric key cryptosystems developed in the 1970s based their security on mathematical problems such as prime factorization and discrete logarithms that are now conjectured to belong to the **NP-intermediate** subclass of NP. This subclass consists of problems that are believed not to have polynomial-time solutions on DTMs but at the same time are also not as hard as the hardest problems in NP.
- The latter belong to the subclass ***NP-complete***. Following ***Shor's algorithm*** in the 1990s, it became clear that at least some NP-intermediate problems are amenable to efficient solutions on quantum computers that are not DTMs.

- Therefore, modern quantum-safe cryptography schemes are based on NP-complete problems or related ***NP-hard*** problems, which currently are not known to be solvable efficiently even on quantum computers.

Types of Post-Quantum Cryptographic Algorithms : Mathematical structures

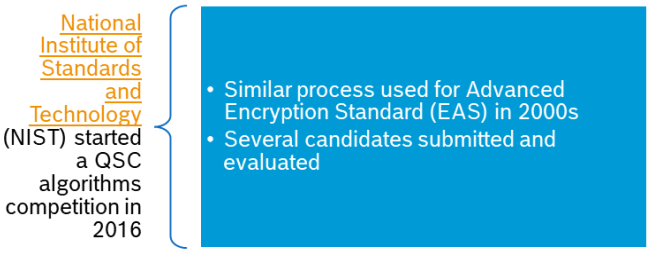
Cryptologists have put forth a number of different mathematical structures that satisfy the necessary hardness requirements as potential candidates for quantum-safe migration of asymmetric key cryptosystems. Some well-known families include:

- **Lattice-based cryptography**: ***Lattice-based cryptography*** class of algorithms relies on the hardness of problems such as the shortest vector problem (SVP) and the closest vector problem (CVP) in ***lattice structures***. Notable lattice-based schemes include and ***NTRU*** and ***Learning with Errors*** (LWE).
- **Code-based cryptography**: ***Code-based cryptography*** is based on the difficulty of decoding a general linear code. The most notable example is the ***McEliece cryptosystem***.
- **Multivariate cryptography**: ***Multivariate cryptography*** involve equations of multiple variables over a finite field. A well-known system in this category is the HFE (Hidden Field Equations) scheme.
- **Hash-based cryptography**: ***Hash-based cryptography*** use only cryptographic hash functions. They are often used for digital signatures, like the Merkle signature scheme.
- **Isogeny-based cryptograph**y: ***Isogeny-based cryptography*** are based on the difficulty of certain problems in the algebraic structure of elliptic curves. ***Supersingular Isogeny Diffie-Hellman*** (SIDH) is an example.

Homomorphic Encryption

An innovative concept arising from some PQC algorithms is **Homomorphic Encryption**. This technique allows computations on encrypted data without decrypting it. Applications include privacy-preserving data analysis and secure voting, offering significant advancements in data security and processing.

NIST Standardization of QSC



NIST Initiative

- NIST announced a list of **four finalists**
- **Three are lattice-based** - Lattice-based cryptography therefore seems well-positioned to form the basis for the first-generation of QSC standards.
- **CRYSTALS-Kyber, CRYSTALS-Dilithium** - part of the **CRYSTAL cryptographic suite** , were selected to be general-purpose protocols for **key encapsulation and digital signatures**, respectively.
- **FALCON** - Recommended for applications requiring smaller digital signatures than those provided by Dilithium.
- One hash-based (**SHPINCS+**) — selected as a backup as it uses different area of mathematics.

QSC Algorithm	Cryptographic family	Application
CRYSTALS-Kyber	Lattice-based	Key encapsulation mechanism
CRYSTALS-Dilithium	Lattice-based	Digital signatures
FALCON	Lattice-based	Lightweight digital signatures
SPHINCS+	Hash-based	Digital Signatures

Finalists of the first NIST quantum-safe cryptography standardization effort

- In August 2023, **NIST published three draft standards** for comments - which include the algorithms above:
- **FIPS 203, Module-Lattice-Based Key-Encapsulation Mechanism Standard**
- **FIPS 204, Module-Lattice-Based Digital Signature Standard**
- **FIPS 205, Stateless Hash-Based Digital Signature Standard**

Lattice-Based Cryptography

As the name suggests, ***lattice-based cryptography*** (LBC) is based on the hardness of certain problems defined on mathematical structures called ***lattice***.

Of fundamental importance are two computational problems on lattices, namely the **shortest vector problem** and the **learning with errors problem**, which we discuss below after some preliminary definitions.

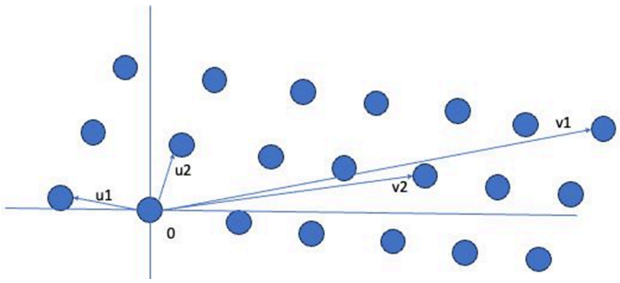
Conceptual Overview:

- **Lattice Structure:** Imagine a multidimensional grid where lattice points represent possible values. Cryptographic problems involve finding short vectors in this lattice.
- **Example Analogy:** Lattices can be observed in various forms around us. For instance, notable structures like the Eiffel Tower in Paris and the Bird's Nest Stadium in Beijing embody the concept of a lattice. These structures consist of numerous interconnected elements that can be visualized as a network of points connected by straight lines.
- Diamonds provide another example of a lattice, where the clear, repetitive structure of carbon atoms bonded together forms a crystal. This specific arrangement grants diamonds their unique properties. However, different crystal structures, which can also be thought of as types of lattices, exhibit varying layouts and, consequently, distinct characteristics. This is why lattices are crucial in fields like cryptography, as they form the mathematical basis that influences the properties of various structures.
- From a mathematical perspective, a lattice is essentially a set of points spaced regularly at fixed intervals. The positions of these points relative to one another are defined using vectors, with the specific arrangement of these vectors known as the basis of the lattice.
- Consider a box of toy construction bricks as an analogy. Building different structures with these bricks requires a specific arrangement of pieces, which serves as the basis for each unique construction. Changing the selection or arrangement of these bricks results in a different structure with distinct characteristics.
- Similarly, in a lattice, the basis acts like a set of fundamental building blocks that define the structure of the lattice. By varying the arrangement of these building blocks (atoms or points), you can generate different lattice structures, each with unique properties. Just as altering the toy bricks changes the built object, modifying the basis alters the characteristics and properties of the lattice.
- It's important to note that lattices are not confined to two or three dimensions; they can extend into much higher dimensions. In advanced applications like cryptography, lattices might encompass thousands of dimensions.
- **Chessboard Lattice:** Visualize a knight moving on an infinite chessboard. The challenge is to find moves that approximate a

target position, reflecting the problem-solving nature of lattice-based cryptography.

Further Concepts- LBC

- Not every basis is unique - it may just be a different perspective of the same structure.
- This leads to an important concept in lattice mathematics, that of ***lattice-basis reduction***. This is the process of taking a given integer lattice and attempting to find a good basis comprising short, nearly orthogonal vectors.



Lattice-basis reduction in two dimensions from a "bad" basis {v1, v2} to a "good basis" {u1, u2}

Lattice-basis reductions can be performed in polynomial-time using the ***Lenstra-Lenstra-Lovasz*** (LLL).

Current and Future Outlook

Adoption of PQC:

- **NIST Standardization:** NIST selected **CRYSTALS-Kyber and Saber** for **Key Encapsulation(KEM)** and **CRYSTALS-Dilithium and FALCON** for **Digital signatures.** This marks a significant step towards integrating PQC into real-world applications.

Risks and Challenges:

- **Algorithm Robustness:** While algorithms like Kyber are currently considered quantum-resistant, no algorithm is entirely quantum-proof. The field is still evolving, and new vulnerabilities may emerge.
- **Early Adoption:** Some PQC algorithms, have faced issues, emphasizing the need for extensive testing and validation.

Practical Implementations:

- **Open Quantum Safe Project:** Develops libraries like liboqs for PQC encryption, widely used across various projects.
- **Signal Protocol:** Utilizes PQXDH, based on Kyber, enhancing encryption in applications like WhatsApp and Google RCS.
- **Microsoft and Cloudflare:** Both are integrating PQC into widely used encryption libraries and systems, ensuring resistance against future quantum threats.

Module-LWE and the CRYSTALS suite

The learning with errors (LWE) problem, introduced in a simplified form above and generally valid on arbitrary lattices, has been extended to algebraic rings within the so-called **Ring-LWE** framework primarily to improve the efficiency of resulting cryptosystems. However, the extra algebraic structure of Ring-LWE may be potentially exploitable, even though no such exploits are currently known.

Two of the four finalists in NIST's QSC standardization process — namely, the **CRYSTALS-Kyber** key encapsulation mechanism (KEM) and the **CRYSTALS-Dilithium** digital signature protocol — are based on structures known as ***Module lattices*** and the related ***Module-LWE***.

Key Encapsulation Mechanisms and CRYSTALS-Kyber

Traditional asymmetric key cryptosystems are most heavily deployed for their key-exchange and ***digital signature*** functionalities and as such, the NIST standardization process sought to develop quantum-safe alternatives for these two functionalities.

The ***CRYSTALS-Kyber*** protocol is therefore designed as a dedicated ***Key Encapsulation Mechanism (KEM)*** rather than as a general-purpose encryption scheme such as RSA.

IND-CPA and IND-CCA security in lattice-based cryptography

§Traditional cryptosystems whether symmetric (such as AES) or asymmetric (such as RSA) use deterministic functions to implement encryption operations. This means that a given plaintext, combined with a given encryption key will always encrypt to the same ciphertext. Such deterministic cryptosystems are vulnerable to chosen plaintext attack whereby an adversary is able to extract information by requesting encryptions of arbitrary plaintexts of their choice from the deterministic encryption function.

To achieve IND-CPA security in this context, ***additional randomness*** is introduced at encryption time either through ***initialization vectors or padding***. For instance, AES is only IND-CPA secure when used in ***Cipher Block Chaining(CBC) or Gaolis/ Counter Mode(GCM) modes of operation*** that use random initialization vectors. Similarly with RSA, ***OAEP padding*** is need to ensure IND-CPA security.

In contrast, lattice-based schemes for encryption are inherently randomized due to the problem definition itself. In particular, in the LWE based encryption scheme outlined above, there are two distinct elements of randomness:

(1) The error (or noise) ε drawn from the distribution **X**

(2) The random binary vectors r∈{0,1}N used for encrypting each bit in the message.

The errors ε contribute to the security of the public key, ensuring that it's computationally hard to deduce the secret key ***s***. The random binary vectors ***r*** on the other hand provide the essential randomness needed for making repeated encryptions of the same plaintext bit non-deterministic. Thus, LWE based schemes are considered IND-CPA secure without the need for external mechanisms such as padding.

Modern cryptosystems aim to achieve so called ***IND-CCA*** security which stands for **indistinguishability under chosen-ciphertext attack**. In this setting the adversary has the ability to obtain decryptions of a non-trivial set of ciphertexts of their choosing with the aim of extracting information to subsequently break the cryptosystem. A scheme is IND-CCA secure if, even with this capability, the adversary cannot do better than random guessing when trying to distinguish encrypted messages. IND-CCA is a stronger security notion than IND-CPA and subsumes it.

Quantum safe KEMs such as **Kyber** are designed to be IND-CCA secure. This is achieved in two steps:

-An IND-CPA secure public key encryption(PKE) scheme is defined. In the case of Kyber such a PKE is based on **Module-LWE**.

-A variant of the ***Fujisaki-Okamoto Transform*** (FO) is applied to obtain a CCA-secure KEM. The FO transformation is a generic method to convert encryption schemes that are IND-CPA secure into ones that are IND-CCA secure. For details we refer readers to the ***Original papers***.

For more information on the security features of **Kyber** and **Dilithium**,
- **CRYSTALS suite documentation**.

Quantum-Safe Cryptography: Future Directions

The looming threat of quantum computing necessitates the development of quantum-safe cryptographic standards. Researchers are exploring several approaches to ensure cryptographic resilience against quantum attacks:

- **Lattice-Based Cryptography:** Relies on the computational hardness of lattice problems, such as the Shortest Vector Problem (SVP) and Learning with Errors (LWE). Notable examples include CRYSTALS-Kyber for key encapsulation and CRYSTALS-Dilithium for digital signatures.
- **Code-Based Cryptography:** Based on the complexity of decoding linear codes. The McEliece cryptosystem exemplifies this approach.
- **Multivariate Cryptography:** Involves solving systems of multivariate polynomial equations. The Hidden Field Equations (HFE) scheme is a prominent example.
- **Hash-Based Cryptography:** Utilizes cryptographic hash functions for digital signatures, as seen in the Merkle signature scheme.
- **Isogeny-Based Cryptography:** Exploits algebraic properties of elliptic curves, exemplified by the Supersingular Isogeny Diffie-Hellman (SIDH) protocol.

Resources:

- **NIST Post-Quantum Cryptography Standardization** - Information about NIST's ongoing efforts to standardize quantum-resistant cryptographic algorithms.
- **Quantum-Safe Cryptography: A Survey** - Comprehensive survey on various quantum-safe cryptographic approaches and their practical implications.

Implementing PQC in Automotive Systems

1. Assessment and Planning

Implementing PQC necessitates a comprehensive assessment of the existing cryptographic infrastructure in vehicles. This includes identifying potential vulnerabilities and understanding the quantum threat landscape. The planning phase involves:

- **Identifying Critical Systems**: Prioritizing systems needing immediate attention, such as communication channels, software updates, and safety-critical systems.
- **Selecting PQC Algorithms**: Choosing appropriate PQC algorithms tailored to the vehicle's specific needs, considering factors like computational efficiency and implementation complexity.
- **Hardware Security Modules (HSM):** With integrated security capabilities, including secure boot, ***cryptographic acceleration***, and TrustZone support to ensure data protection and secure operation. The module should be equipped with tamper detection mechanisms to protect against physical attacks.

2. Integration into Vehicle Systems

The integration of PQC into automotive systems involves several key steps:

- **Software Updates**: Developing and deploying software updates that replace traditional cryptographic algorithms with PQC algorithms. This requires seamless collaboration with software developers and vehicle manufacturers.
- **Hardware Adaptations**: In some cases, hardware modifications may be necessary to support the computational demands of PQC algorithms. This could involve upgrading processors or adding dedicated cryptographic hardware.
- **Testing and Validation**: Rigorous testing is essential to ensure that new PQC algorithms do not introduce vulnerabilities and perform efficiently within the vehicle's ecosystem.

3. Post-Deployment Monitoring and Maintenance

Continuous monitoring post-deployment is crucial to maintain the security and performance of PQC implementations. This includes:

- **Regular Security Audits**: Conducting periodic audits to identify and mitigate new vulnerabilities.
- **OTA Updates**: Utilizing over-the-air updates to quickly deploy patches and improvements to PQC algorithms.
- **Incident Response Planning**: Establishing robust incident response protocols to address any security breaches or performance issues.

Challenges and Considerations

1. Performance Impact

PQC algorithms can be computationally intensive, potentially impacting vehicle system performance. Optimizing these algorithms to ensure they do not degrade user experience is a critical challenge.

Algorithm	Classical security level	Public key size	Private key size	Signature size
FALCON-512	120	897	1281	666
FALCON-1024	277	1793	2305	1280
CRYSTALS-Dilithium 2	121	1312	2544	2420
CRYSTALS-Dilithium 3	176	1952	4016	3293
CRYSTALS-Dilithium 5	253	2592	4880	4595
RSASSA-PSS 3072 bit	128	384	384	384
ECDSA with P-256	128	32	32	64

Key Size

- **Processing Power**: Vehicles must manage the increased computational load introduced by PQC algorithms. This may necessitate more powerful processors or specialized cryptographic hardware accelerators, increasing costs and energy consumption.
- **Flash Memory**: Some PQC algorithms, particularly those with large key sizes like code-based cryptography, may require significant storage space. Ensuring that vehicles have sufficient flash memory without impacting other functionalities is essential.
- **Hardware Security Modules (HSM)**: ECUs present a unique challenge, equipped with integrated security features such as secure boot, cryptographic acceleration, and TrustZone support to safeguard data and ensure secure functionality. These modules are also required to have tamper detection mechanisms to guard against physical threats, enhancing their defensive capabilities in critical automotive environments.

2. Standardization and Interoperability

The field of PQC is still evolving, with multiple competing algorithms. Ensuring interoperability between different systems and adhering to emerging standards is crucial for widespread adoption.

3. Future-Proofing

Quantum computing is a rapidly advancing field, with new threats likely to emerge. Continuous research and development in PQC, alongside staying updated with advancements in quantum computing, are necessary to future-proof vehicle security.

Use Cases Learnings in Automotive Systems

Scheme Suitability

Given the challenge posed by Shor's algorithm in breaking asymmetric keys, it is crucial to evaluate two primary features and their corresponding post-quantum cryptographic (PQC) schemes:

Key Encapsulation Mechanisms (KEM): CRYSTALS-Kyber and Saber

- **CRYSTALS-Kyber:** This PQC scheme is highly regarded for its balance between security and performance. CRYSTALS-Kyber uses lattice-based cryptography, offering strong resistance against quantum attacks. It is designed for efficiency in both key exchange and encapsulation processes, making it suitable for secure communication over networks where latency and speed are critical. Additionally, Kyber's relatively low resource utilization makes it adaptable to a range of devices, from high-end servers to more constrained IoT devices.
- **Saber:** Saber is another lattice-based KEM that focuses on security and efficiency. It is characterized by its use of Module-LWR (Learning With Rounding) problems, which contribute to its robustness against quantum computing threats. Saber is particularly noted for its fast key generation and encapsulation times, which are vital for real-time applications. Its implementation is streamlined, which allows for effective use in environments with limited processing power and memory, such as embedded systems and mobile devices.

Digital Signatures: CRYSTALS-Dilithium and FALCON

CRYSTALS-Dilithium: Renowned for its robust security, CRYSTALS-Dilithium is especially suitable for environments where security is a top priority, even at the expense of longer execution times. Due to its larger memory requirements, it is optimal for high-performance electronic control units (ECUs) or multifunctional embedded systems that have access to substantial resources.

FALCON: Known for its rapid verification times and smaller key and signature sizes, FALCON excels in settings that demand fast cryptographic processes, such as secure boot processes or real-time secure communications. Its minimal stack consumption and reduced code size make it an excellent option for dedicated embedded ECUs or contexts where operational efficiency is essential.

Benchmarking Post-Quantum Cryptography in Automotive Embedded Control Units

As the automotive industry evolves towards greater connectivity and automation, ensuring robust data security becomes increasingly vital. With the advent of quantum computing, traditional cryptographic schemes face new challenges. To address these, post-quantum cryptographic (PQC) schemes are being evaluated for their performance across various embedded control units (ECUs) commonly used in automotive systems. This article explores the implications of these evaluations, focusing on execution time, stack consumption, and code size across different hardware platforms.

Understanding Automotive ECUs

Three category of ECUs were taken for the experiment as classified based on their function and performance characteristics:

High-Performance ECUs: High-Performance ECUs

- **Processing Speed:** The high speeds (900 MHz to 1.5 GHz) of these ECUs enable them to handle computationally intensive algorithms, making them well-suited for more demanding post-quantum cryptographic schemes.
- **Core RAM:** With 1 GB to 4 GB of RAM, these systems can support both memory-intensive algorithms and larger code sizes.
- **Flash Memory:** The external micro SD slot provides additional storage, accommodating larger cryptographic libraries.

For high-performance ECUs, post-quantum schemes like **CRYSTALS-Dilithium** are suitable due to their robust security features and compatibility with larger memory and processing capabilities. However, the choice depends on the trade-offs between security requirements and resource availability.

Multi-Purpose Embedded ECUs:

- **Processing Speed:** Operating at 300 MHz, these ECUs offer moderate processing capabilities, balancing performance with power efficiency.
- **Core RAM:** With 992 KB of RAM per core, they can handle algorithms with moderate memory and stack requirements.
- **Flash Memory:** The 6 MB of flash memory is adequate for storing cryptographic algorithms and their data.

For multi-purpose systems, **FALCON** might be preferred due to its smaller key and signature sizes and faster verification times. This efficiency makes it a strong candidate for applications where moderate resources and quick processing are required.

Single-Purpose Embedded ECUs:

- **Processing Speed:** At 20 MHz, these ECUs have limited processing power, suitable for simpler, lightweight cryptographic operations.
- **Core RAM:** With 192 KB of RAM per core, managing stack consumption is crucial to avoid exceeding memory limits.
- **Flash Memory:** The 6 MB of flash is sufficient for storing essential cryptographic functions and data.

In single-purpose ECUs, **FALCON**'s efficiency in terms of stack consumption and code size makes it particularly suitable for real-time operations, such as secure boot, where quick verification is crucial.

Performance Metrics for Cryptographic Operations

1. Execution Time (ms): Execution time is a critical factor for real-time applications. It refers to the time required to perform cryptographic operations. For automotive systems, minimizing execution time is crucial, especially for tasks that demand immediate response, such as secure boot processes.

2. Stack Consumption (bytes): Stack consumption measures the memory required for cryptographic operations. Efficient use of stack memory ensures that the ECU can handle complex operations without running into memory constraints, which is especially important for systems with limited RAM.

3. Code Size (bytes): The code size includes the .text, .data, and .bss sections of the compiled binary. Efficient code size is essential for

compatibility with embedded systems, which often have strict memory limitations.

Evaluating post-quantum cryptographic schemes on various automotive ECUs highlights significant differences in performance, memory requirements, and code size. High-performance ECUs, with their substantial processing power and memory, can accommodate more complex algorithms like CRYSTALS-Dilithium. In contrast, multi-purpose and single-purpose ECUs require cryptographic solutions that balance efficiency with available resources. Lets check there performance for specific use case:

Key Management Systems (KMSs)

Use Case: Secure Key Distribution

In automotive contexts, ECUs (Electronic Control Units) require secure key material from backend systems. The process involves:

1. **Key Generation and Distribution**: Keys are securely generated in a Trusted Execution Environment (TEE) or Hardware Security Module (HSM).
2. **Key Injection**: The key material is injected into ECUs via a diagnostic interface. The ISO 15765-2 standard facilitates efficient key material transmission.
3. **Secure Storage**: Keys are stored in TEEs or HSMs within ECUs. Most of the ECUs require adjustments to storage sizes to accommodate post-quantum key material.
4. **Cryptographic Library Updates**: Updating cryptographic libraries to include post-quantum schemes increases flash memory usage but ensures security.

Use Case: Enhanced Security in Manufacturing

During manufacturing, ECUs receive key material through secure channels to ensure authenticity and integrity. Implementing algorithms like CRYSTALS-Dilithium or FALCON enhances security, despite increased memory and processing demands.

Secure Boot

Use Case: Firmware Integrity Verification

Secure boot ensures firmware integrity using asymmetric cryptography:

1. **Digital Signatures**: Firmware is signed with a private key, and the public key verifies the signature.
2. **Certificate Chains**: Verification through a certificate chain ensures hierarchical trust.
3. **Boot Time Optimization**: Efficient algorithms like FALCON, with faster verification times and lower memory requirements, are preferred.

Use Case: Post-Quantum Secure Boot

Adapting secure boot processes to post-quantum cryptography involves managing increased key sizes and verification times. Hardware acceleration may be necessary to meet boot time requirements.

Secure Diagnostic Access

Use Case: Preventing Unauthorized Access

Secure diagnostic access prevents unauthorized ECU access:

1. **Challenge-Response Mechanism**: Ensures only authorized diagnostic tools access ECUs.
2. **PKI Certificate Exchange**: Enhances security by verifying access requests.
3. **Production Impact**: Post-quantum schemes can increase production time. Algorithms like FALCON, with smaller key sizes and faster verification times, mitigate this impact.

Use Case: Efficient Vehicle Production

Securing diagnostic access efficiently is crucial to maintaining productivity. Choosing algorithms like FALCON, which balance security and performance, helps manage verification times.

Transport Layer Security (TLS)

Use Case: Secure Vehicle-to-Backend Communication

TLS secures vehicle-to-backend communication:

1. **Handshake Process**: Verifies digital signatures and derives symmetric keys. Post-quantum cryptography increases handshake message size and complexity.
2. **Data Volume and Costs**: Larger handshake messages due to post-quantum elements increase data volumes and costs.
3. **Performance Impact**: Post-quantum schemes may affect service responsiveness. FALCON offers faster verification times for server-side authentication, while CRYSTALS-Dilithium is preferred for mutual authentication.

Conclusion

As the automotive industry accelerates towards increased connectivity and advanced digital integration, the imperative for robust cybersecurity measures becomes undeniable. The integration of Post-Quantum Cryptography (PQC) emerges as a pivotal strategy to fortify vehicle systems against the formidable threat posed by quantum computing capabilities. With traditional cryptographic defenses likely to falter under quantum advancements, PQC offers a beacon of security, ensuring that automotive technologies can withstand future cyber threats.

In this landscape, the adoption of specific PQC schemes such as CRYSTALS-Kyber, Saber, CRYSTALS-Dilithium, and FALCON is tailored to address the diverse security needs of modern vehicles. Each of these algorithms brings unique strengths: CRYSTALS-Kyber and Saber excel in key encapsulation with their efficient, lattice-based frameworks, making them ideal for protecting real-time communication within vehicular networks. Meanwhile, CRYSTALS-Dilithium and FALCON enhance digital signature applications, with FALCON's rapid verification capabilities being particularly crucial for operations requiring swift cryptographic responses, such as secure boot processes.

The transition to PQC not only aligns with the evolving technological landscape but also underscores a proactive approach to cybersecurity. By embedding these quantum-resistant algorithms into the fabric of automotive systems—from key management to secure boot and diagnostic access—manufacturers ensure that vehicles remain secure, functional, and resilient against the quantum threats on the horizon.

As the field of quantum computing continues to mature, ongoing research, rigorous testing, and adaptation of PQC will be critical. The automotive industry's commitment to this innovative cryptographic paradigm not only safeguards its technological advancements but also

reinforces the trust and safety of its end-users, steering towards a secure, quantum-resistant future.

Santosh Kumar FIP, CISSP, PMP, CISA, CHFI, AIGP

Cybersecurity & Data Protection Leader | CISO & DPO | GenAI Architect | Fellow of Information Privacy (FIP) 🏫 IIT Madras| IIM Indore

The Critical Role of Post-Quantum Cryptography in the Age of Quantum Computing

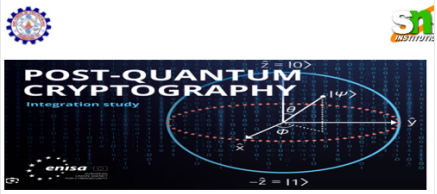
Mehul Jaiswal

16 · 3 comments

Global Leaders in the Post-Quantum Cryptography Transition

Bruno Schneider

5



Securing the Future: Post-Quantum Cryptography

Suguna Muthusamy

4

Questions?
Visit our Help Center.

Manage your account and privacy
Go to your Settings.

Recommendation transparency
Learn more about Recommended Content.

Select Language
English (English)