Application of Quantum Resistant Cryptography to Smart Grid deployments

David Domingo Martín[1,*,†], Rodrigo Martín Sánchez-Ledesma [1,†] and Marta Irene García Cid[2,†]

[1] *Indra Sistemas de Comunicaciones Seguras, 28108 Madrid, Spain*
[2] *Indra Sistemas S.A., 28850 Madrid, Spain*

Abstract
The advances in Quantum Computers, and the risks they pose to current asymmetric cryptography, have triggered the development of a new set of cryptographic algorithms (PQC, which stands for Post Quantum Cryptography), whose underlying mathematical problems are resistant to both quantum and classical computers. These new cryptographic schemes raise some challenges to current deployments, including in environments with performance limitations. This is the case for Smart Grid networks, and more specifically for electricity Smart Meters, usually provided with low processing capacity and deployed in low bandwidth networks. This work, which is based in one of the Use Cases defined in the HORIZON Europe PQ-REACT project, is focused on the inclusion of quantum resistance in Smart Grid deployments by means of the application of PQC Digital Signatures to the Smart Meters Firmware Upgrade process.
This paper fits in the project Eraclito of the SERICS foundation.

Keywords
Post Quantum Cryptography, Smart Grid, Digital Signature, Performance

1. Introduction

The security of Asymmetric Cryptography relies on the complexity of some mathematical problems: RSA based on the hardness of prime factorization, DSA based on the hardness of solving the Discrete Logarithm Problem, while Elliptic Curve Cryptography is based in the hardness of solving the ECDLP (Elliptic Curve Discrete Logarithm Problem). All these hard problems can be efficiently solved by means of Quantum Computing, in particular with a Quantum Algorithm introduced by Peter Shor [1]. This algorithm, by means of several efficient "classical steps", basically reduces the above problems to a "quantum step" involving a Quantum Fourier Transform, which can be efficiently computed in a Quantum Computer.

Current Quantum Computers hardware is in the order of hundreds of qbits, but a Cryptographically Relevant Quantum Computer (CRQC), referred to Quantum Computers with the capacity of solving the cryptographic instances deployed in real life, is estimated to require millions of physical qbits, not only because of the qbits strictly required to implement Shor's algorithm, but also because of the extra qubits required to perform fault-tolerant quantum computations.

Even considering this lack of a current CRQC, to estimate when it is necessary to start considering Quantum Resistant Cryptography it is relevant to consider how long it is required for the information being protected today to remain secure. This is specifically tackled by Mosca's Theorem [2]:

- Information needs to be secure for at least X years.

Joint National Conference on Cybersecurity (ITASEC & SERICS 2025), February 03-8, 2025, Bologna, IT
[*] Corresponding author.
[†] These authors contributed equally.

ddomingo@indra.es (D. Domingo); rmsanchezledesma@indra.es (R. Martín); migarcia@indra.es (M. García)
0009-0001-6184-494X (D. Domingo); 0009-0001-1845-2959 (R. Martín); 0000-0002-6343-1321 (M. García)

© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR-WS.org/Vol-3962/paper6.pdf

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

- Migrating to PQC Cryptography is expected to take Y years.
- The development of a CRQC is estimated to require Z years.
- Mosca's Theorem states that the time that the information needs to be secure X, added to the time required to migrate to Quantum Resistant Cryptography Y MUST be smaller than Z.

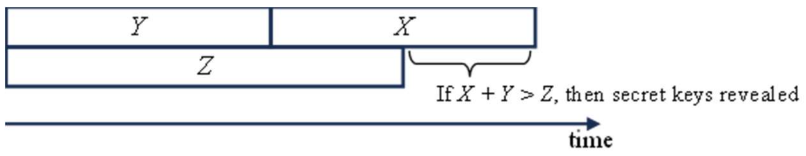

Figure 1: Mosca's Theorem

Finally, to complete the picture, and in order to approximate the value of Z, current relevant estimations [3] conclude that the probability of the quantum threat is far from negligible in a time frame of 10 years, and substantially increased when considering time frames of 15 and 20 years, where the conclusion is that the possibilities of real CRQC is more likely than not.
During December 2016 NIST announced Formal Call for Proposals [4] initiating the process to standardize quantum resistant asymmetric cryptographic algorithms for Digital Signatures and Key Establishment (KEM) schemas.
Depending on the underlying mathematical problem, the PQC algorithms considered for the standardization process can be divided in several post quantum mathematical families.

Lattice-Based Cryptography: Lattices are, conceptually, grids of points regularly distributed in a space. A lattice can be defined by means of a basis (a small set of vectors used to depict the full grid) in a certain vector space. There exists an infinite number of basis that can represent the same lattice, and some basis can be considered good/short basis (allowing easy operation inside the lattice) and some other basis can be considered bad/long basis (NOT allowing easy operation inside the lattice). Provided with a lattice defined by a specific basis, several hard problems can be defined [5], as for instance the Short Vector Problem (SVP) to find the closest lattice point with respect to another given point in that lattice.

Code-Based Cryptography: Code-based Cryptography is based in error correcting codes. When communicating a message errors (bit flips) can occur during the transmission of the message, and error correcting codes allow to fix a certain number of these errors, at the cost of extra message size. One specific type of error correcting codes is linear-codes, which can be depicted as k x n matrices (where k represents the length of the transmitted message and n represents the length of the encoded message). Provided with an encoded message, it is considered a hard problem to decode the message if the underlying linear code is not known, and based in this problem Code-based cryptography schemas can be developed [6].

Multivariate Cryptography: These algorithms are based on the difficulty of solving systems of multivariate polynomials (usually quadratic) over finite fields (MQ problem) [7]. To build this kind of schemas a special structure of the Public Key is required in order to create a hard trapdoor function. This reduces the "hard to solve problem" space and leads to vulnerabilities in these schemas, requiring specific designs, like UOV (Unbalanced Oil and Vinegar Signatures) [8], to remain secure. UOV schemas are based in dividing the system variables in two classes, Oil variables and Vinegar variables, and the polynomials are built so that Vinegar Variables are mixed with all other variables, but Oil Variables are never mixed with themselves.

Isogeny-Based Cryptography: This kind of Cryptography is based on Elliptic Curves, but instead of being based on point/integer multiplication in a curve, as the ECDLP problem underlying Elliptic Curve Cryptography, it is based on hard problems on graphs and isogenies between Elliptic Curves [9]. An Isogeny between two Elliptic Curves can be defined as rational map between them, where the correspondence between coordinates of the points in the curves is defined by some specific relation. This rational map is an Isogeny Φ if it preserves addition, that is, if Φ is an Isogeny between two Elliptic curves E_1 and E_2 (Φ: $E_1 \longrightarrow E_2$), then it is maintained when adding points P and Q in the curves: $\Phi (P + Q) = \Phi (P) + \Phi (Q)$. In this schema, defined over an Elliptic Curve E and some common points P and Q, the private key is the isogeny Φ, and the public keys are the resulting Elliptic Curve E'= $\Phi (E)$ and the coresponding isogeny calculated points $P' = \Phi (P)$ and $Q' = \Phi (Q)$.

Symmetric-Based Cryptography: The basic schema for using symmetric primitives such as hash functions as base for asymmetric signatures come from Lamport's One Time Signature Schemes (OTS) [10]. The idea is to pre-determine the values to be signed, generate a random string as Signature for each value, and publish the hash of the random values as Public Keys. This elementary design is improved by several subsequent designs (Winternitz One Time WOTS & WOTS+ schemas [11] significantly reducing the size of the signature, Merkle Trees and the Merckle Signature Scheme [12] to overcome key management difficulties because one key can being used to sign just one message) allowing the definition of practical Digital Signature schemas.

From the initial Call NIST stated its intention of not choosing just one winner in each category, based in the fact that most of the different new types of cryptography being developed are still in early stages of their development. Most of the PQC schemes are relatively new, and the cryptoanalysis of the various underlying problems is still evolving, with non-negligible probabilities of some of them to be broken.

During the initial phase, beyond exposing the requirements and evaluation criteria to be considered during the algorithm's selection process (availability of the algorithm specification, Intellectual property requirements, implementation characteristics and costs, etc.), NIST also publicized the expected security levels of the algorithms from 1 to 5.

NIST process continued through several rounds until the publication of the "*Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process*" announcing the final chosen algorithms (one KEM and three Digital Signature), and a set of alternative 4th round KEM candidates to be subject to further analysis and later standardization decision [13]:

Table 1
NIST Finalist PQC Algorithms Comparison

Type	Status	Algorithm	Lattice Based	Code Based	Isogeny Based	Symmetric Based
KEM	Standard	Kyber [14]	x			
	Alternate	Bike [15]		x		
		McEliece [16]		x		
		HQC [17]		x		
		SIKE[18]			x	
Digital Signature	Standard	Dilithium [19]	x			
		Falcon [20]	x			
		SPHINCS$^+$ [21]				x

For the Key Establishment algorithms Dilithium-Kyber was chosen as primary recommendation. It was the preferred algorithm between the other Lattice-Based candidates, that were discarded in order to proceed with further analysis during the 4th Round of alternative Code-Based candidates: Bike, McEliece and HQC. & Isogeny-Based: SIKE). After the NIST announcement SIKE, the only Isogeny-Based candidate, initially included in the 4th round, was discarded because of a Key Recovery Attack [22].

Regarding the Digital Signature algorithms, before the NIST publication of the chosen candidates a relevant vulnerability affecting the Multivariate-Based algorithm Rainbow [23] was published, making Key Recovery attacks feasible [24], and raising general concerns over other Multivariate Algorithms. Because of this, NIST decided to standardize the two remaining Digital Signature finalists, CRYSTALS-Dilithium and Falcon, both from the Lattice-Based family. Also, in order not to keep in the standardization track Lattice-Based Digital Signature algorithms, NIST decided to standardize the 3rd Round Alternate Candidate SPHINCS+, that being based on Hash functions, instead of other new mathematical problems less understood, is considered as a conservative candidate, less prone to unforeseen future weaknesses, but with some size/performance issues when compared with other algorithms. Because of the lack of remaining alternative Digital Signature candidates NIST, announced a new Call for additional Digital Signature algorithms to be considered in the Post Quantum standardization process [25], with the main to look for schemes not based on structured lattices in order to "diversify the signature portfolio". This additional Digital Signatures process is currently in its 2nd round with 14 remaining candidate algorithms under analysis [26].

The NIST standards for four algorithms were made available during August 2024:

- **ML-KEM** [27]: The Dilithium-Kyber based Module-Lattice-Based Key-Encapsulation Mechanism Standard.

Table 2
ML-KEM Parameter sets

Security Level	Algorithm	Public Key (bytes)	Private Key (bytes)	Signature (bytes)
1	ML-KEM-512	800	1,632	768
3	ML-KEM-768	1,184	2,400	1,088
5	ML-KEM-1024	1,586	3,168	1,586

- **ML-DSA** [28]**:** Module-Lattice-Based Digital Signature Standard derived from the CRYSTALS-Dilithium submission. Selected as the primary algorithm for a Digital Signature Standard, its Lattice-Based schema with conservative parameters while maintaining the simplicity of its design, minimizing the combined size of its Public Keys and Signatures.

Table 3
ML-DSA Parameter sets

Security Level	Algorithm	Public Key (bytes)	Private Key (bytes)	Signature (bytes)
2	ML-DSA-44	1,312	2,560	2,420
3	ML-DSA-65	1,952	4,032	3,309
5	ML-DSA-87	2,592	4,896	4,627

- **SLH-DSA** [29]**:** Stateless Hash-Based Digital Signature Standard derived from the SPHINCS+ submission. Based on a similar architecture to the stateful XMSS signature schema [30], but being based on FTS schemes (Few Times Signature) instead of OTS, where the FTS algorithm can be used to generate signatures a limited number of times, considering that with each usage some private data is exposed, reducing the security of that key. SLH-DSA parameters are specified depending on the underlying hash function family (SHA2 or SHAKE) and whether the parameter set is designed to have relatively small signatures (*s*) or to have relatively fast signature generation (*f*). The table below shows the parameters for the SHAKE and (*f*).

Table 4
SLH-DSA Parameter sets

Security Level	Algorithm	Public Key (bytes)	Private Key (bytes)	Signature (bytes)
1	SLH-DSA-128	32	64	17,088
3	SLH-DSA-192	48	96	35,664
5	SLH-DSA-256	64	128	49,856

- **Falcon:** The standard for this algorithm is not yet available. NIST still working on it because of some relevant issues related with Falcon's implementation [31]. The Key Generation and Signing operations rely on floating point arithmetic and, in order to avoid side-channel vulnerabilities, the algorithm requires these floating-point operations to be constant-time which can be hard to achieve and also very platform-dependent.

Table 5
Falcon Parameter sets

Security Level	Algorithm	Public Key (bytes)	Private Key (bytes)	Signature (bytes)
1	Falcon-512	897	1,281	752
5	Falcon-1024	1,793	2,305	1,280

In Smart Grid deployments the Smart Meter Firmware Update process is usually protected by means of Elliptic Curve Digital Signatures. For instance, DLMS/COSEM Standards [32] specify the usage of Elliptic Curve Digital Signature Algorithm with P-256/P-384 curves [33]. Elliptic Curve cryptography fits in this kind of environments (low processing capacity, low bandwidth, and limited secure storage) because of its performance advantages: lower signature/verification CPU consumption and smaller private/public key sizes.

The migration to PQC Digital Signatures will be mandatory because of two reasons. First, the ***security*** need of addressing the risk of Quantum Computers being able to break the security of classic cryptography. Second, the ***lack of standards supporting classic cryptography*** since the National Institute of Standards and Technology (NIST) plans to disallow Classical Digital Signatures from 2035 [34].

Smart Grid deployments provide the kind of scenario where migrating to PQC Cryptography (or hybrid Classical/PQC schemas) can impose relevant challenges, because of the aforementioned limitations, and because of the properties of the new asymmetric cryptographic algorithms (increased public/private keys sizes, increased signature sizes, increased CPU consumption depending on the algorithm and the implementation).

2. Smart Meters Firmware Update Overview

In Smart Grid deployments, and because of the several limitations to be considered, the underlying security mechanisms usually rely on symmetric cryptography. However, the process of Smart Meter Firmware Update is an exception of that, as it employs asymmetric cryptography due to several reasons:

- Asymmetric cryptography fits in the Firmware (FW) Update procedure as long as the firmware only needs to be signed once by the Utility. So the signing Private Key can be securely managed by the centralized Utility Management Systems and the corresponding Public Key can be distributed to all the Smart Meters in the deployment for verification purposes.
- The FW Update procedure is by itself quite demanding (e.g.: large FW images transmitted by the Data Concentrator to the Smart Meters Network, time-consuming update process once the firmware has been downloaded to the Smart Meter, etc.), so that securing it with Asymmetric Digital Signatures is very efficient, as it adds minimal extra load to the process.
- The FW Update process is not often executed (e.g.: once every year), so the limitations/delays that could be imposed for relying on asymmetric cryptography are not so relevant.

The next figure depicts the steps of the Firmware Update process:

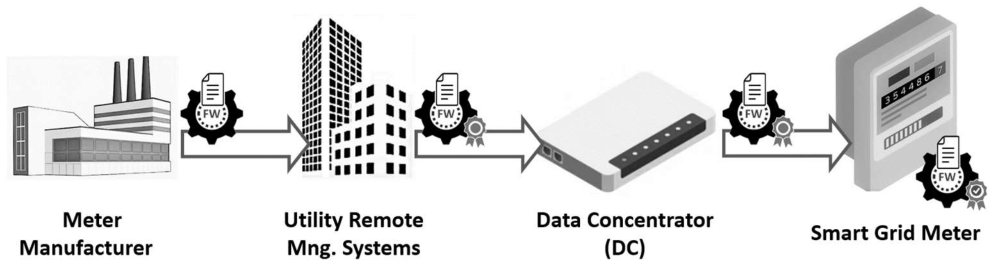

Figure 2: Smart Grid Meter Firmware Update

1. The Smart Meter Manufacturer generates a firmware Image (`FW_IMG`) and sends it to the Utility Remote Management System.
2. Provided with its Private Key (`PRIV_KEY`) the Utility proceeds to sign the FW Image:
   `SIGN(FW_IMG, PRIV_KEY)` ⇒ `FW_IMG_SGN`
3. The Utility forwards the signed Firmware Image (`FW_IMG_SGN`) to the Data Concentrator (DC) controlling the Smart Meters to be updated.
4. The DC sends the signed FW Image (`FW_IMG_SGN`) to the to the Smart Meter to be updated.
5. Provided with the signed Firmware Image (`FW_IMG_SGN`) and with the Utility's Public Key (PUB_KEY), the Smart Meter can verify the validity of the Digital Signature:
   `VERIFY(FW_IMG_SGN, PUB_KEY):`
   - `VERIFICATION_OK`
   - `VERIFICATION_FAILED`
6. Depending on the verification result the Smart Meter proceeds with or halts the Firmware Update process.

2.1. Smart Grid Infrastructure

The equipment & infrastructure involved in this Use Case, and that could be impacted by the migrating the Firmware Update process to PQC Digital Signatures is:

Utility Management Systems. Central systems responsible FW Update process Keys Management involving: the generation of the Public/Private Utility Key pairs, the secure storage and management of the Private Key, and controlling management of the Public Keys in the Smart Meters.

These systems are also responsible for receiving Smart Meters Firmware Images from the manufacturers, executing the FW Images signature process with the corresponding Private Key, and then managing the process to deliver it to the corresponding Smart Meters.

Please note that both processes, Smart Meters Public Keys Management and FW update process, are centrally managed from Utility Management Systems but executed in the deployed Smart Meters via the Data Concentrators.

Data Concentrator (DC). Devices acting as gateways between the Utility Management Systems and the Smart Meters: Connected with the Central systems through regular IP networks and with the specific set of Smart Meters under its control by means of a DLMS/COSEM network.

DCs are responsible for executing the Public Keys Management and FW update process om the Smart Meters, based in the orders and data received from the Utility Management Systems.

Smart Meter. Smart Meters are the receivers of the different management/update orders, responsible for updating the FW Public Key, and securely storing it, when mandated by the DC, and also responsible for performing the FW Digital Signature verification before allowing the FW Update process.

3. Scenarios to evaluate Smart Grid migration to PQC

This section three functional scenarios are introduced to evaluate the impact of migrating the Smart Meters Firmware Update process from Elliptic Curve based Digital Signatures to PQC algorithms.

3.1. Public Key Update

This functional scenario considers the process required to update the Firmware Public Keys deployed in the Smart Meters. This Public Key is expected to be owned by the Smart Meter manufacturer (but also Utility ownership, or mixed Manufacturer/Utility ownership are applicable), and it might be required to be updated in the corresponding Smart Meters by several reasons: Crypto period policy rules, Private Key compromise, Digital Signature algorithm update, etc.…

This process will be directly impacted by the migration to PQC Digital Signature algorithms just by the increment of the sizes of the Public Key, that in the considered environment are distributed to the meters through the low bandwidth DLMS/COSEM DC ⇔ Smart Meters network.
The next figure shows logarithmic graphic with a comparison of the sizes of PQC Digital Signature algorithm, ML-DSA, SLH-DSA and Falcon, with Classical Elliptic Curve algorithms, NIST P and Edwards curves [33]:

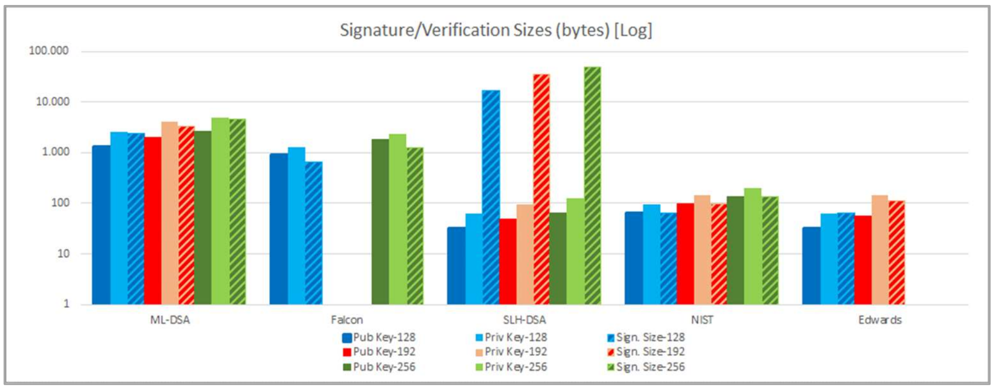

Figure 3: PQC vs Classical Digital Signature Sizes (Public Key, Private Key & Signature Sizes)

3.2. Firmware image delivery to the Smart Meter

The Firmware Update procedure is executed regularly by the Utilities in order to provide new/improved functional capabilities to the Smart Meters.

This process requires the Firmware to be delivered to the Smart Meter, and, as already introduced in the previous scenario, it is distributed through the low bandwidth network connecting DCs and Smart Meters. So, depending on the signature size of the selected PQC algorithm some impact can be expected in this procedure.

3.3. Firmware Signature Verification by the Smart Meter

Once the Firmware Image has been received by the Smart Meter from the DC, before performing the real FW Update the Smart Meter is required to validate the Firmware Digital Signature based in the configured Firmware Public Key. Depending on the specific PQC Algorithm used to sign the FW the time to compute the verification can be impacted.

The next figure shows logarithmic graphic with a comparison of the execution times (µsec.) of PQC Digital Signature algorithms with Classical Elliptic Curve algorithms:

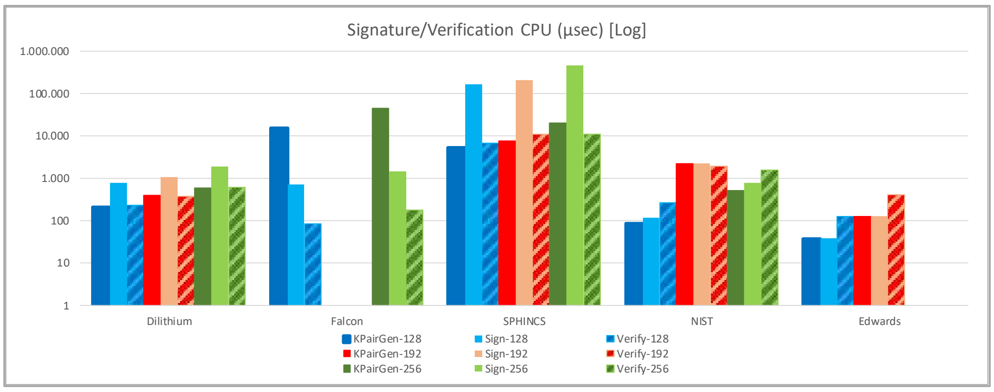

Figure 4: PQC vs Classical Digital Signature Performance (Key Gen., Signature & Verification)

Please note that the above figure must only be considered as reference, as the execution times have been measured in regular computers, results for "*2012 Intel Core i5-3427U*" from [35], and relevant challenges are expected to arise when PQC algorithms are implemented in limited CPU environments as those available in Smart Meters HW. In particular, these figures might suffer changes when measured in the specific HW in which they were to be deployed.

4. Results of the evaluation of PQC Algorithms for Smart Meters FW Update

To evaluate the PQC Digital Signatures applicability into the Smart Grid scenario, this section relies on Key Sizes & Performance metrics obtained from the algorithm's specifications and from [35]. Considering that current Smart Grid deployment protects the Meter's Firmware Image distribution with NIST P-256 Digital Signatures (Security Level 128) the analysis is focused on the performance implications of transitioning to the corresponding Security Level 128 versions of ML-DSA, SLH-DSA and Falcon.

The next table shows the differences between NIST P-256 and its PQC counterparts.

Table 6
NIST P-256 / PQC DSA Algorithms Comparison

Algorithm	Pub. Key (bytes)	Priv. Key (bytes)	Sign. Size (bytes)	Key Pair Gen. (µsec)	Sign. (µsec)	Verify (µsec)
NIST-P256	64	96	64	85	115	261
ML-DSA-87	1,312	2,560	2,420	206	774	233
Falcon-512	897	1,280	752	15,500	727	86
SLH-DSA-1	32	64	17,088	5,240	165,758	6,755

For the analysis Key Pair Generation, Signature Generation and Private Key Sizes are not considered relevant as long as they depend on the Utilities in their Central Management Facilities. Focusing the analysis on the meaningful measurements, the following conclusions are drawn for each of the PQC Digital Signature algorithm:

ML-DSA: Public Key Size (1,312 bytes $\Rightarrow$ +1,950%): Notable cryptographic Key Size increase. High impact can be expected both in Smart Meter secure storage and in Key Update procedures. Specifically, during Key Update procedures the larger keys increase the likelihood of failures during the update process, potentially leaving outdated/compromised keys in use for longer.

Signature Size (2,420 bytes $\Rightarrow$ +3,681%): Huge Signature Size increase. Relevant impact can be expected during FW Update procedures, increasing the chances of failed or longer update processes.

Verification Time: (233 µsec ⇒ -11%): Comparable to classical crypto, so no impact is expected.

Table 7
ML-DSA Pros & Cons

PROS	CONS
• Primary recommendation by NIST • Reduced/no impact in the verification process	• Very increased Public Key & Signature sizes

Falcon: Public Key Size (897 bytes ⇒ +1,302%): Increased Key Size, however not as big as in ML-DSA case. Relevant impact can be expected both in Smart Meter secure storage and in Key Update procedures. The analysis is similar to the one performed for ML-DSA, but Falcon is in a much better position to face the expected challenges because of much smaller Key Sizes (1,312 bytes vs 897 bytes).

Signature Size (660 bytes ⇒ +931%): Increased Signature Size, however not as big as in ML-DSA case. As for the Falcon Public Key Sizes, the analysis for the Signature Size is comparable to the one performed for ML-DSA, but with Falcon having better perspectives because of its much smaller Signature Sizes (2,420 bytes vs 660 bytes).

Verification Time (86 µsec ⇒ -67%): Improved time when compared with classical cryptography, therefore no impact is expected.

Table 8
Falcon Pros & Cons

PROS	CONS
• The PQC Digital Signature algorithm with lower performance impact • Faster verification than classical Digital Signature algorithms	• No draft standard available yet • Algorithm may require modifications to address complex and insecure implementations • Implementation hardness in specific HW

SLH-DSA: Public Key Size (32 bytes ⇒ -50%): Improved size compared to classical cryptography, so no impact is expected.

Signature Size (16,976 bytes ⇒ +26,425%): Extreme increase in Signature Size. Major performance impact can be expected during Key Update procedures, drastically increasing the chances of failed or very long FW Update processes.

Verification Time (6,755 µsec ⇒ +2,486%): Extreme increase in Verification time, which could seriously affect the FW Update process itself. It is worth mentioning that the performance values (µsec) introduced in this document are taken for public reference sources not focusing on constrained devices, so that even worst absolute values can be expected when executed on real Smart Meters hardware.

Table 9
SLH-DSA Pros & Cons

PROS	CONS
• PQC Digital Signature algorithm with better security margins.	• The PQC Digital Signature algorithm with higher performance impact.

Finally, regarding crypto agility, beyond the sheer availability of new standardized PQC algorithms (or PQC algorithms modifications) and its secure implementations depending on possible future vulnerabilities, a relevant topic to be considered is the availability of these algorithms in the standards supporting Smart Grid communications. For instance, in DLMS/COSEM the allowed Crypto algorithms are summarized in Crypto Suites [32]:

Table 10
DLMS/COSEM Crypto Suites

Suite ID.	**Authenticated Encryption**	**Digital Signature**	***Key Agreement***	**HASH**	**Key Transport**
0	AES-GCM-128	-	-	-	AES-128 Key Wrap
1	AES-GCM-128	ECDSA P-256	ECDH P-256	SHA-256	AES-128 Key Wrap
2	AES-GCM-256	ECDSA P-384	ECDH P-384	SHA-384	AES-256 Key Wrap

Future versions are expected to introduce new Suites supporting PQC (or hybrid) algorithms, and more specifically, for the functional scenario considered in this analysis, introducing some new PQC Digital Signature supporting the FW Update process signature/verification. If the Digital Signature algorithm specified in this new Suite is subject of some attack and deemed insecure, some migration challenges are foreseen: Few available Suite IDs to be consumed if new suites are required (from 3 to 15), time modify the DLMS/COSEM standards if new algorithms are standardized, etc…

5. Conclusions

Falcon advantages over the other Digital Signature algorithms are on the performance side: lower key sizes and efficient signature verification. Falcon was chosen considering that in some environments with low computing capacity, or low bandwidth its properties can be desirables, and so the Smart Grid conditions seem the perfect fit for this algorithm, but its side channel and implementation drawbacks still need to be tackled in the NIST future standard specification.

ML-DSA is the primary recommendation from NIST as PQC Digital Signature Algorithm, showing equilibrate sizes/performance metrics together with strong theoretical security.

SLH-DSA was chosen by NIST because its security seems very solid and is based in a different underlying problem than those of ML-DSA and Falcon, but because of its size and performance issues seems the algorithm with the worst fit for the Smart Grid environment.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] Shor, P. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. SIAM Journal on Computing (1997, October). https://doi.org/10.1137/S0036144598347011

[2] Mosca, M. Cybersecurity in a Quantum World: will we be ready? (2015, April). https://csrc.nist.gov/csrc/media/events/workshop-on-cybersecurity-in-a-post-quantum-world/documents/presentations/session8-mosca-michele.pdf

[3] Mosca, M., & Piani, M. 2023 Quantum Threat Timeline Report (2023, December). https://globalriskinstitute.org/publication/2023-quantum-threat-timeline-report/

[4] NIST. Announcing Request for Nominations for Public-Key Post-Quantum Cryptographic Algorithms (2016, December). https://csrc.nist.gov/news/2016/public-key-post-quantum-cryptographic-algorithms

[5] Regev, O. On latices, learning with errors, random linear codes, and cryptography. Journal of the ACM (2009, September), Volume 56, Issue 6. https://doi.org/10.1145/1568318.1568324

[6] Wegger, V., Gassner, N., & Rosenthal, J. A Survey on Code-Based Cryptography (2024, July). https://doi.org/10.48550/arXiv.2201.07119

[7] Ding, J., & Yang, B.-Y. Multivariate Public Key Cryptography. Post-Quantum Cryptography (2009). https://doi.org/10.1007/978-3-540-88702-7_6

[8] Kipnis, A., Patarin, J., & Goubin, L. Unbalanced Oil and Vinegar Signature Schemes. Advances in Cryptology - EUROCRYPT '99 (1999, January).

[9] de Feo, L., Jao, D., & Plut, J. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. Journal of Mathematical Cryptology (2014, June). https://doi.org/10.1515/jmc-2012-0015

[10] Lamport, L. Constructing Digital Signatures from a One Way Function. Technical Report SRI-CSL-98, SRI International Computer Science Laboratory (1979, October). https://www.microsoft.com/en-us/research/uploads/prod/2016/12/Constructing-Digital-Signatures-from-a-One-Way-Function.pdf

[11] Hülsing, A. WOTS+ -- Shorter Signatures for Hash-Based Signature. AFRICACRYPT 2013. https://doi.org/10.1007/978-3-642-38553-7_10

[12] Merkle, R. A certified digital signature. Advances in Cryptology − CRYPTO'89 (1989). https://doi.org/10.1007/0-387-34805-0_21

[13] Moody, D. NIST IR 8413. Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process (2022, July). https://doi.org/10.6028/NIST.IR.8413-upd1

[14] Avanzi, R. CRYSTALS-Kyber. Algorithm Specifications And Supporting Documentation. Version 3.01 (2021, August). https://pq-crystals.org/kyber/data/kyber-specification-round3-20210804.pdf

[15] Aragon, N. BIKE. Bit Flipping Key Encapsulation v5.1 (2022, October). https://bikesuite.org/files/v5.0/BIKE_Spec.2022.10.10.1.pdf

[16] Bernstein, D. Classic McEliece: conservative code-based cryptography: cryptosystem specification (2022, October). https://classic.mceliece.org/mceliece-spec-20221023.pdf

[17] Aguilar, C. Hamming Quasi-Cyclic (HQC). Fourth round version (2023, April). https://pqc-hqc.org/doc/hqc-spec

[18] Jao, D. Supersingular Isogeny Key Encapsulation (2022, September). https://sike.org/files/SIDH-spec.pdf

[19] Bai, S. CRYSTALS-Dilithium. Algorithm Specifications and Supporting Documentation version 3.1 (2021, February). https://pq-crystals.org/dilithium/data/dilithium-specification-round3-20210208.pdf

[20] Fouque, P. Falcon: Fast-Fourier Lattice-based Compact Signatures over NTRU v1.2 (2022, October). https://falcon-sign.info/falcon.pdf

[21] Aumasson, J. SPHINCS+ v.3 (2020, October). https://sphincs.org/data/sphincs+-round3-specification.pdf

[22] Castryc, W., & Decru, T. An efficient key recovery attack on SIDH (2022, July). Retrieved from https://ia.cr/2022/975

[23] Ding, J., & Schmidt, D. Rainbow. Algorithm Specification and Documentation (2022, July). https://csrc.nist.gov/CSRC/media/Projects/post-quantum-cryptography/documents/round-3/submissions/Rainbow-Round3.zip

[24] Beullens, W. Breaking Rainbow Takes a Weekend on a Laptop (2022, February). https://ia.cr/2022/214

[25] NIST. Call for Additional Digital Signature Schemes for the Post-Quantum Cryptography Standardization Process (2022, October). https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/call-for-proposals-dig-sig-sept-2022.pdf

[26] Moody, D. NIST IR 8528. Status Report on the First Round of the Additional Digital Signature Schemes for the NIST Post-Quantum Cryptography Standardization Process (2024, October). https://doi.org/10.6028/NIST.IR.8528

[27] NIST FIPS 203. Module-Lattice-Based Key-Encapsulation Mechanism Standard (2024, August). https://doi.org/10.6028/NIST.FIPS.203

[28] NIST FIPS 204. Module-Lattice-Based Digital Signature Standard (2024, August). https://doi.org/10.6028/NIST.FIPS.204

[29] NIST FIPS 205. Stateless Hash-Based Digital Signature Standard (2024, August). https://doi.org/10.6028/NIST.FIPS.205

[30] Cooper, A. NIST SP 800-208. Recommendation for Stateful Hash-Based Signature Schemes (2020, October). https://doi.org/10.6028/NIST.SP.800-208

[31] Prest, T. FALCON Update 2024 (2024, April). https://csrc.nist.gov/Presentations/2024/falcon

[32] Kohout, D. Smart Metering Cybersecurity-Requirements, Methodology, and Testing. Sensors (Basel) (2023, April). https://pmc.ncbi.nlm.nih.gov/articles/PMC10146320/

[33] Chen, L. NIST SP 800-186. Recommendations for Discrete Logarithm-based Cryptography: Elliptic Curve Domain Parameters (2023, February). https://csrc.nist.gov/pubs/sp/800/186/final

[34] Moody, D. NIST IR 8547. Transition to Post-Quantum Cryptography Standards (2024, November). https://csrc.nist.gov/pubs/ir/8547/ipd

[35] sBACS. Measurements of public-key signature systems, indexed by machine. https://bench.cr.yp.to/results-sign.html