A Quantum-Resistant Advanced Metering Infrastructure

Vinícius Lagrota[1], Mateus de L. Filomeno[2], Leonardo de M. B. A. Dib[3], Julio López[4], Moisés V. Ribeiro[2]

[1] Research and Development Center for Communication Security (CEPESC)
Brasília, DF – Brazil.

[2]Electric Engineering Department
Federal University of Juiz de Fora (UFJF) – Juiz de Fora, MG – Brazil.

[3]WEG Electric Corp. – Jaraguá do Sul, SC – Brazil.

[4]Institute of Computing
Campinas State University (UNICAMP) – Campinas, SP – Brazil.

vinicius.lagrota@gmail.com, delimafilomeno@gmail.com,
ldib@weg.net, jlopez@ic.unicamp.br, moises.ribeiro@ufjf.br

Abstract. *This paper focuses on quantum-resistant cryptographic schemes to enhance the security of smart metering against threats from both quantum and classical computers. In this sense, we consider FrodoKEM and CRYSTALS-Kyber quantum-resistant as asymmetric cryptographic schemes and AES-GCM as a symmetric one. Moreover, we detail the design and implementation of a dedicated cryptographic module customized to ensure quantum-resistant advanced metering infrastructure (QR-AMI). Analysis based on experimental implementation shows the viability of a QR-AMI anchored on dedicated cryptographic modules implemented on a microcontroller (low-power processor) or a system-on-a-chip device.*

1. Introduction

Smart Grids (SGs) have materialized as an urgent solution to overcome the challenges in the electricity sector. These challenges encompass managing a plethora of loads, integrating renewable energy sources, and ensuring reliability, efficiency, sustainability, and flexibility, among others [Ghasempour and Lou 2017]. Yet, this influential paradigm only gains traction when stakeholders communicate effectively, thereby expediting the SG proliferation, facilitating a digital revolution in electric power systems globally [Ghasempour 2019], and unfortunately raising security and privacy concerns [Naderi and Asrari 2023].

In this scenario, Smart Meters (SMs) emerged as catalysts for large-scale SG deployments, having indisputable benefits. However, SMs generate and handle sensitive data of consumers, prosumers, and electric utilities. Consequently, a heightened awareness of security and privacy vulnerabilities in Advanced Metering Infrastructures (AMIs) has steered research and industrial endeavors toward protections against security infringements and unauthorized SM data access within AMI [Li et al. 2017]. Predominantly, these strategies rely on cryptography to safeguard against threats from classical computers in data networks, as evidenced in [Wang and Lu 2013]. However, the impending rise of

stable quantum computers threatens the current state of security, especially asymmetric cryptography, which is more vulnerable to quantum computational capabilities.

When it comes to asymmetric cryptography, its security is typically based on the complexity of integer factorization or the calculation of discrete logarithms, as in Rivest-Shamir-Adleman (RSA) and Elliptic Curve Cryptography (ECC) schemes [Wang and Lu 2013]. Unfortunately, these cryptographic schemes will be made insecure when cryptographic relevant quantum computers execute Shor's algorithm, a polynomial-time quantum algorithm for integer factorization and discrete logarithm operations [Shor 1994].

Integrating quantum-resistant asymmetric and symmetric cryptographic schemes is imperative to thoroughly neutralize the threats posed by quantum computers. In this context, studies centered on asymmetric post-quantum cryptography (PQC) adapted for SG and SM are paramount. In this domain, Cheng *et al.* [Cheng et al. 2019] unveiled a mutual authentication scheme between SMs and gateways, based on PQC. Similarly, Borges *et al.* [Borges et al. 2020] delivered a comparative analysis of key agreement protocols between PQC primitives and classical ones. Extending this, Ahn *et al.* [Ahn et al. 2022] assessed quantum-resistant key distribution frameworks, utilizing both Quantum Key Distribution (QKD) and PQC techniques, within Distributed Energy Resources (DER). More recently, the authors of [da Costa et al. 2022b] and [da Costa et al. 2022a] delved into the execution of the FrodoKEM and CRYSTALS-Kyber schemes for SMs. They established that a hardware/software co-design, built on System-on-a-Chip (SoC) devices, could feasibly incorporate PQC schemes within hardware-limited resources.

Despite recent efforts to address numerous concerns related to PQC for AMI, advancing the implementation of AMI based on quantum-resistant asymmetric and symmetric cryptographic schemes is imperative. These considerations are timely and critical, highlighting the merits of incorporating quantum security to counter threats from quantum and classical computers in AMI. Notably, as large-scale AMI deployments necessitate cost-effective solutions, discerning the suitable technologies that support PQC schemes within AMI is crucial.

To protect key agreements and sensitive data transactions between consumers/prosumers and electric utilities against potential quantum and classical computers, this paper delves into Quantum-Resistant AMI (QR-AMI) and the implementation nuances of the Quantum-Resistant Dedicated Cryptographic Module (QR-DCM), an integral and essential component of QR-AMI. It is important to mention that, regarding asymmetric cryptography, we focus exclusively on implementing PQC within our QR-AMI to address the emerging threat posed by quantum computing capabilities. While the most robust approach would involve hybrid cryptography [Barker et al. 2024] — combining PQC with classical cryptographic techniques like ECC to leverage their complementary strengths — we have chosen to limit our scope to PQC alone. This decision allows us to concentrate on the specific impact of quantum-resistant algorithms without the added complexity of hybrid solutions. Notably, ECC is significantly less memory-intensive than PQC. Thus, implementing a hybrid scheme would not substantially alter our hardware performance outcomes substantially, underscoring the feasibility of extending this work with hybrid techniques in future iterations. The main contributions of this paper encompass:

- A discussion on a QR-AMI harnessing either FrodoKEM or CRYSTALS-Kyber

for asymmetric cryptography and the Advanced Encryption Standard (AES)-256-Galois Counter Mode (GCM) for symmetric cryptography. This includes detailing the QR-DCM to enable quantum-resistant key agreements and the quantum-resistant encryption, decryption, and authentication of sensitive data transmitted through data networks.
- An implementation of the QR-DCM based on a low-power processor and high-power processor (i.e., software-driven), and SoC devices for hardware/software co-design.
- A comparative analysis that contrasts low-power and high-power processor-based implementations with the one on SoC devices, focusing on timing, hardware resource, and energy consumption.

The remainder of this paper is structured as follows: Section 2 formulates the problem under investigation. Section 3 details the QR-AMI based on QR-DCMs. Section 4 details cryptographic schemes chosen for a QR-DCM. Section 5 explores the implementation aspects of the QR-AMI, with a primary focus on QR-DCM. Performance comparisons are presented in Section 6. Section 7 states concluding remarks.

2. Problem Formulation

Smart metering enables monitoring and controlling energy usage of consumers/prosumers with more granularity and encompasses several functionalities [Bui et al. 2012]. These functionalities are met with data networks for exchanging sensitive information from consumers, prosumers, and utilities. In this context, AMI arises. It supports the acquisition of data by SMs, orchestrates communication among key stakeholders (namely consumers/prosumers and utility providers), and facilitates data storage and analytical processing within the Meter Data Management System (MDMS) [Mohassel et al. 2014].

Given the multi-user nature of AMI, the potential security attack surface is expansive, especially when data traverses potentially non-secure networks. A case in point is that even partial access to a subset of SMs data might provide an eavesdropper with insights that expose vulnerabilities in the critical electric power infrastructure. Furthermore, unauthorized data access at the individual SM level has the potential to reveal details about a consumer's behavioral patterns, including their daily routines and socioeconomic status. Recognizing these vulnerabilities, enhanced security protocols have been incorporated into AMI frameworks. These security measures bolster confidentiality and privacy and ensure the robustness of operations even when an attacker might leverage a classical supercomputer.

Figure 1 illustrates a novel and powerful threat scenario. At its core, a vulnerability in the data network becomes evident when an attacker, namely an eavesdropper, harnesses the power of a quantum computer. This capability allows unauthorized access to sensitive information exchanged between consumers or prosumers (represented by SMs) and electric utilities (denoted as MDMS). To effectively address this challenge, it is required to integrate specific cryptographic primitives into AMI. Specifically, quantum-resistant asymmetric and symmetric cryptographic schemes are vital to counter the potent threats posed by quantum computers.

No standard or regulation currently contemplates a PQC scheme in AMI. Consequently, a critical area of investigation centers on the viability of embedding quantum-

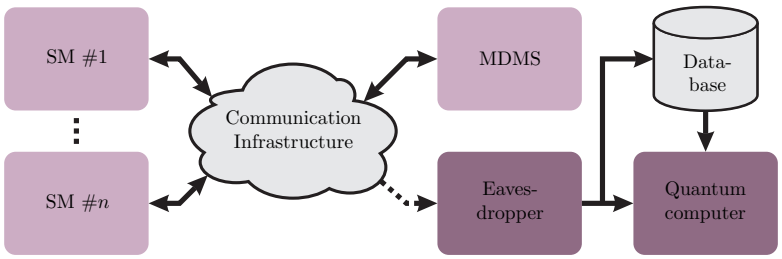

Figure 1. A typical security breach scenario in which an eavesdropper performs a sniffing attack and uses a quantum computer to decipher the SM data.

resistant asymmetric and symmetric cryptographic schemes within AMI—an architecture referred to as QR-AMI. Such integration is important for guaranteeing data security and protecting privacy in smart metering systems, especially in light of the emerging threat of quantum computational attacks. Critical research questions that deserve exploration in this context include: *What amount of hardware resources are requisite to facilitate the state-of-the-art PQC schemes within SMs and MDMS? Is it feasible to implement PQC schemes on processors with varying computational capacities, such as microcontrollers (low-power processor) and high-power processors for software-driven solutions or SoC devices designed for hardware/software co-design one?* The following sections elucidate these timely research questions.

3. A Quantum-Resistant AMI

A QR-AMI is designed to counter threats from adversaries with access to both quantum and classical computers. As such, QR-AMI must incorporate quantum-resistant asymmetric and symmetric cryptographic schemes. Figure 2 shows a block diagram illustrating the architecture of a QR-AMI. A key differentiation between QR-AMI and classical AMI is the cryptographic module employed: QR-AMI utilizes a QR-DCM that operates based on PQC schemes, whereas classical AMI incorporates a non-quantum-resistant Dedicated Cryptographic Module (DCM), which runs classical cryptographic schemes.

In a QR-AMI, any SM has access to PQC schemes. Consequently, a QR-DCM is required to interface with the SM to collect data and encrypt it using a quantum-resistant symmetric cryptographic scheme and send it through the AMI using data networks to MDMS. In this sense, QR-DCMs located on the consumer/prosumer side or on the electric utility side perform a key agreement through a Key Encapsulation Mechanism (KEM), generating a shared secret (i.e., a symmetric key), using a quantum-resistant asymmet-

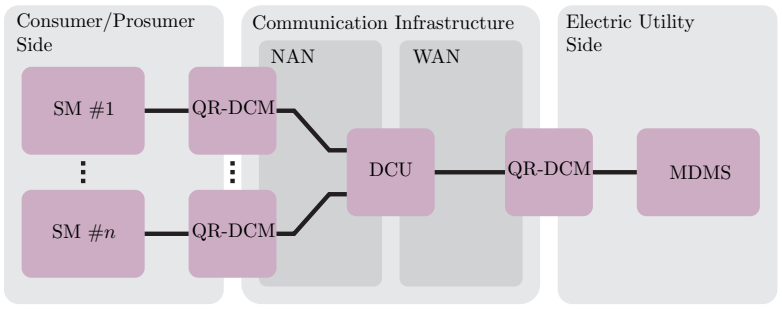

Figure 2. A block diagram of a QR-AMI, in which QR-DCM is the key element.

ric cryptographic scheme before encrypting and authenticating data. The QR-DCM at the consumer/prosumer side also has the capability of receiving encrypted data from the MDMS, which needs to be authenticated using Authenticated Encryption with Associated Data (AEAD), decrypted using the shared secret, and, finally, forwarded to the SM. In the model shown in Figure 2, there is one QR-DCM for each SM, although multiple SMs can be connected to one QR-DCM.

On the other hand, the QR-DCM on the electric utility side is implemented in the MDMS to send and receive encrypted and authenticated data. As the QR-DCM presented on the consumer or prosumer side, it checks the authenticity of received data through the AEAD before decrypting it using a quantum-resistant symmetric cryptographic scheme. Next, data correctly authenticated is forwarded to the MDMS. Naturally, a key agreement is previously executed using a quantum-resistant asymmetric cryptographic scheme. It is important to note that the QR-DCM on the electric utility side receives/sends a considerable amount of data depending on the size of the AMI (i.e., depending on how many SMs are presented on it). Therefore, the hardware resource of this QR-DCM must be carefully dimensioned for SMs and MDMS.

4. Cryptographic Schemes For a Quantum-Resistant DCM

QR-AMI uses asymmetric and symmetric cryptographic schemes to enable the secure key agreement between parties and the encryption/decryption of sensitive data. In this sense, Subsection 4.1 and Subsection 4.2 briefly describe the chosen quantum-resistant asymmetric and symmetric cryptography schemes, respectively.

4.1. Asymmetric Cryptographic Schemes

It is well-known that classical cryptographic schemes can perform a key agreement based on the hardness of factoring integers or computing discrete logarithms, such as RSA and ECC. Nonetheless, they are not quantum-resistant. With this concern in mind, the National Institute for Standards and Technology (NIST) started a PQC standardization process seeking to evaluate and select one or more quantum-resistant Public Key Encryption (PKE)/KEM and signature cryptographic schemes [Chen et al. 2016]. Among the submitted schemes, FrodoKEM [Alkim et al. 2020] and CRYSTALS-Kyber [Avanzi et al. 2021] schemes stood out from the others in the PKE/KEM category. The former is based on the Learning With Errors (LWE) problem proposed by Regev [Regev 2010], which is easily scalable, facilitating the adaptation of different security levels. Moreover, it is based on very conservative mathematical primitives due to the lack of any additional structure (i.e., ring structure), which provides, in theory, robust cryptographic principles despite degrading timing performance and requiring long keys and ciphertext. On the other hand, CRYSTALS-Kyber relies on the Module-Learning With Errors (M-LWE) problem, proposed by Langlois and Stehlé in [Langlois and Stehlé 2015], which is a generalization of the Ring-Learning With Errors (R-LWE) problem, proposed by Lyubashevsky in [Lyubashevsky et al. 2013]. Due to an additional structure, the R-LWE has opposite characteristics compared to the LWE. On the one hand, it considerably increases the performance and reduces the keys and ciphertext sizes, but on the other hand, it might be less resistant against malicious attacks.

M-LWE tries to get the best of the LWE and R-LWE. It has an additional structure but is less structured than R-LWE. According to recent cryptanalytic progress, a success-

ful attack against schemes relying on the M-LWE problem tends to be rarer than against schemes relying on the R-LWE [Bos et al. 2018]. It also performs similarly to the R-LWE and similar keys and ciphertext sizes. Finally, the M-LWE has better scalability than the R-LWE problem. Overall, we can say that the LWE problem is the most conservative, privileging consolidated principles, while the R-LWE is focused on performance. The M-LWE considers a trade-off between consolidated principles and performance.

FrodoKEM, relying on the LWE problem, remains as an alternate candidate in the NIST PQC standardization process in the PKE/KEM category and it has been still adopted by European authorities (e.g., German BSI and French ANSSI) due to its conservative approach. As requested in the NIST PQC standardization process, FrodoKEM is designed for Adaptive Chosen Ciphertext Attack (IND-CCA2) security at three levels: FrodoKEM-640, FrodoKEM-976, and FrodoKEM-1344, which aim to ensure a security level roughly equivalent to AES-128, AES-192, and AES-256 [Alkim et al. 2020], respectively. The CRYSTALS-Kyber, on the other hand, is the winner of the NIST PQC standardization process in the PKE/KEM category and has been standardized as ML-KEM in FIPS-203 [of Standards & Technology 2025]. It is also designed for IND-CCA2 security at three levels: CRYSTALS-Kyber-512, CRYSTALS-Kyber-976, and CRYSTALS-Kyber-1024 [Avanzi et al. 2021]. In this sense, FrodoKEM and CRYSTALS-Kyber schemes are considered for performing quantum-resistant KEM in QR-AMIs, and consequently, they are analyzed to integrate QR-DCM.

Table 1 summarizes the main characteristics of FrodoKEM and CRYSTALS-Kyber schemes. As mentioned, FrodoKEM relies on the LWE problem and the CRYSTALS-Kyber on the M-LWE problem. The type of the considered problem is reflected directly in the public key, private key, and ciphertext sizes, which are more than ten times larger in FrodoKEM than in CRYSTALS-Kyber. Considering the problem each scheme relies on, it can be said that the FrodoKEM scheme follows a conservative security approach, while the CRYSTALS-Kyber scheme is lightweight. In this sense, the CRYSTALS-Kyber scheme tends to consume fewer memory resources and can be significantly faster than the FrodoKEM scheme.

Moreover, Table 1 shows that the public key, private key, and ciphertext sizes of these PQC schemes are much larger than in classical cryptographic schemes. For instance, RSA usually employes keys with less than 4096-bits and ECC with less than 521-bits. Consequently, implementing PQC schemes in hardware- and memory-constrained devices is a sensitive issue; however, Tasopoulos *et al.* [Tasopoulos et al. 2023] showed that regarding performance, asymmetric quantum-resistant and classical cryptography demand almost the same energy consumption.

It is important to state that while this work adopts KEM for key agreement, it does not include any digital signature schemes, either classical or quantum-resistant. This design choice is deliberate and tied to our primary objective: Evaluating the feasibility of quantum-resistant KEMs and symmetric schemes within the hardware constraints of AMI devices. However, we recognize that the KEM alone does not authenticate the sender. In practice, digital signatures are essential to provide message origin authentication, particularly for control messages sent from utility providers to smart meters. This is a critical aspect of security in real-world deployments, as it prevents impersonation attacks and ensures non-repudiation. In future iterations of the QR-AMI, integrating post-quantum sig-

Table 1. Overview of FrodoKEM and CRYSTALS-Kyber schemes.

	FrodoKEM 640/976/1344	CRYSTALS-Kyber 512/768/1024
Class	Lattice	Lattice
Mathematical Problem	LWE	M-LWE
Public Key Size (bytes)	9616/15632/21520	800/1184/1568
Private Key Size (bytes)	19888/31296/43088	1632/2400/3168
Ciphertext Size (bytes)	9720/15744/21632	768/1088/1568
Approach	Conservative	Lightweight

nature schemes, such as ML-DSA, is crucial to achieving a complete quantum-resistant architecture.

4.2. Symmetric Cryptographic Schemes

A quantum computer will not significantly impact symmetric cryptography due to its mathematical principles. Consequently, all secure and standardized symmetric algorithms are potentially quantum-resistant. In addition, a suitable symmetric algorithm for AMIs must be reliable, fast, and lightweight. It is also desirable for this algorithm to provide AEAD. A suitable choice is, therefore, the AES, a standardized algorithm by the NIST in 2001 [Dworkin et al. 2001].

The AES is a block cipher of 128-bits, with three possible key sizes: 128, 192, and 256-bits. In this regard, the most conservative choice (i.e., 256-bit key) seems more appropriate in light of Grover's attack [Grassl et al. 2016]. Furthermore, the AES can be combined with the Galois/Counter Mode (AES-GCM), specified in NIST Special Publication $800 - 38$D [Dworkin 2007], for providing authenticated encryption (confidentiality and authentication). Therefore, the AES-256-GCM is considered and evaluated as the quantum-resistant symmetric cryptographic scheme for QR-DCMs. For the sake of simplicity, it will be called AES-GCM henceforth.

5. Experimental Implementation of QR-AMI

Upgrading the processing capacity of a MDMS is generally straightforward for electric utilities. In contrast, adapting existing SMs poses a significant challenge due to their limited processing capabilities. Hence, a fair implementation of QR-DCM becomes critical for ensuring quantum security in QR-AMI. Three primary approaches for implementing QR-DCM are:

- **Approach** #1: It integrates the QR-DCM into an existing internal processor of SMs. Due to the limited computational capacity of current SMs, this approach

requires a comprehensive hardware redesign, including replacing the existing processor. This approach is best suited for SMs in the design phase.

- **Approach** $\#2$: It incorporates a specialized and auxiliary processor into the SM, exclusively for the execution of the QR-DCM. While this also requires significant hardware modifications, executing the cryptographic schemes on a specialized processor confers an additional layer of security. It is also advantageous for designing new SMs.
- **Approach** $\#3$: It employs a dedicated external module with its own processor to implement the QR-DCM. This module can easily interface with existing ports on SMs, thus providing an efficient avenue for implementing quantum-resistant cryptographic schemes in existing SMs. It is most appropriate for SMs already deployed in the field.

Although SMs can be significantly modified, such changes would require a rigorous re-certification process to ensure compliance with industry standards and national regulations, a task that is both time-consuming and costly. Considering the vast number of SMs already in deployment, **Approach** $\#3$ emerges as the most viable approach for fortifying existing AMI systems against both classical and quantum computational threats. The downside of this approach lies in the potential vulnerability of the serial communication between the dedicated external module and the SM, which can be mitigated by physically integrating the QR-DCM within the SM. However, in line with the Zero Trust principle, physical security alone is insufficient. A detailed discussion of this issue is beyond the scope of this paper.

By leveraging this approach, the QR-DCM offloads the cryptographic workload to an external module, avoiding extensive redesign of resource-constrained SMs. It should be highlighted that QR-DCM may be implemented on either a microcontroller (low-cost processor) or an SoC, each with distinct trade-offs. A microcontroller-based solution offers cost-efficiency at the expense of longer execution times, while an SoC-based solution, although more expensive, delivers hardware acceleration for compute-intensive tasks, significantly enhancing execution speed.

Concerning **Approach** $\#3$, Figure 3 shows the experimental implementation employed to simulate a QR-AMI utilizing QR-DCMs implemented on both specialized microcontroller and SoC development boards. Subsequent subsections will elaborate on the various aspects of this implementation.

5.1. Smart Meter

The SMs, tagged as $\#1A$ and $\#1$B in Figure 3, are SMW3000 models from WEG [WEG 2022], an ANSI bottom-connected three-phase meter for direct measurement up to 120A. This model supports the Device Language Message Specification (DLMS)/Companion Specification for Energy Metering (COSEM) communication protocol via various interfaces (e.g., local port, Ethernet, optical, and serial), allowing integration with multiple technologies. In this implementation, the SM communicates with the QR-DCM via serial DLMS/COSEM.

5.2. Quantum-Resistant DCM

The QR-DCMs, identified as $\#2A$, $\#2B$, and $\#2C$ in Figure 3, serve as critical elements for achieving quantum security in QR-AMI. These modules incorporate quantum-

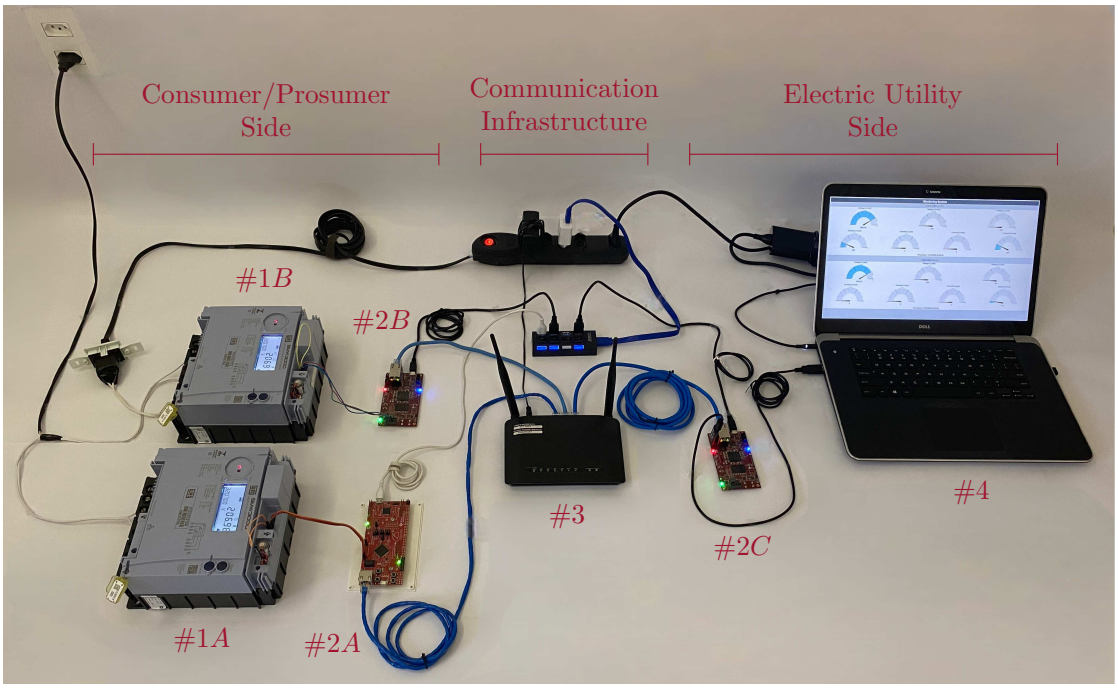

Figure 3. An experimental implementation of a QR-AMI.

resistant cryptographic schemes and the DLMS/COSEM communication protocol. The choice of processing power for the QR-DCM largely depends on the scale of the AMI. Two types of technologies are considered for implementing the QR-DCM, namely, microcontroller-based and SoC-based platforms:

- **Microcontroller** ($\#2A$): It relies on a device with low power consumption and, consequently, a not-so-fast processing unit for performing its tasks. It fits well for low processing power (e.g., processing a small amount of data). For instance, on the consumer/prosumer side, connected to only one or a few SMs, or even on the electric utility side when an AMI is dedicated to a small number of SMs. It uses the EK-TM4C129EXL kit [Instruments 2014], based on a 120 MHz Advanced RISC Machine (ARM) Cortex-M4F with 1024 kB flash, 256 kB SRAM, and 6 kB EEPROM. This platform supports only software implementations, resulting in longer execution times. Source code is available at [Lagrota 2024d].
- **SoC device** ($\#2B$ and $\#2C$): Recommended for high processing needs (e.g., many SMs), on either side of the AMI. It employs the MicroZed 7010 board [Xilinx 2018], which includes a Xilinx Zynq-7000 SoC featuring 1 GB DDR3 SDRAM, 128 Mb QSPI Flash, and an XC7Z010-1CLG400C chip. This chip integrates a Processing System (PS) (ARM Cortex-A9) and a Programmable Logic (PL) (Field Programmable Gate Array (FPGA)) with 28k logic cells, 17.6k Lookup Tables (LUTs), 35.2k registers, 60 Block Random Access Memory (BRAM), and 80 Digital Signal Processing (DSP) blocks. A 33.33 MHz oscillator feeds clocks to the PS and PL via a Phase-Locked Loop (PLL). This bsoard enables both software and hardware/software co-design implementations, with cryptographic routines accelerated in hardware. Implementations are available at [Lagrota 2024a, Lagrota 2024c, Lagrota 2024b].

Regardless of the platform, all QR-DCMs communicate with SMs via serial DLMS/COSEM and with the MDMS using JSON-formatted serial communication. Ethernet is used to interface with the Data Concentrator Unit (DCU), ensuring interoperability across different AMI components.

5.3. Communication Infrastructure

The communication infrastructure only comprises a commercial D-Link DSL-2740E router, tagged as $\#3$ in Figure 3, which forwards packets from one node to another (i.e., allows two-way communication) using the Ethernet network. Note that all the QR-DCMs are connected in the same data network, so they can easily communicate. As we aim to focus on the implementation aspects of the quantum-resistant functionality in a QR-AMI, the type of communication infrastructure is not a relevant issue; therefore, no further efforts are necessary to implement the communication infrastructure.

5.4. MDMS

The MDMS, tagged as $\#4$ in Figure 3, is emulated on a Python application with Flask, a micro-framework suitable for small applications with simple requirements, communicating with the QR-DCM. Mainly, it receives decrypted and authenticated data from the QR-DCMs, which contain information collected by SMs, and presents it on a screen using Flask. Similar to the communication infrastructure, no further efforts were spent trying to emulate an MDMS because it would not bring any relevant contribution to the scope of this work.

6. Experimental Results

This section analyzes the experimental implementation of a QR-AMI, focusing on the QR-DCM as the key enabler for quantum security in the AMI. Quantum-resistant asymmetric (FrodoKEM and CRYSTALS-Kyber) and symmetric (AES-GCM) cryptographic schemes are implemented on two different platforms as described in Subsection 5.2, using the following implementations:

- Impl. $\#1$: A software-only implementation executed on the ARM Cortex-M4F in the EK-TM4C129EXL.
- Impl. $\#2$: A software-only implementation using the ARM Cortex-A9 on the MicroZed 7010 board. The ARM Cortex-A9 is a high-power processor.
- Impl. $\#3$: A hardware/software co-design implementation leveraging both the ARM Cortex-A9 and the FPGA on the MicroZed 7010 board (i.e., SoC device).

Impl. $\#1$ and $\#2$ aim to contrast the performance capabilities between low- and high-power processing units that rely solely on a microcontroller (ARM Cortex-M4F) and a more potent processor (ARM Cortex-A9), respectively. Conversely, Impl. $\#3$ aims to quantify the benefits of hardware acceleration. Notice that hardware acceleration for symmetric cryptography is not considered in this analysis, as its impact is deemed negligible, which is substantiated in the subsequent discussions.

To ensure a fair comparison, the source codes for all the implementations are adapted from those submitted to the NIST PQC standardization process [Chen et al. 2016]. Specifically, regarding FrodoKEM, only the FrodoKEM-640 version is considered for hardware acceleration due to the excessive resource consumption, particularly

BRAMs, of its more secure variants (i.e., FrodoKEM-976 and FrodoKEM-1344) [da Costa et al. 2022b]. For CRYSTALS-Kyber, we focus on CRYSTALS-Kyber-512 to make a fair security-level comparison with FrodoKEM.

Moreover, asymmetric and symmetric cryptographic scheme implementations regarding resource usage, timing performance, and energy consumption are detailed in Subsections 6.1, 6.2, and 6.3, respectively. Lastly, Subsection 6.4 summarizes this section with a few comments.

6.1. Resource Usage

This subsection analyzes the resource usage of Impl. #1, #2, and #3. In this sense, Subsection 6.1.1 focuses on memory resources and consequently pays attention to the software aspects, whereas Subsection 6.1.2 addresses the hardware resources from FPGA, emphasizing the hardware/software co-design.

6.1.1. Memory Resources

This analysis assesses the memory resource usage for the software components of Impl. #1 and #2, as well as the software aspect of Impl. #3. Three primary memory segments are scrutinized for each configuration: (i) the text segment, addressing the executable instructions; (ii) the read-only data segment, storing immutable constants such as global constants; and (iii) the data segment, containing initialized static variables, which are mutable at runtime. The text and read-only data segments usually reside in flash memory because they are unchangeable, whereas the editable data segment is stored in RAM.

Figure 4 illustrates the memory consumption across these three segments. It is evident that the ARM Cortex-M4F-based Impl. #1 utilizes fewer memory resources compared to the ARM Cortex-A9-based Impl. #2. Although the source codes across these implementations are functionally equivalent, the compiled binaries exhibit discrepancies in resource utilization. Platforms with higher computational power, like the ARM Cortex-A9, inherently entail more complexity, particularly in background management processes, thus leading to elevated memory consumption. These platforms are, however, endowed with more abundant resources to accommodate this added complexity, as elaborated in Section 5.2.

A direct comparison between Impl. #2 and #3, both based on the ARM Cortex-A9, reveals that the hardware/software co-design approach (Impl. #3) is more memory-efficient. This efficiency stems from the fact that functions executed in hardware do not require compilation, thereby conserving memory. Nonetheless, these savings are marginal, attributed to the partial hardware acceleration of certain functions, as elaborated in [da Costa et al. 2022b, da Costa et al. 2022a].

Interestingly, within the same implementation environment, FrodoKEM is found to be more memory-efficient than CRYSTALS-Kyber in Impl. #2 and #3. This trend is inverted in Impl. #1, where CRYSTALS-Kyber demonstrates greater memory efficiency, except in the read-only data segments, where the sizes are nearly identical. Despite the similar source codes for both cryptographic schemes, the observed discrepancy is due to differences in the compiler technologies employed. As such, the memory utilization characteristics under one compiler configuration may not directly extend to another.

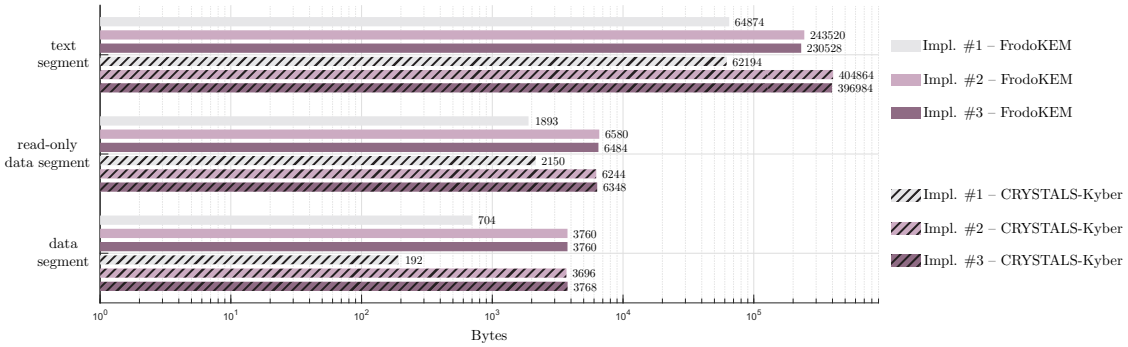

Figure 4. Memory usage for text, read-only data, and data segments in bytes of the FrodoKEM and CRYSTALS-Kyber schemes for Impl. $\#1$, $\#2$, and $\#3$.

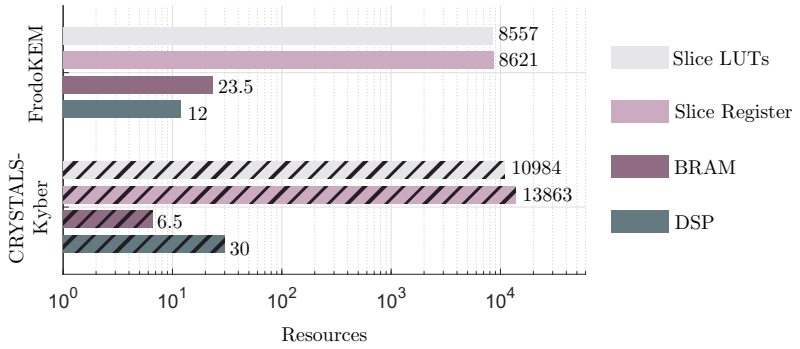

Figure 5. Hardware resource usage of the FrodoKEM and CRISTALS-Kyber schemes in terms of Slice LUTs, Slice Register, BRAM, and DSP blocks.

6.1.2. Hardware resources

This analysis compares the hardware resource usage of the FrodoKEM and CRYSTALS-Kyber schemes, and therefore, it considers only Impl. $\#3$. Four main resources are evaluated: (i) slice LUTs, which are small asynchronous RAMs manipulated to implement combinational logic; (ii) slice registers, based on Flip Flop (FF) and used to hold states; (iii) BRAM, based on synchronous RAM and used to store large volumes of data; and (iv) DSP blocks, which are dedicated blocks for performing multiplications, increasing performance, and reducing logic use (e.g., slice LUTs and registers).

Figure 5 shows the hardware resource usage of the FrodoKEM and CRISTALS-Kyber. It reveals that CRYSTALS-Kyber consumes more slice LUTs, slice registers, and DSP blocks than FrodoKEM, owing to its more complex hardware functions [da Costa et al. 2022b, da Costa et al. 2022a]. Conversely, FrodoKEM uses more BRAM due to its longer key lengths, as detailed in Table 1.

6.2. Timing Analysis

This subsection presents a timing analysis. In this regard, Subsection 6.2.1 compares the execution time of the asymmetric cryptographic schemes, while Subsection 6.2.2 presents the timing analysis of the symmetric cryptography.

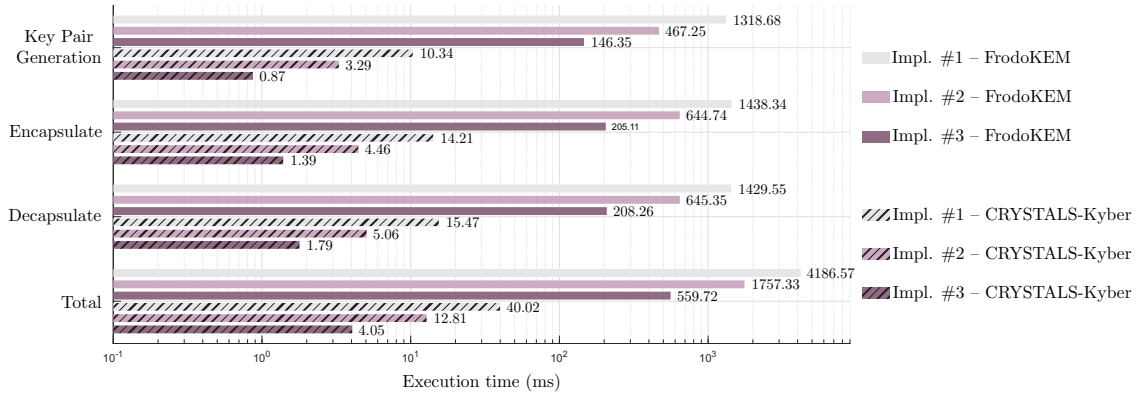

Figure 6. Timing performance in ms of the FrodoKEM and CRYSTALS-Kyber schemes using Impl. $\#1$, $\#2$, and $\#3$.

6.2.1. Asymmetric cryptography

It provides an in-depth evaluation of the execution times associated with key pair generation, encapsulation, and decapsulation for the FrodoKEM and CRYSTALS-Kyber schemes, as implemented using Impl. $\#1$, $\#2$, and $\#3$.

Figure 6 shows the total execution times for the aforementioned cryptographic schemes. The performance disparity between the FrodoKEM and CRYSTALS-Kyber is significant. Regardless of the implementation method, CRYSTALS-Kyber consistently exhibits a performance that is at least a hundredfold superior to that of FrodoKEM. These empirical results substantiate the theoretical comparisons presented in Subsection 4.1.

Concerning FrodoKEM, the hardware-accelerated Impl. $\#3$ significantly outperforms its solely software-based counterparts. Specifically, it is around 7.5 and 3.1 times faster than Impl. $\#1$ and $\#2$, respectively. This substantially reduces total execution time, decreasing from 4186.57 ms and 1757.33 ms to a 559.72 ms when hardware acceleration is leveraged.

Similarly, for the CRYSTALS-Kyber scheme, Impl $\#3$ shows superior performance, being approximately 9.9 and 3.2 times faster than Impl. $\#1$ and $\#2$, respectively. Consequently, the total execution time for Impl. $\#1$ and $\#2$ is reduced from 40.02 ms and 12.81 ms to an impressive 4.05 ms when utilizing hardware acceleration, as in Impl. $\#3$.

6.2.2. Symmetric cryptography

In symmetric cryptography, the execution time needed for data encryption, decryption, and data authentication is closely tied to the data length in the AES-GCM scheme. For instance, to encrypt, decrypt, and authenticate a 50-byte data packet, Impl. $\#1$ and $\#2$ require 0.262 ms and 0.062 ms, respectively. These results show that the execution time for symmetric cryptography is negligible compared to asymmetric cryptography. Hence, implementing hardware acceleration for AES-GCM would not yield substantial time-saving benefits for small data packets, as typically encountered in AMIs. On the other hand, such hardware acceleration would consume additional hardware resources. Consequently, symmetric cryptography in this experiment is confined to a software implementation.

Table 2. Energy consumption between FrodoKEM and CRYSTALS-Kyber

Scheme	Impl.	Time (ms)	Power (W)	Energy (mJ)
FrodoKEM	Impl. #1	4186.57	0.35	1454.79
	Impl. #2	1757.33	1.65	2903.11
	Impl. #3	559.72	2.05	1147.43
CRYSTALS-Kyber	Impl. #1	40.02	0.35	13.91
	Impl. #2	12.81	1.65	21.16
	Impl. #3	4.05	1.83	7.43

6.3. Energy Consumption

This subsection analyzes the energy consumption patterns. It employs timing estimates delineated in Subsection 6.2.1 and the energy consumption for key pair generation, encapsulation, and decapsulation. Power consumption estimate for Impl. $\#1$ is extracted from the EK-TM4C129EXL datasheet [Instruments 2014]. For Impl. $\#2$ and $\#3$, power consumption estimates are furnished by the synthesizer of the SoC device. Note that these estimates include activating the communication layer, which enables data exchange between QR-DCMs and contributes to significant power consumption.

Table 2 lists the energy consumption of FrodoKEM and CRYSTALS-Kyber. First, as FrodoKEM is considerably slower than CRYSTALS-Kyber while having a similar power estimate for each implementation, the former consumes much more energy. Second, Impl. $\#1$ consumes much less power than Impl. $\#2$ and $\#3$ which reflects in its performance. However, no significant difference exists between Impl. $\#2$ and $\#3$, indicating that the hardware acceleration is energy efficient. Lastly, we see that the Impl. $\#3$ is the most energy efficient, followed by Impl. $\#1$. Overall, the most sophisticated hardware (i.e., SoC device) attains the lowest energy consumption, indicating its feasibility for QR-AMIs.

6.4. General Comments

In summary, the choice of cryptographic scheme and hardware platform for QR-DCM should be based on the specific requirements of the QR-AMI. Regarding cryptographic schemes, FrodoKEM, despite its higher computational requirements, is preferable when robust cryptographic principles based on well-established mathematical foundations are important, as discussed in Section 4.1. However, its slower execution times could be a bottleneck in large-scale QR-AMI deployment, potentially failing to meet timing constraints. On the other hand, CRYSTALS-Kyber offers a more efficient alternative for most scenarios. As for symmetric cryptography, AES-GCM appears universally suitable, being lightweight and efficient enough to be implemented on any platform, as noted in Subsection 4.2.

In the context of hardware choices for QR-DCM, the scenarios best suited for microcontroller-based hardware (i.e., Impl. $\#1$ and $\#2$) and SoC devices (i.e., Impl. $\#3$) differ significantly. If the computational demands are low, such as in cases where the QR-DCM is interfacing with only a single SM, then a microcontroller-based implemen-

tation (Impl. $\#1$) would be more cost-effective. For moderate to high computational demands, such as when multiple SMs are connected, an ARM Cortex-A9-based implementation (Impl. $\#2$) would be appropriate. For scenarios demanding extremely high computational capabilities, Impl. $\#3$ would be the logical choice.

From the perspective of electric utilities, the scalability of the QR-AMI should be a key factor in deciding the appropriate implementation. Small-scale QR-AMIs may manage with Impl. $\#1$, but as the size and complexity grow, transitioning to Impl. $\#2$ or even Impl. $\#3$ would become increasingly necessary to meet timing, computational requirements, and energy consumption.

7. Conclusions

This paper has discussed QR-AMI for securing sensitive data against attacks from quantum and classical computers. The QR-AMI mainly relies on QR-DCMs, which implement an asymmetric quantum-resistant cryptographic scheme, one with more robust cryptographic principles (i.e., FrodoKEM) or another with less computational burden (i.e., CRYSTALS-Kyber). In addition, QR-DCMs implement a symmetric cryptographic scheme that efficiently encrypts, decrypts, and authenticates data (i.e., AES-256-GCM). It also has shown that QR-DCM in a microcontroller suits well in SMs with less processing power. On the other hand, SoC device is necessary for MDMSs that exchange a large volume of data. Overall, QR-DCMs based on external and dedicated hardware are suitable for making the existing SMs quantum-resistant.

References

[Ahn et al. 2022] Ahn, J. et al. (2022). Toward quantum secured distributed energy resources: Adoption of post-quantum cryptography (PQC) and quantum key distribution (QKD). *Energies*, 15(3). pages 2

[Alkim et al. 2020] Alkim, E. et al. (2020). FrodoKEM: Learning with errors key encapsulation. Technical report, NIST, Gaithersburg, MD. pages 5, 6

[Avanzi et al. 2021] Avanzi, R. et al. (2021). CRYSTALS-Kyber algorithm specifications and supporting documentation. Technical report, NIST. pages 5, 6

[Barker et al. 2024] Barker, W., Polk, W., and Souppaya, M. (2024). NIST IR 8547 — transition to post-quantum cryptography standards. Technical report, NIST. pages 2

[Borges et al. 2020] Borges, F., Reis, P. R., and Pereira, D. (2020). A comparison of security and its performance for key agreements in post-quantum cryptography. *IEEE Access*, 8:142413–142422. pages 2

[Bos et al. 2018] Bos, J. et al. (2018). CRYSTALS-Kyber: a CCA-secure module-lattice-based KEM. In *Proc. IEEE Eur. Symp. Security and Privacy*, pages 353–367. pages 6

[Bui et al. 2012] Bui, N., Castellani, A. P., Casari, P., and Zorzi, M. (2012). The internet of energy: A web-enabled smart grid system. *IEEE Netw.*, 26(4):39–45. pages 3

[Chen et al. 2016] Chen, L. et al. (2016). Report on post-quantum cryptography. Technical Report 8105, NIST, Gaithersburg, MD. pages 5, 10

[Cheng et al. 2019] Cheng, C., Qin, Y., Lu, R., Jiang, T., and Takagi, T. (2019). Batten down the hatches: Securing neighborhood area networks of smart grid in the quantum era. *IEEE Trans. Smart Grid*, 10(6):6386–6395. pages 2

[da Costa et al. 2022a] da Costa, V. L. R., Camponogara, Â., López, J., and Ribeiro, M. V. (2022a). The feasibility of the CRYSTALS-Kyber scheme for smart metering systems. *IEEE Access*, 10:131303–131317. pages 2, 11, 12

[da Costa et al. 2022b] da Costa, V. L. R., López, J., and Ribeiro, M. V. (2022b). A system-on-a-chip implementation of a post-quantum cryptography scheme for smart meter data communications. *Sensors*, 22(19). pages 2, 11, 12

[Dworkin 2007] Dworkin, M. J. (2007). *Sp 800-38d. recommendation for block cipher modes of operation: Galois/counter mode (GCM) and GMAC*. National Institute of Standards & Technology. pages 7

[Dworkin et al. 2001] Dworkin, M. J., Barker, E., Nechvatal, J., Foti, J., Bassham, L., Roback, E., and Dray, J. (2001). Advanced encryption standard (AES). pages 7

[Ghasempour 2019] Ghasempour, A. (2019). Internet of things in smart grid: Architecture, applications, services, key technologies, and challenges. *Inventions*, 4(1):22. pages 1

[Ghasempour and Lou 2017] Ghasempour, A. and Lou, J. (2017). Advanced metering infrastructure in smart grid: Requirements, challenges, architectures, technologies, and optimizations. In *Smart Grids: Emerging Technologies, Challenges and Future Directions*. Nova Science Publishers. pages 1

[Grassl et al. 2016] Grassl, M., Langenberg, B., Roetteler, M., and Steinwandt, R. (2016). Applying Grover's algorithm to AES: Quantum resource estimates. In Takagi, T., editor, *Post-Quantum Cryptography*, pages 29–43, Cham. Springer International Publishing. pages 7

[Instruments 2014] Instruments, T. (2014). Tiva™ TM4C129ENCPDT Microcontroller. pages 9, 14

[Lagrota 2024a] Lagrota, V. (2024a). Frodo FPGA. `https://gitlab.com/vinicius.lagrota/frodo-fpga`. pages 9

[Lagrota 2024b] Lagrota, V. (2024b). IP-repository. `https://gitlab.com/vinicius.lagrota/ip-repository`. pages 9

[Lagrota 2024c] Lagrota, V. (2024c). Kyber FPGA. `https://gitlab.com/vinicius.lagrota/kyber-fpga`. pages 9

[Lagrota 2024d] Lagrota, V. (2024d). PQC TM4C129 - ARM M4F. `https://gitlab.com/vinicius.lagrota/pqc-tm4c129`. pages 9

[Langlois and Stehlé 2015] Langlois, A. and Stehlé, D. (2015). Worst-case to average-case reductions for module lattices. *Designs, Codes and Cryptography*, 75(3):565–599. pages 5

[Li et al. 2017] Li, S., Xue, K., Yang, Q., and Hong, P. (2017). PPMA: Privacy-preserving multisubset data aggregation in smart grid. *IEEE Trans. Ind. Inform.*, 14(2):462–471. pages 1

[Lyubashevsky et al. 2013] Lyubashevsky, V., Peikert, C., and Regev, O. (2013). On ideal lattices and learning with errors over rings. *J. ACM*, 60(6). pages 5

[Mohassel et al. 2014] Mohassel, R. R., Fung, A., Mohammadi, F., and Raahemifar, K. (2014). A survey on advanced metering infrastructure. *Int. J. Electr. Power Energy Syst.*, 63:473–484. pages 3

[Naderi and Asrari 2023] Naderi, E. and Asrari, A. (2023). A remedial action scheme to mitigate market power caused by cyberattacks targeting a smart distribution system. *IEEE Trans. Ind. Inform.*, pages 1–12. early access, doi: 10.1109/TII.2023.3304049. pages 1

[of Standards & Technology 2025] of Standards & Technology, N. I. (2025). *Module-Lattice-Based Key-Encapsulation Mechanism Standard*. National Institute of Standards & Technology. pages 6

[Regev 2010] Regev, O. (2010). The learning with errors problem (invited survey). In *Proc. IEEE Annu. Conf. on Computational Complexity*, pages 191–204. pages 5

[Shor 1994] Shor, P. W. (1994). Algorithms for quantum computation: Discrete logarithms and factoring. In *Proc. 35th Annu. Symp. Foundations of Computer Science*, pages 124–134. pages 2

[Tasopoulos et al. 2023] Tasopoulos, G., Dimopoulos, C., Fournaris, A. P., Zhao, R. K., Sakzad, A., and Steinfeld, R. (2023). Energy consumption evaluation of post-quantum TLS 1.3 for resource-constrained embedded devices. Cryptology ePrint Archive, Paper 2023/506. `https://eprint.iacr.org/2023/506`. pages 6

[Wang and Lu 2013] Wang, W. and Lu, Z. (2013). Cyber security in the smart grid: Survey and challenges. *Comput. Netw.*, 57(5):1344 – 1371. pages 1, 2

[WEG 2022] WEG (2022). Intelligent power meters – SMW series. pages 8

[Xilinx 2018] Xilinx (2018). Zynq-7000 SoC Data Sheet: Overview. pages 9