Performing Classical and Post-Quantum Cryptography on IoT Data: An Evaluation

Darshana Rawal [1], Jan Seedorf [1], Nikhil Jha[1]
1 Hochschule für Technik Stuttgart, Schellingstr 24,70174 Stuttgart
(darshana.rawal, jan.seedorf)@hft-stuttgart.de,nikhiljha122@gmail.com

Keywords: Post Quantum, Cryptography, Internet of Things(IOT), Sensor Community.

Abstract

The increasing number of Internet of Things (IoT) devices requires the implementation of reliable and efficient cryptographic solutions to ensure the security of data transmission and storage. Traditional cryptographic algorithms can be challenging to implement without compromising performance or security, particularly due to the significant resource constraints faced by many IoT devices. This paper aims to evaluate the performance of classical and post-quantum cryptographic libraries on IoT hardware, specifically using the PyCryptodome Python library for traditional cryptography and the Kyber 512 algorithm for post-quantum encryption. The study assesses the security features, computational efficiency, and feasibility of these libraries in resource-constrained environments. Experiments were conducted on the Raspberry Pi 5 platform to analyze encryption and decryption performance, providing insights into the trade-offs between security and processing overhead. The findings will help identify the best cryptographic solutions for IoT devices based on their performance, security, and hardware limitations. This study includes a practical implementation of the Kyber 512 algorithm on the Raspberry Pi platform, exploring the feasibility of deploying quantum-resistant cryptography in IoT environments using Python. The research examines cryptographic performance in constrained environments by measuring encryption and decryption times scientifically. The results offer valuable insights into the computational overhead and feasibility of implementing post-quantum cryptographic solutions on limited-resource hardware. Scientific graphs were created to illustrate the complexity of encryption and decryption times, providing a visual representation of performance variations in various scenarios. Understanding these graphs clarifies the trade-offs between security and computational efficiency, which are crucial for real-world IoT applications. Furthermore, the study discusses potential optimizations and future directions that could enhance the efficiency of cryptographic operations in IoT hardware. In summary, this paper contributes to the study of cryptographic applications for IoT by investigating the viability of post-quantum cryptographic algorithms in constrained environments. By considering the findings, IoT security can be strengthened against emerging cyber threats through the selection of optimized cryptographic libraries that balance security, performance, and hardware limitations.

1. Introduction

1.1 Background

The increasing popularity of Internet of Things (IoT) devices across various industries, including healthcare and smart cities, highlights the urgent need to protect data and communications. As IoT continues to grow rapidly, the attack surface has expanded, allowing potential vulnerabilities through encryption errors, weak authentication methods, insecure communication protocols, and inadequate key management practices. Choosing the right cryptography library for IoT devices, which often have resource constraints, is crucial. When employing traditional encryption methods—such as the Advanced Encryption Standard (AES) for symmetric encryption and Rivest-Shamir-Adleman (RSA) for asymmetric encryption—it is important to carefully assess both efficacy and security. Additionally, algorithms like Kyber512 and other post-quantum solutions are gaining traction due to their potential resistance to quantum attacks. An optimized cryptographic library is essential for selecting the appropriate algorithm. Although open-source cryptographic libraries, like PyCryptodome for traditional encryption or specific implementations of Kyber512 for post-quantum security, offer transparency, adaptability, and affordability, their security and performance can differ significantly between systems and use cases. This paper evaluates various libraries for IoT scenarios to ensure both security and performance(Ding, J. 2006, NIST,2022, W. S. P. "Post-Quantum Cryptography Standardization,NIST(2022).

The rise of quantum computers may render standard encryption techniques obsolete, making it essential to explore post-quantum cryptography algorithms (Bernstein, D. J., Lange, T., et al. ,2017; Ducas, L., et al., 2022). It is important to adapt IoT devices, which are already limited in energy and processing capabilities, to address these new security threats. Evaluating cryptographic libraries is a way to prepare IoT ecosystems for future advancements in computing technology while ensuring that the cryptographic libraries are effective for current needs.

A careful balance between computing efficiency and security robustness is necessary for the diverse range of IoT applications. Security requirements and limitations vary for each device used in smart homes, healthcare monitoring, and industrial automation. By systematically examining cryptographic libraries and their effectiveness on IoT hardware, this research establishes a foundation for informed decisions regarding the security of IoT networks (T. Boneh 2020)

Another critical factor is the scalability of cryptographic systems in large IoT installations. Many IoT ecosystems contain thousands of connected devices that must communicate and exchange data safely in real-time. Effective cryptographic systems need to ensure low latency while maintaining strong encryption to prevent security from becoming a barrier to IoT operations. This paper examines

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLVIII-4-W16-2025-91-2025 | © Author(s) 2025. CC BY 4.0 License.

how different cryptographic libraries address scalability challenges in constrained situations.

As the legal framework for IoT security rapidly expands, governments and industry bodies are enforcing stricter compliance standards. Businesses implementing IoT solutions must adhere to regulations such as the General Data Protection Regulation (GDPR) and other international standards.

1.2 Problem Statement

The rapid growth of the Internet of Things (IoT) has significantly increased the demand for secure communication and data protection. IoT devices often have limited resources, making it essential to use efficient cryptographic solutions to ensure data integrity and confidentiality. The long-term security of Rivest-Shamir-Adleman (RSA), a widely used classical cryptographic method, is at risk due to the advancement of quantum computing. In contrast, the Advanced Encryption Standard (AES), especially AES-256, is resilient against quantum attacks and is considered quantum-safe. There is an urgent need for post-quantum cryptography (PQC) algorithms that are both resistant to quantum threats and practical for resource-constrained IoT devices ("Advanced Encryption Standard (AES) - FIPS 197," 2001).

Implementing encryption and decryption techniques in IoT environments can be done easily and cost-effectively using open-source cryptographic libraries. However, the performance of these libraries can be significantly influenced by the hardware capabilities and the cryptographic algorithms employed. This study aims to thoroughly investigate open-source cryptographic libraries that are suitable for IoT devices and evaluate their performance when implementing post-quantum techniques. By examining their effectiveness on limited hardware, this paper discusses how to choose the best cryptographic libraries to enhance IoT security.

This study emphasizes the importance of using post-quantum cryptography algorithms, such as Kyber512, in real-time applications on a Raspberry Pi 5 platform. We will utilize real-world sensor data to evaluate the encryption and decryption capabilities of post-quantum cryptography in comparison to traditional AES and RSA encryption methods. By implementing these algorithms on an IoT device, we can assess their processing overhead and effectiveness.

The objective of this research is to create scientific graphs that depict the temporal complexity of both encryption and decryption processes for traditional and post-quantum cryptography methods. These visualizations will make it easier for researchers and practitioners to assess the trade-offs between security and performance, facilitating an impartial comparison of computing costs. The analysis will consider various performance parameters, including processing time, memory usage, and energy consumption, to provide a comprehensive understanding of the viability of cryptographic techniques on IoT devices.

This research endeavor seeks to advance the security of Internet of Things (IoT) systems by addressing the increasing necessity for quantum-resistant cryptographic solutions within resource-constrained environments. Utilizing PyCryptodome and Kyber512, both open-source cryptographic libraries, we will conduct a thorough evaluation of the practical implementation of these algorithms. The findings derived from this investigation will equip researchers, developers, and industry practitioners with the requisite knowledge to make informed choices regarding the application of cryptographic libraries and algorithms in IoT settings. Ultimately, this study aspires to bridge the dichotomy between theoretical advancements in cryptography and their pragmatic implementations in IoT, thereby fostering a secure and resilient framework for interconnected devices.

2. Innovative Technology

Cryptography methods for protecting IoT devices have been continuously developed in response to growing threats and advancements in computing power. Traditional cryptographic approaches, which have historically provided strong security for digital communications, now need to be adapted and optimized to address the limitations of Internet of Things (IoT) technology. With the rise of quantum computing, there is an urgent demand for post-quantum cryptography (PQC) solutions that can ensure long-term security for IoT systems.

Methods for protecting IoT devices have been continuously developed in response to growing threats and advancements in computing power. Traditional cryptographic approaches(Becker, A., et al. (2019; Bernstein, D. J., Lange, T., et al.(2017)), which have historically provided strong security for digital communications, must now be adapted and optimized to address the limitations of Internet of Things (IoT) technology. With the rise of quantum computing, there is an urgent need for post-quantum cryptography (PQC) solutions that will ensure long-term security for IoT systems.

2.1 Advancements in cryptographic techniques for the protection of IoT

To ensure data integrity, confidentiality, and authentication, IoT devices rely heavily on cryptographic methods. These cryptographic techniques have evolved over time to address the challenges posed by limited processing power, memory constraints, and the need for energy efficiency in IoT hardware.

Traditional Cryptographic Algorithms
IoT security relied on proven cryptographic techniques like:

- The symmetric encryption method known as AES (Advanced Encryption Standard) is frequently employed to guarantee efficiency and security in data storage and communication.
- Public keys are used by the RSA (Rivest-Shamir-Adleman) method to facilitate safe key exchange. It is less appropriate for resource-constrained environments, such as IoT devices, due to its computationally demanding nature.
- Elliptic Curve Cryptography (ECC): This effective substitute for RSA provides robust security with reduced key sizes, which makes it ideal for contexts with constraints.

Despite their efficacy, the computing demands and energy consumption of these conventional cryptographic techniques present difficulties in Internet of Things applications.

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLVIII-4-W16-2025-91-2025 | © Author(s) 2025. CC BY 4.0 License.

Lightweight Cryptography

Researchers have developed lightweight cryptographic algorithms specifically designed for IoT devices to address the limitations of traditional cryptographic methods. Some of these innovations include:

- Lightweight AES Variants : These algorithms maintain security while reducing computational and memory requirements.
- Hummingbird and Grain : These are stream ciphers developed for restricted environments, enabling efficient encryption with minimal processing power in IoT settings.
- TinyCrypt: This library is tailored for embedded devices, providing essential cryptographic features while consuming minimal resources.

Lightweight cryptography strikes a balance between security and efficiency, making it suitable for various IoT applications. However, despite advancements in this field, concerns remain regarding its resilience against potential quantum threats.

Emergence of Post-Quantum Cryptography (PQC)

The rise of quantum computing presents a significant threat to traditional cryptographic techniques, especially those based on integer factorization (RSA) and discrete logarithms (ECC). In response, researchers have developed post-quantum cryptographic (PQC) algorithms designed to withstand quantum attacks. Several notable approaches to PQC have emerged:Lattice-based Cryptography (e.g.The efficient key generation and encryption processes provide a strong defense against quantum attacks.

Code-based cryptography has long been a secure solution against quantum adversaries. Multivariate and hash-based cryptography offer secure digital signatures and authentication mechanisms as alternative post-quantum cryptography techniques.

Kyber512 is emerging as a strong candidate for post-quantum cryptography in IoT, offering a balance of security and performance. This study evaluates its practicality for real-world IoT applications by implementing Kyber512 on Raspberry Pi hardware (Ducas, L., et al., (2022); Pöppelmann, T., et al. (2019); Sendrier, N. et al (2018).

Comparative Analysis of Cryptographic Techniques for IoT

The security, efficiency, and feasibility of IoT applications must be carefully weighed against the trade-offs associated with different cryptographic methods. This comparative analysis highlights the importance of finding a balance between security and efficiency, particularly for post-quantum cryptographic solutions. Research focused on implementing post-quantum cryptography (PQC) algorithms in constrained IoT devices is crucial for ensuring that IoT security remains resilient against potential quantum threats.

Additionally, investigating how to evaluate open-source cryptographic libraries is essential, with a specific emphasis on real-time implementation and performance benchmarking of post-quantum cryptography in IoT environments.

3. Data Gathering from Sensor Community

Evaluating the performance of cryptographic systems requires a systematic and automated approach to extracting, processing, and structuring data. To simplify this process, a variety of tools and methodologies were used to enhance data extraction, transformation, and storage. The processed data is crucial for analyzing the encryption and decryption performance of both classical and post-quantum cryptographic libraries on IoT hardware.

3.1 Implementation of Automated Data Extraction methods using Selenium and Python

Selenium, a popular web automation tool, was utilized to interact dynamically with the Sensor community platform. The system aimed to:

- Identify and engage with map-based sensor locations by detecting active hexagons for available sensor nodes.
- Automate navigation and selection of sensor locations by clicking on unvisited green hexagons to retrieve air quality data.
- Extract structured sensor readings, including PM2.5 concentration levels and timestamps, ensuring chronological accuracy.
- Manage delays and dynamic elements through explicit waits to prevent errors from incomplete page loading.

This automated approach enhances consistency, scalability, and adaptability, making it effective for long-term evaluations of IoT-based sensor networks.

- Reduces manual errors for improved precision and consistency.
- Enables real-time monitoring for up-to-date cryptographic evaluations.
- Enhances scalability for easy expansion to additional sensor locations.
- Cuts processing time, allowing for quick extraction of large datasets.

The resulting dataset supports cryptographic performance evaluations, particularly for benchmarking encryption and decryption times in real-world IoT conditions.

3.2 Processing & Structuring Data into Excel using OpenPyXL.

Once the raw data was extracted, it was essential to process and organize it into a readable format for analysis. OpenPyXL, a Python library for Excel files, facilitated efficient data management.

Data Processing and Structuring Workflow

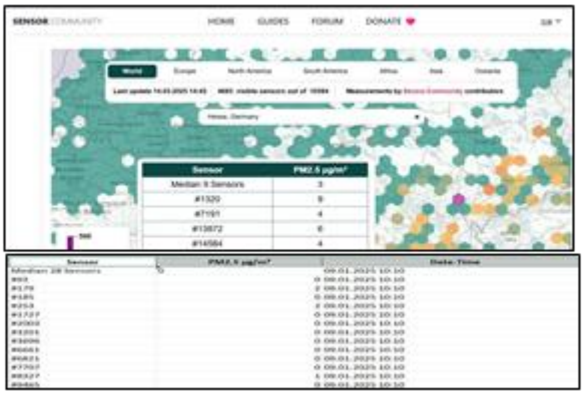
- Data Cleaning: Identified and corrected inconsistencies and empty values to ensure reliable analysis.
- Data Structuring: Organized sensor readings, PM2.5 levels, and timestamps into distinct columns in an Excel spreadsheet.
- Data Formatting: Standardized timestamps into a consistent date-time format and ensured numeric values were comparable.

- Handling Missing Values: Used placeholders for missing data and developed strategies for interpolation or review.
- Storage Optimization: Reduced redundancy in the Excel file for easier data management.

Extraction of Data

This section provides a preview of the dataset retrieved through the automated extraction and processing pipeline. It includes sensor readings, PM2.5 concentration levels, and timestamps, offering a structured view of real-time environmental conditions. The dataset consists of three key columns:

- Sensor Number : Identifies the specific sensor, ensuring traceability in air quality monitoring.
- PM2.5 (µg/m³) Indicates the concentration of PM2.5 in micrograms per cubic meter, an essential metric for assessing air pollution.
- Date and Time: Records the exact timestamp of the data, allowing for chronological analysis.



Sensor	PM2.5 µg/m³
Median 8 Sensors	3
#1320	8
#7191	4
#13872	6
#14584	4

Figure 2 Data Extraction from Sensor Community.

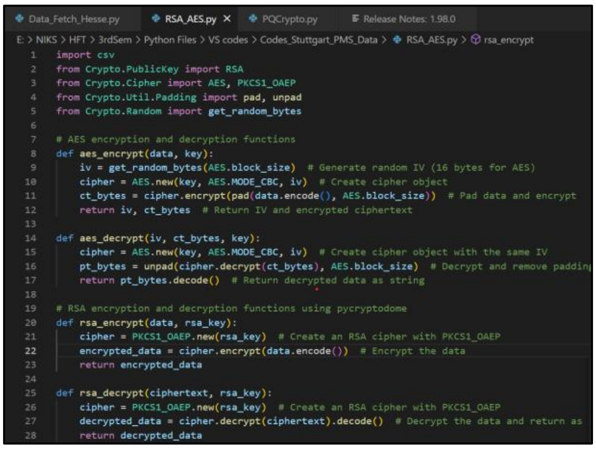
A Figure 1 is showing Extracted Data Set from Sensor community website and demonstrating the format and accuracy of the extracted data. This automated pipeline enhances data accuracy and consistency, facilitating the evaluation of environmental sensor data for cryptographic performance assessments in IoT environments.

4. Implementation and experiments

Cryptography plays a crucial role in ensuring the security of sensitive data in today's digital world. Protecting the confidentiality and integrity of information requires the use of encryption techniques. The purpose of this study is to explore two essential cryptographic algorithms that are widely utilized in data protection: AES (Advanced Encryption Standard) and RSA (Rivest-Shamir-Adleman) (Dworkin, M. (2021); Koblitz, N., et. al., (1985); "Advanced Encryption Standard (AES) - FIPS 197," (2001)).

- The same key is used for both encryption and decryption in AES, which is a symmetric encryption algorithm. It is highly regarded for its efficiency and is commonly employed to encrypt large datasets.
- A public-private key pair is used by RSA, an asymmetric encryption algorithm. Encoding smaller data pieces, such as encryption keys or secure communication, is typically what it is used for.

The aim of this segment is to show how to use both AES and RSA encryption/decryption methods to secure sensor data.


Data_Fetch_Hesse.py
RSA_AES.py
PQCrypto.py
Release Notes: 1.98.0
E: > NIKS > HFT > 3rdSem > Python Files > VS codes > Codes_Stuttgart_PMS_Data > RSA_AES.py > rsa_encrypt

```
import csv
from Crypto.PublicKey import RSA
from Crypto.Cipher import AES, PKCS1_OAEP
from Crypto.Util.Padding import pad, unpad
from Crypto.Random import get_random_bytes

# AES encryption and decryption functions
def aes_encrypt(data, key):
    iv = get_random_bytes(AES.block_size)  # Generate random IV (16 bytes for AES)
    cipher = AES.new(key, AES.MODE_CBC, iv)  # Create cipher object
    ct_bytes = cipher.encrypt(pad(data.encode(), AES.block_size))  # Pad data and encrypt
    return iv, ct_bytes  # Return IV and encrypted ciphertext

def aes_decrypt(iv, ct_bytes, key):
    cipher = AES.new(key, AES.MODE_CBC, iv)  # Create cipher object with the same IV
    pt_bytes = unpad(cipher.decrypt(ct_bytes), AES.block_size)  # Decrypt and remove paddin
    return pt_bytes.decode()  # Return decrypted data as string

# RSA encryption and decryption functions using pycryptodome
def rsa_encrypt(data, rsa_key):
    cipher = PKCS1_OAEP.new(rsa_key)  # Create an RSA cipher with PKCS1_OAEP
    encrypted_data = cipher.encrypt(data.encode())  # Encrypt the data
    return encrypted_data

def rsa_decrypt(ciphertext, rsa_key):
    cipher = PKCS1_OAEP.new(rsa_key)  # Create an RSA cipher with PKCS1_OAEP
    decrypted_data = cipher.decrypt(ciphertext).decode()  # Decrypt the data and return as
    return decrypted_data
```

4.1 AES & RSA Encryption

It is possible to break down the implementation of AES and RSA encryption and decryption into a series of steps.

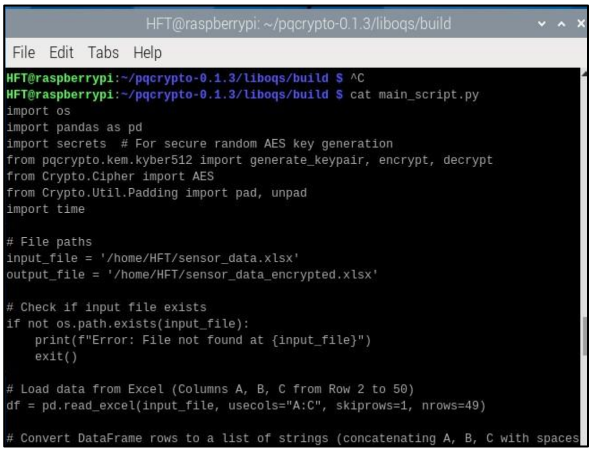
Figure 1 Classical cryptography Implementation Code Snippet

Above is outlines each step in the implementation process. The code snippet provided in Figure 2 for effectively demonstrates the overall procedure, ensuring clarity and ease of understanding for all stakeholders involved.

The Pycryptodome library and CSV Module are used to implement classical cryptography by manipulating encrypted and decrypted datasets.

4.2 Implementation Post-Quantum Cryptography

The exponential expansion of quantum computing is putting traditional encryption systems at risk, leading to the need for more robust security measures. Due to the sensitive data they handle, secure communication is necessary for IoT devices. Quantum computers can solve mathematical problems that are simple to solve, making common encryption techniques like RSA and ECC vulnerable to future vulnerabilities.



```
HFT@raspberrypi:~/pqcrypto-0.1.3/liboqs/build $ ^C
HFT@raspberrypi:~/pqcrypto-0.1.3/liboqs/build $ cat main_script.py
import os
import pandas as pd
import secrets  # For secure random AES key generation
from pqcrypto.kem.kyber512 import generate_keypair, encrypt, decrypt
from Crypto.Cipher import AES
from Crypto.Util.Padding import pad, unpad
import time

# File paths
input_file = '/home/HFT/sensor_data.xlsx'
output_file = '/home/HFT/sensor_data_encrypted.xlsx'

# Check if input file exists
if not os.path.exists(input_file):
    print(f"Error: File not found at {input_file}")
    exit()

# Load data from Excel (Columns A, B, C from Row 2 to 50)
df = pd.read_excel(input_file, usecols="A:C", skiprows=1, nrows=49)

# Convert DataFrame rows to a list of strings (concatenating A, B, C with spaces
```

Figure 3 Post Quantum Implementation Code Snippet

Post-quantum cryptography (PQC) is being created in figure 3 to tackle this issue and safeguard data even in the case of quantum computers becoming even more powerful. The goal of this is to secure secure IoT data by utilizing Kyber512, a hybrid encryption technique. Combining Kyber512 for secure key exchange and AES for fast data encryption ensures effectiveness and defense against quantum attacks(Becker, A., et al. (2019); Ding, J. (2009); Ducas, L. et al., (2022)).

4.3 The use of Kyber512 as post-quantum cryptography is being implemented.

Kyber512 has emerged as a top contender for the standardization of quantum-safe cryptographic algorithms by the National Institute of Standards and Technology (NIST) due to its great efficiency and robust security assurances.

Kyber512 is constructed on the Learning With Errors (LWE) problem, a mathematical foundation that is immune to quantum attacks, unlike traditional asymmetric encryption methods that depend on integer factorization or elliptic curve difficulties (Ducas, L. et al(2022)).

Why Kyber512 for IoT Security?

Post-quantum cryptography (PQC) can enhance security, but it can be challenging to implement many of these algorithms on resource-constrained IoT devices due to their high computational requirements. The lightweight and efficient nature of Kyber512 makes it ideal for IoT applications. Kyber512 was chosen for this implementation due to the following reasons:

Quantum Security: Resistant to attacks from both classical and quantum computers.

Optimized for Embedded Systems: Efficient enough to be deployed on IoT devices.

Fast Key Encapsulation & Decapsulation: Ensures quick secure key exchange for real- time applications.

Hybrid Cryptography Compatibility: Can be seamlessly integrated with AES to enhance efficiency.

The use of Kyber512 for key encapsulation in this study ensures that the AES encryption key is secure against quantum attacks even if an adversary intercepts the transmission (Ducas, L. et al(2022); Pöppelmann, T., et al.(2019)).

5. Result and Discussion

The effectiveness of conventional and post-quantum cryptography techniques in Internet of Things (IoT) settings is thoroughly examined in this section. The studies on a Raspberry Pi 5 platform provide actual proof of the effects of various cryptographic techniques on feasibility, security, and processing efficiency. Because IoT devices have limited processing capacity, choosing the right cryptographic algorithms is essential to providing safe but effective data storage and transfer. Our benchmarking and real-time implementation studies yielded data that shed light on processing trade-offs, encryption and decryption overheads, and useful suggestions for implementing these cryptographic techniques in actual Internet of Things applications.

Performance Analysis of Cryptographic Algorithms

1. Benchmarking Encryption and Decryption Times

This phase focused on analyzing the AES algorithm's performance with varying key sizes and plaintext lengths, given its widespread use in IoT applications. The results highlight how processing time correlates with data volume and key size.

Key observations from the AES benchmarking:

- AES-128 consistently outperforms AES-256 in processing speed while ensuring strong security, making AES-128 more suitable for resource-constrained IoT devices.
- Encryption and decryption times increase with larger plaintext sizes, suggesting that optimizing data packet sizes can enhance cryptographic efficiency in IoT.
- AES-256 encryption is nearly 40% slower than AES-128, indicating a clear performance-security trade-off. While AES-256 provides a higher level of security, its increased processing overhead may not be ideal for time-sensitive IoT operations.

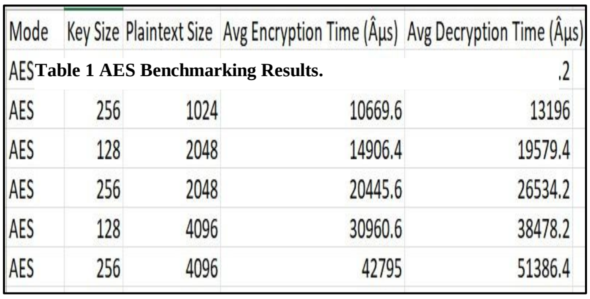


Table 1 AES Benchmarking Results.

Mode	Key Size	Plaintext Size	Avg Encryption Time (Âµs)	Avg Decryption Time (Âµs)
AES	[illegible]	[illegible]	[illegible]	.2
AES	256	1024	10669.6	13196
AES	128	2048	14906.4	19579.4
AES	256	2048	20445.6	26534.2
AES	128	4096	30960.6	38478.2
AES	256	4096	42795	51386.4

The benchmarking results are summarized in Table 1, which presents the average encryption and decryption times for different AES configurations.

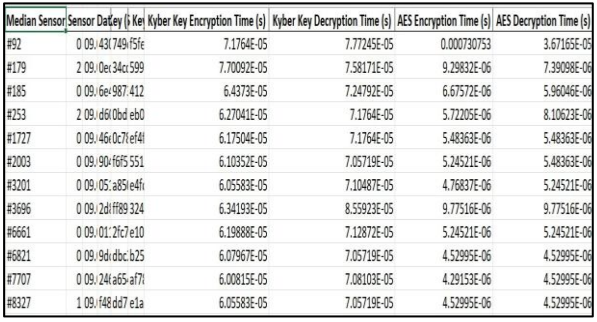
Hybrid Encryption Scheme: Real-Time Performance Analysis.

A hybrid encryption scheme was implemented to evaluate the integration of post-quantum cryptography into IoT security. It utilized:

- AES for fast data encryption of real-time sensor readings.
- Kyber512 for securing AES keys, adding quantum resistance to key exchange.
- The primary goal of this implementation was to assess the computational cost of using a post-quantum algorithm in combination with a classical encryption scheme.

Table 2 Real-Time Sensor Data - Encryption and Decryption Performance

The encryption and decryption times for real-time sensor data were measured, considering both AES data encryption and

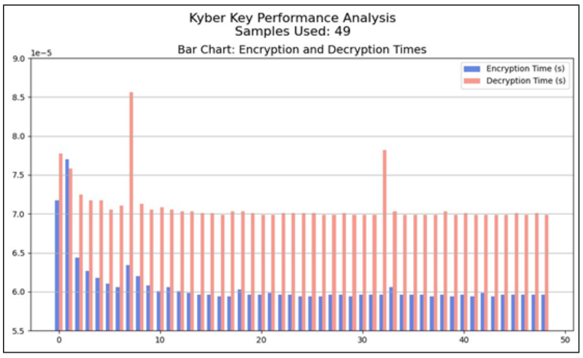


Median Sensor	Sensor Dat			Key (!	Key	Kyber Key Encryption Time (s)	Kyber Key Decryption Time (s)	AES Encryption Time (s)	AES Decryption Time (s)
#92	0	09.	43(	749	f5fe	7.1764E-05	7.77245E-05	0.000730753	3.67165E-05
#179	2	09.	0ec	34cc	599	7.70092E-05	7.58171E-05	9.29832E-06	7.39098E-06
#185	0	09.	6e	987:	412	6.4373E-05	7.24792E-05	6.67572E-06	5.96046E-06
#253	2	09.	d6(	0bd	eb0	6.27041E-05	7.1764E-05	5.72205E-06	8.10623E-06
#1727	0	09.	46	0c7	ef4	6.17504E-05	7.1764E-05	5.48363E-06	5.48363E-06
#2003	0	09.	90	f6f5	551	6.10352E-05	7.05719E-05	5.24521E-06	5.48363E-06
#3201	0	09.	05:	a85	e4f	6.05583E-05	7.10487E-05	4.76837E-06	5.24521E-06
#3696	0	09.	2d	ff89	324	6.34193E-05	8.55923E-05	9.77516E-06	9.77516E-06
#6661	0	09.	01:	2fc7	e10	6.19888E-05	7.12872E-05	5.24521E-06	5.24521E-06
#6821	0	09.	9d	dbc:	b25	6.07967E-05	7.05719E-05	4.52995E-06	4.52995E-06
#7707	0	09.	24(	a65	af7	6.00815E-05	7.08103E-05	4.29153E-06	4.52995E-06
#8327	1	09.	f48	dd7	e1a	6.05583E-05	7.05719E-05	4.52995E-06	4.52995E-06

Kyber512-based key encapsulation. The results are shown in Table 2.

The primary goal was to assess the computational cost of combining a post-quantum algorithm with classical encryption. Encryption and decryption times for sensor data were measured, with results summarized in Table 2.

Graph 1 Encryption and Decryption times for Post Quantum Implementation plotted on a bar chart.



Above Graph 1 Scientific graphs generated using the dataset to visualize sensor data acquisition, post-quantum Kyber512 encryption and decryption, and AES encryption and decryption performance.

AES key generation, post-quantum encryption and decryption using Kyber512, along with AES encryption and decryption, with recorded times for each process to simulate real-world applications of post-quantum cryptography in IoT security.

6. Conclusion

The limited resources of hardware and limited resources still pose a major challenge when it comes to ensuring secure data transmission and storage in IoT ecosystems. With an emphasis on Advanced Encryption Standard (AES) for classical encryption and Kyber512 for quantum-resistant cryptography, this study examined the effectiveness of both conventional cryptographic algorithms and post-quantum encryption techniques on Internet of Things devices. Through experiments on the Raspberry Pi 5 platform, this study compared computational effectiveness, encryption strength, and viability in limited settings.

The study's outcomes emphasize the importance of deciding on cryptographic approaches for Internet of Things applications, taking security, processing speed, and hardware capabilities into account. This study also highlights crucial performance trade-offs and potential disadvantages of utilizing post-quantum encryption on IoT hardware. The findings are part of the ongoing discussions on how to safeguard IoT security from emerging quantum threats while maintaining performance efficiency.

Summary of Key Findings

The research revealed numerous significant insights about the feasibility and efficiency of cryptographic techniques in IoT environments:

- Increasing key and plaintext sizes significantly impact processing speed, which could pose limitations for real-time IoT applications, according to the evaluation of AES encryption and decryption times.
- Kyber512 encryption and decryption, which has post-quantum cryptographic efficiency, was found to be a viable alternative to securing IoT data; However, increased computational overhead was observed, particularly when compared to conventional AES encryption.
- Despite performing adequately in traditional cryptographic libraries, post-quantum cryptographic implementations required additional processing resources, potentially affecting low-power IoT devices, as demonstrated by the experiments on the Raspberry Pi 5.
- Scientific graphs and data analysis were used to effectively present the relationship between cryptographic strength, computational overhead, and processing time through visual visualization of performance metrics, offering a visual comparison of trade-offs.
- Post-quantum cryptography has been shown to improve security in IoT, but practical implementation on resource-limited devices requires optimization and further research to reduce computational overhead, according to the findings.

This research provides a base for enhancing cryptographic frameworks in IoT applications and can aid in identifying the most appropriate encryption techniques for specific use cases.

Acknowledgements

This work is an outcome of the project "Datasecurity4icity", a subproject of the project "ICity: Intelligent city" (https://www.hft-stuttgart.com/research/projects/i-city)". We extend our gratitude for the funding received through the FH-Impuls program under the number 13FH9E04IA by the German Federal Ministry of Education and Research (BMBF).

References

Becker, A., et al. "Kyber: A KEM with the Best Security and Efficiency Characteristics for Post-Quantum Cryptography," IACR Cryptology ePrint Archive, 2019. Available at:
https ://eprint . iacr. o rg /20 19/ 701. pdf.

Bernstein, D. J., Lange, T., et al. "Post-Quantum Cryptography: Current State and Open Problems," IACR Cryptology ePrint Archive, 2017. Available at: https ://eprint . iacr. o rg /20 17/ 939. pdf.

Ding, J. "Multivariate Public Key Cryptography," Post-Quantum Cryptography, Springer, 2009.

Dworkin, M. "Recommendation for Block Cipher Modes of Operation," NIST Special Publication 800-38A, 2001.

Ducas, L., E. Alkim, P. Schwabe, "Kyber Post-Quantum KEM," NIST PQC Standardization Conference, 2022.

Koblitz, N., V. Miller, "Elliptic Curve Cryptography," Mathematics of Computation,1985.

National Institute of Standards and Technology (NIST), "Advanced Encryption Standard (AES) - FIPS 197," 2001.

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLVIII-4-W16-2025-91-2025 | © Author(s) 2025. CC BY 4.0 License.

National Institute of Standards and Technology (NIST), "Post-Quantum Cryptography Standardization," National Institute of Standards and Technology, 2022.

Pöppelmann, T., et al. "Kyber: A New Hope for Post-Quantum Public-Key Encryption," Proceedings of the 2019 ACM Conference on Computer and Communications Security (CCS'19), ACM, 2019. DOI: 10.1145/3319535.3354234.

Sendrier, N. "Code-Based Cryptography for Post-Quantum Security," Information Security Conference, 2018.

T. Boneh, "Cryptographic Trends in IoT," Journal of Cryptographic Engineering, 2020.

W. S. P. "Post-Quantum Cryptography Standardization," National Institute of Standards and Technology, 2022.