Towards μJoule PQC

“How does the post-quantum transition impact mobile device energy budgets?”

Markku-Juhani O. Saarinen
mjos@pqshield.com

PQShield Ltd. – Oxford, UK

ETSI/IQC Quantum Safe Cryptography Workshop 2019
November 7, 2019 – Seattle, United States
I Introduction \textit{[mostly skipping due to time limit]}

II Measuring NIST Post-Quantum Algorithms

III Wireless Communication Energy

IV Conclusions
November 2019: The NIST PQC submission deadline was two years ago, and we’re roughly halfway through the project, with 26 algorithms remaining.

Assumption 1: Little impact on symmetric cryptography

→ Most bulk data transfer still with AEADs (e.g. AES-GCM) and stream ciphers (e.g. ZUC). PQC Impacts mainly handshake and authentication.

Assumption 2: No fundamental protocol re-engineering

→ IETF and ETSI have been sitting on the fence, waiting for NIST to finish.
→ Quantum-secure signature and KEM algorithms can use equivalent external APIs to current standard ECDSA, ECDH, RSA cryptography.
→ Drop-in replacement to most current applications and protocols.
Quick recap: Power and Energy

These physical measures are surprisingly often confused.

**Electrical Power and Energy**

\[
P = V \times I \\
E = P \times t
\]

Energy (J: Joule) = Power (W: Watt) × Time (s: Second)

Lovely older units: Calorie (1 cal = 4.184 J), horsepower (1 hp = 764 W), etc.

- **Power** is momentary, **energy** is cumulative (think velocity vs. distance).
- To measure energy (J) we integrate (or “sum”) power (W) over time.
- Voltage (V) is usually a known, regulated value (such as 3 V). We can use an ammeter to measure the current (Amps). Power is the product.
Common Derived Units

An older energy integrator (meter) using kWh = 3.6 MJ.

3.85 V × 3 Ah × 3600 s/h = 41.6 kJ.

- Derived units: e.g. kW\text{h} (kilowatt hour) = 1000 W × 3600 s/h = 3.6 MJ.
- Batteries are often specified in mA\text{h} (milliampere hour). One needs to know the voltage to compute the actual energy that the battery has.
From integrated circuit theory:

**Dynamic Power Equation**

\[ P_{\text{dyn}} = \alpha \cdot C \cdot V^2 \cdot f \]

- \( P = \text{Power} \)
- \( \alpha = \text{activity} \)
- \( C = \text{Capacitance} \)
- \( V = \text{Voltage} \)
- \( f = \text{Frequency} \)

- **Dynamic power** dissipation \( P_{\text{dyn}} \) is caused by activity in the circuit.
- \( P_{\text{dyn}} \) is generally linear to frequency and area, quadratic to voltage.
- **Activity** \( \alpha \) is sometimes called “**switching factor**” as the energy is consumed when the circuit transitions from one state to another.
- The \( \alpha \) of a processor can vary a lot, depending on what it is doing.
- **Static power** dissipation \( P_{\text{stat}} \) when the circuit is idle. \( P = P_{\text{stat}} + P_{\text{dyn}} \).
Most CPUs have one or more **sleep states**, also affecting peripherals.

When asleep, instructions are not executed: $P_{\text{dyn}}$ is very low.

MCUs typically wake up only via an **interrupt** (timer or external event).

Modern CPUs can also control (“scale”) their clock frequency $f$ (and $V$).

A typical power consumption model for IoT microcontrollers (“sleepy edge node”).
This is what “active” can look like on a real MCU (ARM Cortex M4).

Power is rapidly fluctuating between 35mW and 100mW (3× range).

Cycle count is clearly not telling the full story about this algorithm.
Talk Outline

I  Introduction [mostly skipping due to time limit]

II  Measuring NIST Post-Quantum Algorithms

III  Wireless Communication Energy

IV  Conclusions
New PQC “IoT” Energy Measurements

- LPM01A “PowerShield” £50 power measurement board is also used for the EEMBC IoTConnect™ benchmarks.
- STM32F411RE target has a Cortex M4 core, the reference embedded platform of the NIST PQC project.
- I measured PQC implementations from the PQM4 project, also Ken MacKay’s “micro-ecc” ECDSA & ECDH code.
- **Goal:** Precise, independently repeatable.

Source code and a description of the lab:
https://github.com/mjosaarinen/pqps

It’s a dev board sandwich: LPM01A sits on top of the Nucleo64 target.
STM32F411RE: Average Power @ 96 MHz

Let me filter this data for you..
Cortex M4: Distinctive KEM Clusters (1/3)

- ECDH-p256 (reference)
- Kyber and NewHope
- Three Bears
- ECDH-secp256r1
- NTRU Prime

Time [s] (logarithmic)
Average Power [mW]
Cortex M4: Some Signature Algorithms

- Dilithium
- FALCON (sign, kg)
- ECDSA-p256 (reference)
- FALCON (verify)
- Dilithium2
- Dilithium3
- Dilithium4
- Falcon-1024
- Falcon-512

Graph showing the average power and time for various signature algorithms on Cortex M4.
In the microjoule range there are many examples where algorithm’s rank by timing is different from rank by energy.

**Meaningful, but:**

Are there general techniques to trade power for time?

Do the very distinctive power profiles translate to other microcontroller targets?
A log-log plot of the NIST PQC 2nd round set looks quite linear.

There is a range of over **four orders of magnitude** in the complexity of PQC algorithms.

This completely dwarfs the observed $\approx 50\%$ range in power. So “cycle counts” can be used to estimate energy, but results have **less than one significant digit of precision**.
Intel PCs: RAPL (Running Average Power Limit)

Intel PC/Server Measurements

- Inspired by [1], I modified the “official” SUPERCOP benchmarking system to record energy usage via Intel’s RAPL.
- Profiled 159 variants of about 20 NIST PQC algorithms in the benchmark.
- Power is highly dependent on target, but within that target not as varied as with IoT MCUs. Platform $[nJ/cycle]$ and cycle count leads to a good estimate.

https://github.com/mjosaarinen/pqps/tree/master/suppercop

Talk Outline

I Introduction [mostly skipping due to time limit]

II Measuring NIST Post-Quantum Algorithms

III Wireless Communication Energy

IV Conclusions
All PQC candidate algorithms have larger key- and message sizes than current RSA and Elliptic Curve cryptography. How much is too much?

**Total Energy: Compute it + Transmit it**

\[
\begin{align*}
E_{KG} & + e_{tx} |\text{pubkey}| & \text{Key generation.} \\
E_{Enc} & + e_{tx} |\text{ciphertext}| & \text{Encapsulation.} \\
E_{Sign} & + e_{tx} |\text{signature}| & \text{Authentication.}
\end{align*}
\]

.. or whatever is relevant in the protocol in question.

- Uplink (transmit) energy \( e_{tx} \gg e_{rx} \) downlink (receive), both in Joule/bit.
- **Two** algorithm factors (complexity, message sizes) and **two** platform factors (computational efficiency and communication efficiency).
- We need **measurement data** to determine their relative importance.
Communication efficiency $e_{tx} \approx 0.1 \mu J/\text{bit}$ (pre 5G)

(Source: Mads Lauridsen, Aalborg University, 2013)
**KEMs:** Need to relate KG, Enc, Dec computation energy to transmit energy of Public Key and Ciphertext (PK, CT).

For **ephemeral key exchange** we may consider total energy $E_{KG} + E_{Enc} + E_{Dec} + e_{tx}(PK + CT)$.

For **signer** (auth): Consider $E_{Sign} + e_{tx} \cdot SL$, where SL is signature length. **Verifier** $E_{Ver}$ also downloads ($e_{rx}$) some certificates which have public keys and signatures.. depends.
Example: Ephemeral Key Exchange

Use Case Observations

→ PQC Lattice (RLWR) can be $\approx 10\times$ more efficient than current ECDHE but needs $\approx 10\times$ bytes.

→ PQC Isogeny (SIDH) needs 30-60% of RLWR Lattice bytes, 300-2000 times more energy.

We can determine energy crossover points for $e^{tx}$ [J/bit] from lattice schemes to elliptic curves and SIDH.
Crossover from Round5 to ECDHE at $e_{tx} \approx 2 \mu J/\text{bit}$

- $L1$: $1.88 \mu J/\text{bit}$
- $L3$: $2.72 \mu J/\text{bit}$
- $L5$: $3.18 \mu J/\text{bit}$

*Estimated average for a “random” curve.

Crossover from Round5 to SIKE at $e_{tx} \approx 1 \text{mJ/bit}$

Transmission costs:
- $L1$: $0.77 \text{ mJ/bit}$
- $L3$: $1.10 \text{ mJ/bit}$
- $L5$: $1.44 \text{ mJ/bit}$

$(L1/L3/L5 = \text{classical and quantum security})$
Introduction [mostly skipping due to time limit]

Measuring NIST Post-Quantum Algorithms

Wireless Communication Energy

Conclusions
Conclusions

- PQC algorithms have really distinctive “IoT” MCU power profiles!
- Energy usage range is 4 orders of magnitude in NIST PQC 2nd round.
  - Energy ranking differs from time ranking most with < 1mJ algorithms.
  - Some PQC schemes actually need significantly less energy than ECC.
- Not so much power variation in our Intel PC / Server measurements.
- Consider transmission cost \([J/\text{bit}]\) and computation cost \([J/\text{cycle}]\).
  - ECC still has lower energy for (low bandwidth) channels \(e_{tx} \gtrsim 2 \mu J/\text{bit}\).
  - Isogeny/SIDH bandwidth savings are unlikely to lead to energy savings since computation cost dominates until \(e_{tx} \gtrsim 1 \text{mJ/} \text{bit}\) (very high).
  - Plotting total energy cost as a function of \(e_{tx}\) is a good way to compare algorithms A and B, where other is faster but needs more bandwidth.

Thank You!