Side-Channel Power Trace Dataset for Kyber Pair-Pointwise Multiplication on Cortex-M4

Azade Rezaeezade[1,3], Trevor Yap[2], Dirmanto Jap[2], Shivam Bhasin[2] and Stjepan Picek[3]

[1] Delft University of Technology, The Netherlands
a.rezaeezade-1@tudelft.nl
[2] Temasek Laboratories, Nanyang Technological University, Singapore
{sbhasin,djap,trevor.yap}@ntu.edu.sg
[3] Radboud University, Nijmegen, The Netherlands
stjepan.picek@ru.nl

Abstract. We present a dataset of side-channel power measurements captured during pair-pointwise multiplication in the decapsulation procedure of the Kyber Key Encapsulation Mechanism (KEM). The dataset targets the pair-pointwise multiplication step in the NTT domain, a key computational component of Kyber. The dataset is collected using the reference implementation from the PQClean project.We hope the dataset helps in research in “classical” power analysis and deep learning-based side-channel attacks on post-quantum cryptography (PQC).

Keywords: Side-channel Attacks, Post-Quantum Cryptography, CRYSTAL-Kyber, Dataset

1 Introduction

This dataset contributes to ongoing research in side-channel analysis (SCA) of post-quantum cryptographic schemes. The focus is on one of the basic operations in the Kyber KEM [BDK+18], a lattice-based cryptographic protocol selected by NIST for standardization. While the traces have been collected from a Kyber implementation, a similar operation with some differences has been used in other lattice-based key exchange mechanisms and digital signing algorithms like Falcon and Dilithium. Therefore, the dataset can be used as a toy example for researchers who want to focus on other PQC schemes as well.

The dataset, which we call reference-pair-pointwise-multiplication or Reference-PPM for short, is based on the reference implementation of Kyber from the PQClean repository[1] and uses the `PQCLEAN-MLKEM768-basemul` function as the target of observation. The dataset provides power traces and values of sensitive input and intermediate variables, enabling reproducible and interpretable SCA research.

2 Reference-PPM:

The Kyber reference implementation is known to be unprotected and very leaky. The implementation is in pure C and one can choose a different optimization level to compile it. The provided dataset here used -the o3 level.

[1] https://github.com/pq-crystals/kyber

Data Collection Setup: The Reference-PPM dataset was collected by executing the Kyber decapsulation procedure on an STM32F3 microcontroller, which features a 32-bit ARM Cortex-M4 core running at 7.372 MHz. Power measurements were acquired using the ChipWhisperer CW308 [New] platform in combination with a Lecroy 610Zi oscilloscope. The oscilloscope captured power consumption traces at a high sampling rate of 500 megasamples per second.

Each measurement corresponds to one execution of the decapsulation procedure using a fixed secret key and a varying ciphertext. A total of 100,000 traces were collected. Each trace consists of 50,000 time samples, and the measurements are focused specifically on the pairwise coefficient multiplication in the NTT domain as implemented in the `basemul` function shown in Listing 1.

```c
void basemul(int16_t r[2], const int16_t a[2], const int16_t b[2], int16_t zeta)
{
1:  r[0]  = fqmul(a[1], b[1]);
2:  r[0]  = fqmul(r[0], zeta);
3:  r[0] += fqmul(a[0], b[0]);
4:  r[1]  = fqmul(a[0], b[1]);
5:  r[1] += fqmul(a[1], b[0]);
}
```

Listing 1: Kyber pair-pointwise multipllication in NTT Domain

Dataset Content: The dataset contains two main components:

- Power traces: in total, Reference-PPM provides 100,000 traces, each trace represented with 50,000 time samples. The traces are stored in smaller MATLAB data files (.mat). The files are named in the format tracesA*99.mat, where the asterisk (*) represents a numeric identifier ranging from an empty string ("") to "999", indicating different measurement sets. Each set contains 100 traces, and in total, there are 1,000 trace files.

- Intermediate values: for each trace, there is the corresponding variable set that includes six values (2 of them are the same). All the values are shown in 2 bytes in little indian format. The final values after converting to decimal are the following variables in the NTT domain (see Listing 1):

  – The first two bytes make a[0] coefficient of the secret key.

  – The second two bytes make b[1] coefficient of the ciphertext.

  – The third two bytes make the result of pointwise multiplication of secret key and ciphertext coefficients according to line 4 in Listing 1).

  – The fourth two bytes make a[1] coefficient of the secret key.

  – The fifth 2 bytes are making b[1] again (repetitive).

  – The last two bytes are pointwise multiplication of a[1] and b[1] according to line 1 of Listing 1.

  As for the traces, these sensitive inputs and intermediate values are stored in smaller MATLAB data files. The files are named like noncesA*99.mat, following the same format and partitioning as traces.

The Python code for loading the traces and intermediate values is provided along with the dataset.

3 Availability

The dataset is available here.

4 Notes and Limitations

- The secret key is fixed for all measurements.
- Ciphertext input changes with each decapsulation execution.
- In Reference-PPM, only the `basemul` function is measured, not the full decapsulation.

Acknowledgment

This work was supported by the Netherlands Organization for Scientific Research NWO project DISTANT (CS.019).

References

[BDK+18] Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-kyber: a cca-secure module-lattice-based kem. In *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 353–367. IEEE, 2018.

[New] NewAE Technology. Chipwhisperer. https://newae.com/tools/chipwhisperer.