Project Proposals: Intro

- My research: PQC on Arm processors
- Arm designs CPUs; licenses to chip makers like ST, Infineon, NXP, Qualcomm, Samsung, MediaTek, Apple, ...
- More than 200 billion Arm chips built (so you should have around 25 of them...)
- Cortex-M cores: mircocontrollers (think credit cards/cars/IoT)
- Cortex-A cores: high end (smartphone/Raspberry Pi/Apple M1)

Project Proposal 1: Kyber NTT on Arm Cortex-M55

- Cortex-M55 is the newest generation fo the Cortex-M family
- Adds the M-profile vector extension (MVE) aka. Helium
- Much more powerful than previous M cores due to vector instructions
- Quite different from the A-profile cores (Neon vector extension)
- Only paper `https://eprint.iacr.org/2021/998` (implementing Saber)
- Goals of this project
  - Understand MVE (differences to A cores and previous M cores)
  - Understand the Kyber NTT
  - Implement the Kyber NTT on M55 (come up with a somewhat decent pipelining)
  - Possible extension (e.g., master thesis/paper): Full Kyber on M55
- Hardware
  - Can use qemu for functional testing
  - Soft-core M55 (FPGA prototyping platform) that can be accessed remotely

Project Proposal 2: Keccak on Cortex-M4 with FPU

- Arm Cortex-M4 is still widely used
- PQC heavily relies on the Keccak permutation
- Current implementation we are using is
  `https://github.com/mupq/pqm4/blob/master/common/keccakf1600.S`
- Idea: Use floating point registers to reduce memory accesses
- Goals of this project
  - Understand the Keccak permutation
  - Understand how the Keccak permutation is used in PQC
  - Improve current code using floating point registers on the M4 (or other tricks you can find)
  - Possible extension (e.g., master thesis/paper): Also optimize on A-profile/M55
- Hardware
  - You will be provided with Cortex-M4 development boards

Project Proposal 3: FrodoKEM on Cortex-A72 (Raspberry Pi 4)

- Every modern smartphone has a Cortex-A (or newer: Cortex-X)
- Neon extension very useful for implementing cryptography
- No implementation of FrodoKEM using Neon available yet
- Goal of this project
  - Understand the Neon instruction set
  - Implement FrodoKEM using Neon
  - Possible extension (e.g., master thesis/paper): Also implement FrodoKEM on M55
- Hardware
  - You will be provided with a Raspberry Pi 4
  - Can remotely access a Cortex-X1
  - If you have an Apple M1 you can use that, too