Efficient ML-KEM and ML-DSA on RISC-V with Optimized NTT

Gustavo Banegas – (Inria/IP Paris)

Context

Large-scale quantum computers will be able to break widely used public-key schemes such as RSA and ECDSA. Such a breakthrough would compromise the confidentiality and authenticity of digital communications, software distribution, etc. To counteract this threat, the National Institute of Standards and Technology (NIST) has standardized two lattice-based post-quantum schemes: ML-KEM (Kyber) for key encapsulation and ML-DSA (Dilithium) for digital signatures. Both schemes rely on polynomial multiplication in large finite rings—a computational bottleneck that is accelerated by the Number-Theoretic Transform (NTT). Efficient NTT and inverse NTT implementations are therefore important to achieving practical performance, especially when deploying these algorithms on resource-constrained RV32 microcontrollers.

Thomas Plantard [2] introduced a constant-multiplier technique with enlarged input ranges that accelerates NTT operations. More recently, [1] applied this method to low-resource devices, but without heavily optimized assembly. Extending and refining these ideas within the RISC-V ecosystem promises substantial gains in both speed and code size, making this project both timely and impactful.

Objectives

- Implement constant-time ML-KEM and ML-DSA baselines in portable C for RV32.
- Implement and benchmark an NTT/INTT in C, then hand-optimize an RV32 assembly version.
- Apply Plantard-style modular multiplication and lazy-reduction techniques to improve throughput and code size.
- Compare Montgomery/Barrett vs Plantard arithmetic, C vs assembly, and report cycle counts, memory use, and stack usage.

Expected Deliverables. RV32-C + assembly NTT library; ML-KEM/ML-DSA implementation in portable C, optimizations, and performance comparisons.

Practical information

Regular meetings will be scheduled with the advisors, either via video conference or in Palaiseau. If you are interested, send an email to `gustavo.souza-banegas@inria.fr`.

References

[1] Junhao Huang, Haosong Zhao, Jipeng Zhang, Wangchen Dai, Lu Zhou, Ray C. C. Cheung, Çetin Kaya Koç, and Donglong Chen. Yet another improvement of plantard arithmetic for faster kyber on low-end 32-bit iot devices. *IEEE Transactions on Information Forensics and Security*, 19:3800–3813, 2024.

[2] Thomas Plantard. Efficient word size modular arithmetic. *IEEE Transactions on Emerging Topics in Computing*, 9(3):1506–1518, 2021.