A Low-Power Side-Channel-Secure Configurable Accelerator for Post-Quantum Lattice-Based Cryptography

Utsav Banerjee, Tenzin S. Ukyab and Anantha P. Chandrakasan
Massachusetts Institute of Technology, Cambridge, MA, USA

ABSTRACT

Modern public key cryptography protocols are vulnerable to quantum attacks. Lattice-based cryptography is a prime candidate for post-quantum security. However, high computational complexity of these algorithms makes it challenging to implement them on low-power embedded devices. To address this challenge, we present a side-channel-secure configurable lattice crypto-processor, which provides order of magnitude improvement in performance and energy-efficiency compared to state-of-the-art.

1 INTRODUCTION

Modern public key protocols, such as elliptic curve cryptography (ECC), will be rendered insecure when large-scale quantum computers are built. Given the rapid advancement in quantum computing over the past few years, cryptographers are developing quantum-secure public key algorithms. Lattice-based cryptography is considered one of the most promising candidates for post-quantum security because of extensive security analysis as well as small public key and signature sizes. The National Institute of Standards and Technology (NIST) is currently in the process of standardizing post-quantum cryptography [1, 6], and *lattice-based cryptography* accounts for almost half of the candidates for key encapsulation and signature schemes. The theoretical foundation of several of these lattice-based protocols lies in the *learning with errors* (LWE) problem and its variants such as Ring-LWE and Module-LWE [9].

In this work, we present a *side-channel-secure configurable lattice cryptography processor*, which combines low-power modular arithmetic, area-efficient memory architecture and fast sampling techniques to achieve high energy-efficiency and low cycle count, ideal for securing low-power embedded systems. Our crypto-processor top-level architecture is shown in Fig. 1, and the key technical aspects of our design are as follows:

(1) A low-power modular arithmetic core, with run-time configurable 24-bit prime modulus, is used to accelerate polynomial arithmetic; a pseudo-configurable modular multiplier is also implemented, which provides up to 3× power savings.
(2) A single-port SRAM-based memory architecture for the number theoretic transform (NTT) provides 124k-gate area savings without any loss in performance or energy-efficiency.
(3) An efficient Keccak-f[1600] core is combined with fast polynomial sampling, while supporting a variety of discrete distribution parameters suitable for lattice-based protocols.
(4) These efficient hardware building blocks are integrated together with an instruction memory and decoder to build our crypto-processor, which can be programmed with custom instructions for polynomial sampling and arithmetic.
(5) Our crypto-processor is coupled with an efficient RISC-V micro-processor to demonstrate several NIST Round 2 lattice-based key encapsulation and signature protocols such as Frodo, NewHope, qTESLA, Kyber and Dilithium, achieving order of magnitude improvement in performance and energy-efficiency compared to state-of-the-art assembly-optimized software and hardware implementations.
(6) All the key building blocks, such as NTT, polynomial arithmetic and binomial sampling, are constant-time and secure against timing and simple power analysis (SPA) attacks. While our baseline protocol implementations are not secure against differential power analysis (DPA) attacks, the programmability of our crypto-processor can be utilized to implement masking-based DPA countermeasures.
(7) Our ASIC implementation was fabricated in the TSMC 40nm low-power CMOS process, and all protocol-level demonstrations and side-channel measurements have been conducted on our test chip.

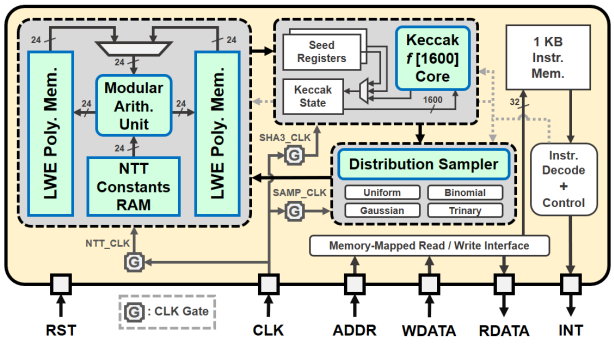

Figure 1: Lattice crypto-processor top-level architecture.

This abstract is adapted from our previous publications in ISSCC 2019 [2] and TCHES 2019 [3]. In the following sections, we briefly describe our design innovations and experimental results.

2 LOW-POWER MODULAR ARITHMETIC

For modular multiplication, we use a 24-bit multiplier followed by Barrett reduction [5], which is very efficient for small prime moduli and is also ideal for supporting configurability. We also implement a modular multiplier with pseudo-configurable modulus, with dedicated reduction logic for commonly used primes, which utilizes the special structure and mathematical properties of these primes to achieve 3× reduction in power consumption.

This modular multiplier is combined with a modular adder / subtractor to build the butterfly module. We implement a unified butterfly architecture which can be configured as both DIT and DIF, thus avoiding expensive bit reversals during NTT computation.

ISLPED Design Contest '20, August 10–12, 2020, Boston, MA, USA
2020 ACM/IEEE International Symposium on Low Power Electronics and Design.

3 AREA-EFFICIENT NTT MEMORY

Hardware architectures for polynomial multiplication using NTT consist of memory banks for storing the polynomials, with the arithmetic unit performing butterfly computations. These memory banks are typically implemented using dual-port or four-port RAMs, which poses large area overheads in resource-constrained devices. To solve this problem, we implement the constant geometry data flow [8] and split each polynomial among 4 single-port RAMs, allowing butterfly inputs and outputs to ping-pong between them without read / write hazards. This architecture provides 124k-gate area savings compared to the traditional approach. We also exploit mathematical properties of the twiddle factors to compress the memory required to store pre-computed tables by 38%.

4 ENERGY-EFFICIENT SAMPLER

Sampling accounts for 60-70% of the computation cost in software implementations of lattice-based protocols [7]. Our discrete distribution sampler reduces this overhead and provides up to two orders of magnitude energy savings over assembly-optimized software. The most important component of our sampler is a 24-cycle 34k-gate Keccak-f[1600] core which can be configured in different SHA-3 modes and used for hashing and pseudo-random bit generation. This is coupled with efficient algorithms for rejection sampling, binomial sampling and several other distributions with configurable parameters. Our design is also constant-time, thus providing higher side-channel security.

5 IMPLEMENTATION RESULTS

Our system, shown in Fig. 2, couples the lattice cryptography accelerator with a low-power RISC-V micro-processor, thus allowing us to demonstrate lattice-based key encapsulation and digital signature protocols through efficient software-hardware co-design. The crypto-processor consists of 106k logic gates and 40.25 KB SRAM, with a total area of 0.28 mm^2 (logic and memory combined). Our test chip supports voltage scaling between 0.68 V and 1.1 V, with maximum operating frequency 12 MHz and 72 MHz respectively.

Fig. 3 shows our evaluation board and setup for power measurement and side-channel analysis. Fig. 4 shows the energy consumption of our hardware-software co-design implementations of key encapsulation schemes NewHope-CCA, Kyber-CCA and Frodo-CCA and digital signature schemes qTesla and Dilithium at different

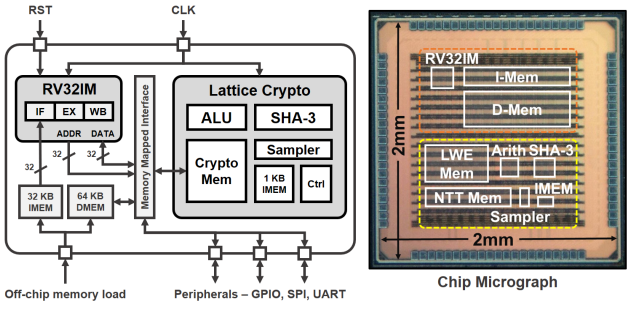

Figure 2: System architecture and test chip.

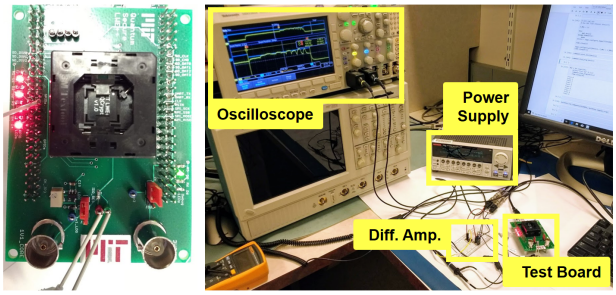

Figure 3: Evaluation board and measurement setup.

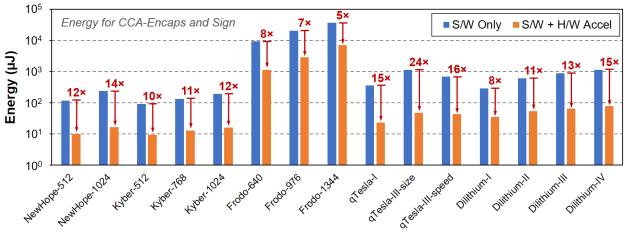

Figure 4: Protocol-level measurement results.

security levels. Compared with pure software implementations, we achieve an order of magnitude improvement in energy-efficiency.

We have also performed extensive measurements to confirm the side-channel security of our chip against timing and simple power analysis (SPA) attacks. All key building blocks, including sampling, polynomial arithmetic and number theoretic transform, are constant-time. Energy consumption of sampling and polynomial arithmetic follows a narrow distribution with coefficient of variation ($= \sigma/\mu$) $\leq$ 0.5%. The SPA resistance of these computations was also verified using difference-of-means test with 99.99% confidence interval. For further architectural details, measurement results and side-channel evaluation, please refer to [3, 4].

REFERENCES

[1] G. Alagic et al. 2019. *Status Report on the First Round of the NIST Post-Quantum Cryptography Standardization Process.* Technical Report 8240. National Institute of Standards and Technology.
[2] U. Banerjee, A. Pathak, and A. P. Chandrakasan. 2019. An Energy-Efficient Configurable Lattice Cryptography Processor for the Quantum-Secure Internet of Things. In *IEEE International Solid-State Circuits Conference (ISSCC).* 46–48.
[3] U. Banerjee, T. S. Ukyab, and A. P. Chandrakasan. 2019. Sapphire: A Configurable Crypto-Processor for Post-Quantum Lattice-based Protocols. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2019, 4 (Aug. 2019), 17–61.
[4] U. Banerjee, T. S. Ukyab, and A. P. Chandrakasan. 2019. Sapphire: A Configurable Crypto-Processor for Post-Quantum Lattice-based Protocols (Extended Version). Cryptology ePrint Archive, Report 2019/1140. https://eprint.iacr.org/2019/1140.
[5] P. Barrett. 1986. Implementing the Rivest Shamir and Adleman Public Key Encryption Algorithm on a Standard Digital Signal Processor. In *CRYPTO '86.* 311–323.
[6] L. Chen et al. 2016. *Report on Post-Quantum Cryptography.* Technical Report 8105. National Institute of Standards and Technology.
[7] M. J. Kannwischer et al. 2019. pqm4: Testing and Benchmarking NIST PQC on ARM Cortex-M4. Cryptology ePrint Archive, Report 2019/844.
[8] M. C. Pease. 1968. An Adaptation of the Fast Fourier Transform for Parallel Processing. *J. ACM* 15, 2 (Apr. 1968), 252–264.
[9] O. Regev. 2005. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. In *Proc. Symposium on Theory of Computing (STOC).* 84–93.