Jérémi Do Dinh

jdodinh.io | jdodinh | jdodinh | jeremi.dodinh@gmail.com

Cryptographic engineer focused on Rust-based ZK proof systems and zkVM internals. At Ligero I've ported the Ligero verifier to Rust with custom integration of RISC Zero zkVM precompiles, developed an interactive concrete-security analysis tool for the Ligero protocol, contributed upstream to Plonky3, and built Claude skills for ZK circuit development. MSc at EPFL under Alessandro Chiesa and Giacomo Fenzi on simulation security in the random oracle model.

Experience

Cryptographic Engineer at Ligero June 2025 - May 2026

- Built a Rust port of the Ligero verifier as a guest inside the RISC Zero VM, enabling remote proving on Base Mainnet via Boundless: verified `i32_mul` benchmarks at 32.76M cycles ($0.02 / 250k gas) and mDL attestation proofs at 390M cycles. Custom `sys_sha_compress` and `sys_bigint` precompiles deliver ∼2.2x speedup on Merkle verification and 4.5x on field-arithmetic steps, with a Montgomery-to-standard switch collapsing BN254 multiplication to a single ecall.
- Built a Rust macro based on Plonky3's BN254 implementation, generating field arithmetic for Ligetron's dual-root-of-unity domain layout.
- Investigated CRT-based polynomial encoding for non-NTT-friendly 256-bit primes (experimental Rust benchmark): benchmarked against direct NTT on BN254 (NTT-friendly control), achieving 24% speedup on 128-core cloud instances via custom Montgomery arithmetic with AVX2 SIMD, Rayon parallelism (enabled by upstream Plonky3 contributions), and Criterion-based benchmarking.
- Led the Ligetron documentation effort (docs.ligetron.com), including a concrete-security calculator for the Ligero protocol: computes interactive and non-interactive (Fiat-Shamir) soundness guarantees from user-supplied parameters via BCIKS20 proximity-gap bounds and round-by-round Fiat-Shamir analysis; supports BN254, Goldilocks, BabyBear, and Mersenne31 extensions across unique-decoding and Johnson-bound regimes.

Post Graduation Projects October 2024 - June 2025

- Implemented the sum-check protocol from scratch in Rust (arkworks).
- Studied STIR (debugging contributions to the Rust reference implementation) and WHIR (in Plonky3).

Software Engineering Intern at SonarSource September 2023 - February 2024

- Contributed to Python static-analysis features in a production Java codebase under a TDD + Scrum workflow.

Education

EPFL - MSc in Computer Science September 2021 - August 2024

- Thesis: *"Simulation Security in the Random Oracle Model"* – PDF (GPA: 5.25/6.0)
- Supervised by Alessandro Chiesa and Giacomo Fenzi.

McGill University - BSc in Mathematics & Computer Science September 2017 - April 2021

- Minor in Musical Science & Technology. (GPA: 3.87/4.0)
- Exchange semester at UBC Vancouver (January-April 2020).

Research & Publications

Tight inapproximability of well-supported Nash equilibria in public goods games 2023

– with Alexandros Hollender – ipl.2024.106486 arXiv:2402.14198

– Obtained hardness results for computing approximate equilibrium points in public goods games, significantly improving the previous upper bound. Completed at THL5, EPFL.

Integer Programming with Complete Constraint Matrices Report 2022

– Investigated properties of integer vectors ($b \in \mathbb{N}^m$) and their relation to the existence of a solution x on the binary hypercube such that $Ax = b$, where the constraint matrix $A \in \{0,1\}^{m \times 2^m}$ is *complete*.

– Master's Semester Project Supervised by Alexandra Lassota, DISOPT, EPFL.

Skills and Interests

Technologies	Rust, C++20, Git, CXX (Rust–C++ FFI), Python.
Cryptography	RISC Zero zkVM, proof-system frameworks (Plonky3, arkworks), IOP-based proof systems, NTT/FFT, Merkle trees, hash primitives (Poseidon2, SHA-256), field arithmetic (BN254, Goldilocks, BabyBear; Montgomery form).
Languages	*Fluent*: English, French, Polish. *Intermediate*: Italian, German (B1, fide-certified).
Interests	Playing guitar and songwriting, kitesurfing, skiing, tennis.