Mark (Zeheng) Mu Canadian

pqcnerd@gmail.com • **+1 (647) 960 8358** • https://www.linkedin.com/in/mark-mu/

Relevant Languages: Rust, Go, C, C++, Python, TypeScript, Javascript, Bash, SQL, CUDA, Haskell, Racket, Java, x86 Assembly, HTML, CSS
Relevant Tools: AWS (EC2, ECS, CloudFormation), Docker, Git, Linux, React, Next.js, Vim

EDUCATION

University of Toronto, Canada (mark.mu@mail.utoronto.ca) **Sep 2022 - Est. Dec 2026**

Honours Bachelor of Science, Computer Science
Relevant course work: Data Structures & Algorithms (CSC263), Software Design (CSC207), Operating Systems (CSC369), Computer Organization (CSC258), Systems Programming (CSC209), Discrete Mathematics (CSC236), Probability & Statistics (STA256), Linear Algebra (MAT223), Parallel Programming (CSC367), Machine Learning (CSC311), Cryptography & Security (CSC347), Programming Languages & Compilers (CSC324). Supported by knowledge gained by taking courses @edX, Coursera, Udacity, Eduonix, MITOCW.

PROFESSIONAL EXPERIENCE

Stealth Startup, Toronto, Ontario, Canada **Jul 2024 - Present**

Part Time Smart Contract Developer & Cryptographic Engineer (formerly Full Time)

- Implemented Solidity contracts (ERC-20/721/1155) with Hardhat/Foundry; invariant/property based fuzzing, static/dynamic analysis, and differential testing; storage/ABI layout tuning and opcode level gas optimization.
- DeFi protocol engineering (AMM + order book DEX, lending/borrowing, yield): oracle integration, DAO governance, liquidity/fee mechanics, slippage controls, and MEV/frontrunning mitigations (commit reveal, tx ordering assumptions).
- Privacy preserving DEX using zk-SNARKs (Groth16): confidential transfers with onchain verification; circuit tuning for MSM/FFT/NTT throughput; verifier gas reduction by constraint minimization and pairing precomputation; ZK features integrated into DeFi flows.
- Cosmos SDK modules (Go) and CosmWasm (Rust): IBC interoperability, ABCI/Tendermint integration, cross chain message handling; security audits and code reviews focused on cryptographic soundness and protocol attack surfaces.

SIGNIFICANT PROJECTS & EXTRACURRICULAR

VoteNow: Privacy-Preserving Decentralized Voting (priv. BitBucket repository) **May 2024 – Aug 2024**

- zk-SNARK protocol (Groth16) providing voter unlinkability and verifiable tally consistency; onchain verification with offchain Paillier ciphertext aggregation and deterministic nonce discipline.
- Circuit/R1CS optimizations (custom constraints, pairing precompute, batched modular arithmetic) to reduce proving time and verifier gas; homomorphic accumulation with Pedersen commitments; zero-knowledge double vote prevention.
- React+Ethers.js dashboard for ballot casting, proof visualization, & auditability on Ethereum testnets.

UoftCTF Team Member and Participant (https://ctftime.org/) **Nov 2024 - Present**

- Team results: Top 20 global / #1 Canada. (frequently varies, accurate at time of writing this)
- Cryptography, reverse engineering, blockchain: automated exploit pipelines (Python/Rust/SageMath) for RSA key recovery, lattice methods, DFA, side-channel modeling; binary/contract RE with

Ghidra/angr and LLVM IR instrumentation to surface cryptographic misuse and memory corruption vectors.

Minimal Linux OS, a Linux From Scratch Project (priv. GitHub repository) **May 2023 - Aug 2023**

- Deterministic, self hosted LFS toolchains; compiled/configured Linux kernel, glibc, GCC, binutils, systemd; cross compilation/linking in chroot; bootable ISO <100 MB (compressed).
- Bash automated multistage builds with dependency graph resolution, reproducible logs, and modular profiles; kernel performance analysis (perf/strace/ftrace), scheduling/memory/syscall instrumentation; minimal init and static-linking optimizations.

Scriptorium: Full-Stack Web Application (https://scriptorium-six.vercel.app/) **Sep 2023 - Dec 2023**

- Next.js (TypeScript) with SSR/dynamic routing/ISR; Node.js/Express microservices with SQLite persistence, session management, and auth middleware; Vercel CI/CD.
- Multi-runtime sandbox using Docker namespaces, seccomp, and cgroups for untrusted code execution; architecture aligned with 12 factor and REST constraints.
- For CSC309, every student has the same task but implementation can differ drastically.

Notare: AI-Powered Note Software & Web (https://notare-web.vercel.app) **Jan 2024 - Apr 2024**

- TypeScript/React/Next.js + Node/Express + MongoDB; microservices on Docker + AWS ECS; REST APIs with horizontal scaling, rate limited inference endpoints, and fault tolerant sessions.
- OpenAI integration for summarization/auto tagging/within note completions; async job queues for batch inference and token budgeting; JWT/OAuth2 with RBAC and multisession revocation; field level AES-GCM; indexed aggregations and concurrent write strategies; IaC (CloudFormation), CI/CD, environment scoped secrets.
- Resolved production incident where AI answers landed in wrong note locations (~7.8% of inserts) by replacing fragile char offset anchors with context aware, hash based anchors and grapheme aware tokenization, adding ETag based optimistic concurrency and a suggestion mode fallback; restored correctness in <2 hrs, reduced errors to 0.1%, and cut p95 latency 620ms -> 380ms.
- For CSC301 but unique idea planned and implemented at the behest of nobody but us.

Simplistic 3D Game Engine (priv. GitHub repository) **Sep 2020 - Jun 2022 & Sep 2024 - Oct 2024**

- C++17 + Vulkan 1.2: VMA managed memory; triple buffered swapchain; render passes/subpasses; pipeline cache reuse; descriptor sets by lifetime (per frame UBOs, per draw push constants, per material samplers); secondary command buffers; explicit barriers (layout/visibility) with fence/semaphore sync.
- Forward PBR renderer (Cook Torrance GGX + Schlick F_0), cascaded shadow maps (PCF), HDR + ACES tone mapping, MSAA resolve, depth pre-pass, frustum culling; shader hot reload (GLSL->SPIR-V, `glslc`), specialization constants, reflection driven binding generation.
- Validation + profiling with Vulkan layers, RenderDoc, Nsight; sustained 120 FPS on GTX 1060 @1080p in test scene; built with a peer using Git feature branches and code reviews.

MISCELLANEOUS / RECREATION

edX.org Harvard CS50 **Aug 2022**

Chess.com Player **1900 Rating**

END OF RESUME