Varun Thakore

MTech(RA), EE, IIT Bombay

vrnthakore@gmail.com varunthakore.github.io varunthakore

Research Interests

Applied Cryptography, Zero-Knowledge Proofs, Blockchains

Education

2021 – 2024	**Indian Institute of Technology Bombay** Master of Technology, Electrical Engineering. *Specialisation in Communication Engineering.*	GPA = 9.33/10.0
2015 – 2019	**Sardar Patel College of Engineering** Bachelor of Technology, Electrical Engineering.	GPA = 7.58/10.0

Publications and Drafts

1 **MProve-Nova: A Privacy-Preserving Proof of Reserves Protocol for Monero**
Varun Thakore and Saravanan Vijayakumaran.

Research Experience

2022–23 **Privacy-Preserving Proof of Reserves for Monero** *EE, IIT Bombay*
Prof. Saravanan Vijayakumaran | MTech Project
– Developed a **privacy-preserving** proof of reserves (POR) protocol for Monero based on **Nova**, such that the exchanges do not reveal the addresses and the amounts that they own.
– Implemented it in **Rust** which involves working with **non-native field** and **Merkle trees**.
– The protocol has a proving time of about **7Hrs** for **10,000** addresses. The verification time (**4.5s**) and proof size (**27KB**) are **constant** irrespective of the number of addresses.
– Implemented a **non-collusion** protocol to prevent exchanges from colluding to generate POR.

2022 **Review of Elliptic Curve Pairings** *EE, IIT Bombay*
Prof. Saravanan Vijayakumaran | MTech Seminar
– Studied **elliptic curves** including their representations, **Group law** and other properties.
– Surveyed literature on **bilinear pairings** including **Divisors** which are used to define pairings, **Weil pairing**, **Tate pairing** and **Miller's Algorithm** which is used to compute pairings.

Professional Experience

2024–* **Cryptography Engineering Intern** *zksecurity.xyz*
– Implementing a portable version of **Starkware's Stone** Prover and Verifier in Rust.
– Conduct comprehensive research on the **FRI protocol** and **Merkle commitment scheme**.
– Analyze and convert key functions from Starkware's C++ implementation to Rust.

2021–24 **System Administrator (Part-time), EE Department** *EE, IIT Bombay*
– **Headed** the transition of department Mail, Proxy, LDAP and Web Servers from Physical systems to **Virtual Machines** using virtualization platforms like **Proxmox VE**.
– Responsible for configuring and securing Dept. **Mail Servers** and **Network Infrastructure**.

2019-21 **Proposals Engineer - Hybrid and Energy Storage** *Sterling and Wilson Pvt Ltd*

*Currently in progress

Key Projects

2023 **Nova SHA-512** *Guide: Prof. Manoj Prabhakaran*
Course Project: Cryptography and Network Security (Won ZK MOOC Hackathon)
– Implemented R1CS for computation of **SHA-512** using **Rust** and **bellpepper** library. Implemented **SHA-512 compression function** as the step function within the Nova computation.
– For input of size **64 bytes**, proving time is **5.9s**, proof size is **10KB** and verification time is **268ms**.

Private ECDSA Signature Verification *Guide: Prof. Manoj Prabhakaran*
Course Project: Adv. Tools from Modern Cryptography
– Implemented R1CS circuit for **ECDSA** signature verification on **secp256k1** curve using **Rust** and **bellpepper** library. Involves writing circuit for curve operations in the base field of **secp256k1**.
– Circuits for point addition and scalar multiplication implemented in **36** and **3343** number of constraints, respectively and circuit for signature verification implemented in **3389** constraints.

R1CS Circuits for variants of Merkle Trees *Self Project*
– Implemented **regular Merkle tree** and R1CS to verify **inclusion proofs** using **bellpepper**
– Implemented **indexed Merkle tree** which is efficient to prove non-inclusion and R1CS circuits for **tree insertion**, verify **inclusion proof** and verify **non-inclusion proofs** using **bellpepper**

2024 **Nova Eddsa: High Throughput Ed25519 Signature Verification** *Self Project*
– Implemented R1CS circuit for **Ed25519** signature verification which is represented as a step function in Nova. Implemented in **Rust** and **bellpepper**, involves working with **non-native field**.
– For **32** signatures, proving time is **68s**, verification time is under **1s** and proof size is **11KB**.

Major Open Source Contributions
– Implemented circuit for **SHA-512** hash and **u64** representation in **bellpepper-gadgets**
– Optimized the **Nova** implementation by removing the absorbing of running instance
– Implemented **zero-knowledge** in **Nova** by porting relevant commits from an older version

Extracurricular Activities

2024 **Top 11** in **ZK Hack IV**, a global event which includes workshops and puzzle competition
2023 Won 2^{nd} prize for "**Category 2: Circuits for Recursive SNARKs**" of **ZK MOOC Hackathon** hosted by **UC Berkeley RDI**, which had **600 participants** from over **60 countries**
2022 **Teaching Assistant, ACM Winter School on Digital Trust, Trust Lab, IIT Bombay**
Teaching Instructor: Prof. Saravanan Vijayakumaran
– Assisted in conducting a workshop on **Smart Contract Development** for over **50** students.
– Workshop covered **Solidity**, compiling & deploying contracts using **Remix IDE** and **Hardhat**
2021 **Finalist** in **Shell.ai** Hackathon 2021 which had **2,000 registration** from over **50 countries**

Technical Skills

Programming	Rust, Python, Bash, Solidity, C, C++
Software & Tools	Bellpepper, Arkworks, Git, LaTeX, Pytorch, NumPy, Pandas, SciPy and Matplotlib

Relevant Coursework

Cryptography and Network Security	Information Theory and Coding	Error Correcting Codes
Adv. Tools from Modern Cryptography	Cryptocurrency & Blockchain Tech	Communication Networks
Game Theory and Mechanism Design	Foundations of Machine Learning	Advanced Machine Learning