ADITHYA BHAT

Purdue University, West Lafayette, IN
(+1) 765-476-3319 ⋄ bhat24@purdue.edu
Github - adithyabhatkajake ⋄ Linkedin - adithyabhatk

EDUCATION

Purdue University, West Lafayette *August 2018 - Present*
Ph.D. under Prof. Aniket Kate GPA: 3.88/4.0
Department of Computer Science

National Institute of Technology Karnataka (NITK) *August 2014 - May 2018*
Bachelor of Technology, Information Technology GPA: 9.50/10.0

WORK EXPERIENCE

Indian Statistical Institute, Kolkata July 2017 - December 2017
Undergraduate Research Fellow *Prof. Sushmitha Ruj*

· Worked on implementing a publicly verifiable data storage framework on Ethereum. We built an end-to-end system that takes a file, stores it on a machine and responds to challenges.

Morgan Stanley May 2017 - July 2017
Software Analyst

· Worked on evaluating an elastic-search visulization plugin for key-value database (leveldb).

Indian Institute of Science, Bangalore May 2016 - July 2016
Summer Research Fellow *Prof. C. E. Veni Madhavan*

· Worked on evaluating Pollard's rho, William's p+1, p-1 factorization and elliptic curve factorization methods for efficient batch factorization of numbers generated during the sieving phase of GGNFS.

PROJECTS

E2C - Energy Efficient Consensus
E2C is an energy efficient BA protocol for Cyber-Physical systems such as IoT devices. We employ cryptographic techniques and leverage networking techniques to optimize the energy costs of performing consensus and demonstrate the improvements using a test-bed and simulations.

Verifiable Time Lock Puzzles for Blockchains
We build an efficient verifiable time-locked puzzles using linearly homomorphic time-lock puzzles that can be used to remove timing information from any atomic multi hop lock system.

Reparo - Publicly Verifiable Repair Layer for any Blockchain
Reparo (PDF) is a consensus agnostic, backwards compatible edit layer for blockchains that allows, for example, in Ethereum to fix buggy contracts and undo transactions securely and with accountability.

Transitive Network - A Tokenless IOU based Credit Network Transitive Network is a credit network implementation in Ethereum using smart contracts. We show that it can be achieved without introducing tokens unlike Ripple and is even cheaper than using Ripple.

Publicly Verifiable Data Storage in Ethereum An academic implementation of Shachams and waters scheme in the Ethereum client as a native contract to enable publicly verifiable data storage.

INTERESTS

Byzantine Fault Tolerance (Distributed Systems)	Familiar with the works on deterministic consensus such as PBFT, Paxos, Zyzzyva, Hotstuff, Sync Hotstuff, SBFT, ... Familiar with works on randomized protocols for consensus in dynamic graphs.
Blockchains	Expert in Ethereum, implemented reparo and shachams and waters scheme, familiar with the codebase of Ethereum Client geth; familiar with Bitcoin , Lightning and ripple clients.
Applied Cryptography	Implemented brainpool EC curves for mbed-os from RFC 5639 and linearly homomorphic time-lock puzzles.

SKILLS

Programming Languages	C, Python, NodeJS, Go-lang, Solidity, C++
Libraries	GMP, PBC, OpenSSL, NS3, Web3, Ripple-Lib, Bitcoin-RPC
Softwares	Go-ethereum, Bitcoind, Rippled, Lnd, Blocksci NVim > Vim > Emacs, Tmux
Misc. Favorites	Bash scripting

RELEVANT COURSES

- Cryptography - CS555 (A)
- Practical and Applied Cryptography - CS590 (A)
- Complexity Theory - CS584 (A+)
- Data Communication and Networks - CS536 (A-)
- Compilers and Programming Systems - CS502 (A-)

ONGOING PROJECTS

Anonymous Notification Systems for Smart Contracts in Ethereum

An empirical analysis of Ripple Ledgers

Dynamic Graphs and Fault Tolerance for mobile CPS nodes

OTHER PROJECTS

Analysis of Paging and Caching overheads in the Linux Kernel (Linux Kernel, C) [2017]

K-Rack - A lightweight metadata encrypted steganographic filesystem (Linux Kernel, C) [2017]

A comparative analysis of Fast Fourier Transform using GPU and CPU. We find that 1-D FFT is faster on a CPU than on a GPU.

Daedalus - A tool implementing the attacks on RSA as proposed in Dan Boneh's survey paper *20 years of attacks on RSA* (Python) [2016]