Priyanka Mondal

+1-831-251-2070 • pmondal@ucsc.edu • priyanka-mondal.github.io
Priyanka-Mondal • mondalp

Summary

- **Broader interests:** Applied Cryptography, Security in Distributed Systems, Fault-tolerant Systems, Decentralized Consensus, Blockchain Technologies, Language-based Security
- **Experienced in:** Information Flow Control Policies, Oblivious Computation, Searchable Encryption, C++

Education

PhD, *Computer Science* , University of California, Santa Cruz **GPA: 4.0** 2017–June'24(expected)
Master of Engineering, *Computer Science* , Indian Institute of Science, Bangalore **GPA: 6.7/8.0** 2013-15
Bachelor of Engineering, *Computer Science* , Bengal Engineering & Science University, Kolkata 2009-13

Skills

Programming skills: C++, C, Java, Haskell, Coq, Dafny, HTML/CSS, Python, JavaScript, Matlab, Scala

Technical skills: Docker, Git, LaTeX, GDB, Wireshark, OpenSSL, SQL/MySQL, Django, Blockchain, AWS, VS Code, Bash, Linux, Unix, FreeBSD

Relevant Coursework: Cryptography, Programming Languages, Distributed systems, Computer Architecture, Relational Databases, Analysis of Algorithms, Automated verification, Data Mining, Compiler Design, Operating Systems, Probability and statistics

Research Experience

Research projects

- **System-wide security for Dynamic Searchable Encryption schemes**
  - Implemented Oblivious RAM with B-trees in C++, reducing the access time by **2-6**× than the existing AVL-tree based constructions
  - Mechanised an oblivious data-structure to completely eliminate information leakage during file retrieval phase from untrusted remote servers
- **Locality-aware Dynamic Searchable Encryption**
  - Invented a new algorithm that performs oblivious computation in de-amortized fashion in the cloud
  - Designed and implemented the novel searchable encryption algorithm in C++, that improves the search time by **4-179**×, both on disk (HDD/SSD) and in memory, than the existing counterparts
- **FLAQR: A language to implement consensus, replication and secret-sharing**
  - Designed a lambda calculus based programming language and its type system that enables programmers to write fault tolerant quorum protocols that are secure by construction
  - Added availability policies to Flow-Limited authorization model for the first time and implemented robustness proofs in Coq proof assistant to verify its security
  - Incorporated FLAQR's fault-tolerant language features into a Haskell library called HasChor, and added information flow control policies that ensures end-to-end information security
  - Invented two binary operators called *partial-and* & *partial-or* to realize security-policies in the presence of network failures (e.g. crash faults/byzantine faults)
- **Vote them out: Detecting and eliminating byzantine peers**
  - Developed an agreement protocol called `PEACH`, in which replicas vote against malicious nodes
  - Implemented correctness (safety and liveness) proofs in Alloy for byzantine fault-tolerant protocols
  - Developed Ethereum smart contracts in Solidity
- **Flowstate: A Language for Secure Replicated Computation**
  - Built a programming model for distributed shared memory based systems, that enforces confidentiality, integrity and availability policies
  - The language design supports optimistic concurrency control model, compatible with heterogeneous quorum

replication protocol, and enforces secure information flow between multiple clusters

- **Atomicity Checking with Blame Assignment for Android Applications**
  - Developed a bug detection tool in Java, which found 21 bugs in real world Android applications (e.g. Gmail, Wikipedia, MyTracks)
- **Intra and Inter Procedural Program Slicing**
  - Implemented a debugging tool in Java: given a program and a slicing criterion, this tool outputs a subset of program statements that help in understanding the flow of the code.

Selected publications

1. ***I/O-Efficient Dynamic Searchable Encryption meets Forward & Backward Privacy*** USENIX'24
   Priyanka Mondal, Javad Ghareh Chamani, Ioannis Demertzis, and Dimitrios Papadopoulos
2. ***Flow-Limited authorization for consensus, replication, and secret sharing*** JCS'23
   Priyanka Mondal, Maximilian Algehed and Owen Arden
3. ***Applying consensus and replication securely with FLAQR*** (Distinguished Paper Award) CSF'22
   Priyanka Mondal, Maximilian Algehed and Owen Arden

Industry Experience

- **Citrix R&D Pvt. Ltd, Bangalore.** Networking & Cloud team ***Software Engineer II***, 2015-17
  - Implemented an algorithm to transmit JSON data from Packet Engines to Amazon S3, that doubled the speed of the Unified Logger Daemon
  - In-charge of implementing an algorithm to convert HAProxy to Netscaler configuration
  - Developed an Wireshark plugin that increased efficiency of internal testing by 30%
- **Nomura Research Institute, Kolkata.** Enterprise Data Warehouse team ***Summer Intern***, 2012
  - Deployed an automated parsing technique to extract information from incoming XML data packets, resulting in 70% improvement of the system in-terms of speed

Additional Information

Selected Talks

- Applying replication and consensus securely with FLAQR - PLAS'21 (Virtual), CSF'22 (Technion, Israel)
- Flow Limited Authorization for Quorum Replication - PLCrypt'22 (Stanford Research Inst., Menlo Park)
- Flowstate: A Language for Secure Replicated Computation - CSF'19 (Stevens Institute, New Jersey)

Awards and Scholarships

- Distinguished paper award, CSF 2022
- Computer Security Foundations Travel Grant, 2019 and 2022
- Programming Languages Mentoring Workshop grant, 2019
- Oregon Programming Languages Summer School Student scholarship, 2018
- UC Santa Cruz Regents Fellowship, Winter 2018
- All India Council for Technical Education Scholarship, 2013-2015

Service and Outreach

- Vice President - Women in Cyber-security (WiCyS) students chapter, UC Santa Cruz
- Member - Women in Science and Engineering (WiSE), UC Santa Cruz
- External paper reviewer - AsiaCCS'24, Sigmod'23, FCS'22

Teaching Assistant Experience (at UC Santa Cruz)

- Foundations of Programming Languages - CSE114 (***Head Teaching Assistant***) Winter'24
  - Higher-order functions, Lambda calculus, Type-system, Haskell
- Programming Languages - CSE210 (***Graduate course***) Spring'21, Spring'23, Spring'24
  - Mentored students for their course projects, Formal verification in Coq proof assistant
- Advanced Programming - CMPS109 Spring'19, Spring'22
  - Object-oriented programming, Multi-threaded client/server applications
- Compiler Design, CMPS104 Fall'17