Priyanka Mondal

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

Summary

- 6+ years of experience as Security Researcher, and 2+ years of experience as Software Engineer
- **Broader interests:** Applied Cryptography, Security in Distributed Systems, Formal Verification, Language-based Security, Program Analysis, Oblivious Computation, Encrypted Search, Decentralized Consensus

Education

Ph.D., *Computer Science* , University of California, Santa Cruz, **GPA: 4.0/4.0** 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, **GPA: 8.1/10.0** 2009-13

Skills

Programming skills: C++(**proficient**), C, Java, Haskell, Coq, Python, HTML/CSS, Matlab

Technical skills: Docker, Git, LaTeX, GDB, OpenSSL, SQL, Django, AWS, VS Code, Bash, Linux, Unix

Research Experience

Graduate Research Assistant, 2018 - present

- **Secure and Efficient search on remotely stored Encrypted databases**
  - Designed and implemented a novel encrypted search algorithm in **C++**, that improves the search time on the remote database by **4-179**×, both on disk (HDD/SSD) and in memory, than the existing counterparts
  - Implemented a secure data-structure (Oblivious RAM) using B-trees in **C++**, reducing the access time by **2-6**× than the existing AVL-tree based construction (**7k+** lines of C++ code)
- **FLAQR: A programming model to securely implement consensus, replication and secret-sharing**
  - Designed a new lambda calculus-based programming model & **type-system** with information flow control policies, that enables programmers to write fault-tolerant, end-to-end secure distributed applications
  - Formally verified robustness of integrity, confidentiality, & availability policies of FLAQR language model using **Coq** proof assistant (**700+** lines of Coq code)
  - Implemented FLAQR's fault-tolerant language features in **Haskell** and incorporated them into HasChor library
  - Mathematically proved more than 15 Theorems (e.g. noninterference, liveness) for FLAQR language model
- **Detecting and eliminating malicious peers in a distributed consensus protocol**
  - Developed an agreement protocol called `PEACH`, in which replicas vote against malicious hosts
  - Implemented correctness (safety and liveness) proofs in **Alloy analyzer** for byzantine fault-tolerant protocols
- 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
- Developed a bug detection tool in **Java**, which found **21 bugs** in real world Android applications (e.g. Gmail)

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 in **Python** to transmit JSON data from Packet Engines to Amazon S3 buckets, that **doubled** the speed of the Unified Logger Daemon
  - In-charge of implementing an algorithm (in **C++, Shell scripts**) to convert HAProxy to Netscaler configuration
  - Fixed more than **20** existing bugs in the codebase of Netscaler load-balancer
  - 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 in **Java** to extract information from incoming XML data packets, resulting in **70%** improvement of the system in-terms of speed