

# Md (Muhammad) Shafiuzzaman

[mdshafiuzzaman@ucsb.edu](mailto:mdshafiuzzaman@ucsb.edu) | [shafiuzzaman-md.github.io](https://shafiuzzaman-md.github.io)

Ph.D. candidate in Computer Science at the University of California, Santa Barbara (UCSB), advised by [Prof. Tevfik Bultan](#). I work at the intersection of AI-assisted software engineering, software security, program analysis, and formal methods. My framework **STASE** (Static Analysis-Guided Symbolic Execution) was developed in a DARPA-funded project on software hardening, has uncovered previously unknown UEFI and Linux kernel vulnerabilities, confirmed multiple CVEs, and is being transitioned to U.S. Army adversarial-testing workflows.

## Research Interests

- **Building trust in AI-assisted systems:** Optimize precision-scalability trade-offs among symbolic, static, and dynamic analysis to deliver *secure-by-construction* code and patches in human–LLM coauthored software.
- **Scaling formal methods with AI:** Automate the construction of contracts, harnesses, invariants, and machine-checkable evidence so that verification scales to large, safety- and mission-critical software systems.
- **LLMs and agentic systems for software engineering:** Design, evaluate, and deploy symbolic-reasoning models and collaborative agents for complex development, testing, and maintenance workflows.
- **Cyber defense automation:** Build AI- and formal-methods-driven workflows for automated security assessment and end-to-end exploit discovery, triage, and mitigation.

## Education

Ph.D. (in progress), Computer Science, University of California, Santa Barbara (*Expected Summer 2026*)

*Dissertation: LLM-enhanced hybrid program analysis for automated discovery of vulnerabilities and exploit chains.*

M.S., Software Engineering, University of Dhaka (2016)

*Thesis: A Proactive Self-Adaptive Framework for Context-Aware Mobile Applications.*

B.S., Software Engineering, University of Dhaka (2014)

## Recent Publications

- STASE: Static Analysis Guided Symbolic Execution for UEFI Vulnerability Signature Generation. *ASE 2024*.  
[DOI:10.1145/3691620.3695543](#)
- Rare Path Guided Fuzzing. *ISSTA 2023*. [DOI:10.1145/3597926.3598136](#)
- Stateful Behavior Inference and Runtime Enforcement for Vehicle Network Security. *USENIX VehicleSec 2025*.  
[usenix.org/conference/vehiclesec25/presentation/desai](http://usenix.org/conference/vehiclesec25/presentation/desai)
- CHAINER: Discovering Vulnerability Chains with Memory-Aware Symbolic Analysis. *Under submission*.
- CEGIR-Harness: Static Analysis Guided, LLM-Assisted Harness Refinement for Symbolic Execution. *Under submission*.

Full publication list: [scholar.google.com/.../1mue4woAAAAJ](https://scholar.google.com/.../1mue4woAAAAJ).

## Research Experience

Graduate Student Researcher, Verification Lab (VLab), UCSB

2022–Present

- *DARPA HARDEN* (2022–Present): Vulnerability detection and exploit-chain reasoning; machine-checkable signatures; eval team participation; contributions toward UCSB’s “Top Performer” title.
- *U.S. Army TARDEC – Validating Vehicle Communications* (2023–2025): Runtime enforcement for vehicular networks; model-based validation of ECU protocol invariants.
- *Amazon Research – Information Leakages in Crypto Libraries* (2024): Program analysis and testing to detect and quantify side channels and information leakages in cryptographic libraries.
- *LLM-Enhanced Formal Methods for Effective Security Assessment* (2025–Present): Automated harness generation for symbolic execution using static analysis and LLMs; integration with fuzzing and exploit-chain reasoning pipelines.

## Research Artifacts, Tools & Community

- STASE (Static Analysis–Guided Symbolic Execution): Framework for automated vulnerability analysis ([github.com/shafiuzaaman-md/stase-2.0](https://github.com/shafiuzaaman-md/stase-2.0)).
- ChainBench: Curated vulnerability-chain dataset and harness suite for exploit-chain analysis ([github.com/shafiuzaaman-md/chainbench](https://github.com/shafiuzaaman-md/chainbench)).
- Service & community: Sub-reviewer for premier SE venues (ICSE, ASE, FSE, ISSTA); Artifact Evaluation participation.

## Grants & Proposals (Role: Contributor/Co-Author)

- DARPA ASEM, 2025: Assessing Security of Encrypted Messaging Applications.
- Sony Research Award Program, 2025: LLM-enhanced formal methods for embedded Linux security.
- NSF Future CoRe/SHF, 2025: Secure and Verifiable Code Generation via Specialized Agents.
- NSF SaTC Medium, 2024: Side-channels: Proofs, Attacks, and Defenses.
- DARPA TRACTOR, 2024: Translating C to Rust.
- Amazon Trusted Crypto, 2024: Detecting and Quantifying Information Leakages in Crypto Libraries.
- U.S. Army TARDEC, 2023: Validating Vehicle Communications Between Trusted and Untrusted Vehicle Control Systems.

## Teaching Experience

- Instructor, UCSB Summer 2022 Data Structures & Algorithms I – course design, lectures, assessments, mentoring.  
Materials: [github.com/shafiuzaaman-md/CS130A\\_Summer2022](https://github.com/shafiuzaaman-md/CS130A_Summer2022)
- Teaching Assistant, UCSB  
Fundamentals of Database Systems, Data Structures & Algorithms I/II. 2021–2022
- Lecturer, Jashore University of Science & Technology  
Software Engineering, Structured Programming, Algorithm Analysis & Design, Software Testing and Operations 2018–2021

## Academic Leadership & Mentoring

[Graduate Student Mentor](#), Graduate Scholars Program (2025–2026): Orientation support; research skill-building; coordination with Diversity & Outreach; Fellowship application preparation.

[Undergraduate Research Mentor](#), Early Research Scholars Program (ERSP), UCSB Computer Science (2023–2024; 2025–2026):

- Cohort 2023–2024 (4 students). Project: *Vulnerability Signature Generation* – symbolic execution to extract pre/postconditions at vulnerability sites for machine-checkable signatures.
- Cohort 2025–2026 (5 students). Project: *LLM-Assisted Fuzzing and Symbolic Execution Harness Generation* – agentic/LLM workflows to infer harnesses (symbolic inputs, stubs, assertions) from code/static findings and steer analysis toward likely bug sites.

## Pedagogy & Instructional Training

- Participated in UCSB [STIA](#) pedagogy program (modules on learning objectives, lesson planning, assessment, inclusive teaching).
- Developed Data Structures & Algorithms I syllabus, lecture plan, and instructional materials.

## Industry Experience

[Software Engineer Intern](#), Office of Research, UC Santa Barbara

Summer 2022

Integrated Central Authentication Service (CAS) single sign-on for enterprise apps; added role-based access and audit logging, reducing unauthorized access incidents and support tickets. (C#/NET)

<b>Software Engineer</b> , Mobil Jamuna Lubricants, Bangladesh	2016–2018
Re-engineered the Lube Enterprise System data pipeline and auth layers, improving data integrity and security posture across inventory, billing, and reporting. ( <i>C#/.NET, JavaScript, MySQL</i> )	
<b>Junior Software Engineer / Intern</b> , Wolters Kluwer, USA	2014–2016
<ul style="list-style-type: none"> <li>Modernized financial compliance modules handling sensitive U.S. mortgage data with input validation and test coverage expansion. (<i>C#/.NET, Angular/TypeScript, WebAPI, TDD, MS SQL</i>)</li> <li>Introduced CI-friendly unit/integration tests and database migration scripts, reducing regression defects across releases.</li> </ul>	

## Technical Skills

**AI/LLMs:** Prompt/agent design for code analysis, retrieval-augmented triage, evidence-bearing generation (spec/claim loops).

**Formal/Analysis:** SMT (Z3, Boolector), symbolic execution (KLEE), abstract interpretation, slicing, Datalog.

**Security/Testing:** Fuzzing (AFL++, libFuzzer), sanitizers (ASan/UBSan/MSan), exploit-chain modeling.

**Systems/Toolchains:** LLVM/Clang, GCC, Linux kernel build envs, UEFI EDK2.

**DevOps/Scale:** Docker, GitHub Actions (CI/CD).

**Programming:** Python, C, C#, Java/JS/TS, Go, Racket/Rosette, SMT-LIB.