TIANROU XIA

Research Scientist, Georgia Institute of Technology
txia98@gatech.edu

RESEARCH INTERESTS

I am broadly interested in system and software security, with a focus on program hardening, memory safety, static and dynamic analysis, vulnerability detection, and Rust security. More recently, my work has expanded toward forensics analysis of robotics and edge AI, as well as large language model security.

EDUCATION

The Pennsylvania State University, University Park, United States *Aug 2020 - Dec 2025*
Ph.D. in College of Information Sciences and Technology
Advisor: Dinghao Wu, co-advisor: Taegyu Kim.

The Pennsylvania State University, University Park, United States *Aug 2018 - Jul 2020*
M.S. in Department of Computer Science and Engineering
Advisor: Sencun Zhu.

Northeastern University, Shenyang China *Oct 2014 - Jun 2018*
B.E. in College of Information Security

PUBLICATIONS

LiteRSan: Lightweight Memory Safety Via Rust-specific Program Analysis and Selective Instrumentation
Tianrou Xia, Kaiming Huang, Dongyeon Yu, Yuseok Jeon, Jie Zhou, Dinghao Wu, Taegyu Kim
arXiv, 2025. [PDF]

DEEPTYPE: Refining Indirect Call Targets with Strong Multi-layer Type Analysis
Tianrou Xia, Hong Hu, and Dinghao Wu
USENIX Security Symposium, 2024. [PDF | Slides | Video]

Toward A Network-Assisted Approach for Effective Ransomware Detection
Tianrou Xia, Yuanyi Sun, Sencun Zhu, Zeeshan Rasheed, Khurram Shafique
EAI Endorsed Transactions on Security and Safety, 2021. [PDF]

SELECTED RESEARCH PROJECTS

Fault Localization for Rust programs *(Ongoing)*

- Developing a Rust-specific static analysis and post-mortem reasoning framework that leverages ownership and borrowing semantics to localize root causes of program crashes with improved precision.

Full Memory Safety *(Ongoing)*

- Designing a hybrid fat pointer scheme for C/C++ that combines static analysis–guided pointer classification with efficient in-place metadata to enforce spatial and temporal safety with minimal overhead.

Bridging Static and Dynamic Analysis for Indirect Calls

- Conducted iterative refinement between static and dynamic analysis tools to reconcile discrepancies in indirect call target resolution. Aimed to optimize both analyses and replace resolvable indirect calls with safe direct calls for improved performance and attack surface reduction.

Static Binary Reassembly and Instrumentation with Uroboros

- Contributed to the Uroboros project, a reassembly-based static binary instrumentation framework targeting stripped binaries. Assisted in symbol recovery and generation of relocatable assembly code, enabling program-wide static instrumentation without dynamic hooks or binary patching.

TEACHING EXPERIENCES

Guest Lecturer

- IST 597 – Cyber-Physical Systems / IoT Security: *Fall 2025*
  Topic: Memory Safety in Rust

Teaching Assistant, The Pennsylvania State University

- SRA 221 - Information Security: *Spring 2025*
- IST 454 - Computer and Cyber Forensics: *Spring 2023, Spring 2024*
- CYBER 100 - Computer Systems Literacy: *Fall 2023*
- IST 597 - Fairness, Incentives, and Mechanism Design: *Fall 2022*
- DS 402 - Games, Algorithms, and Social Choice: *Spring 2022, Fall 2022*
- SRA 268 - Visual Analytics: *Fall 2020*

Mentored Students

- Changyul Lee: M.S. Student, College of IST, The Pennsylvania State University
  Aug 2024 - Current
- Myeonghun Pak: B.S. Student, Department of Information Security, Mokpo National University
  May 2025 - Current

RESEARCH POSITIONS

Research Scientist, CyFI Lab, supervised by Prof. Brendan D. Saltaformaggio
Feb 2026 – Current

Research Assistant, Prof. Dinghao Wu's Team
Jan 2021 – Dec 2021, May 2022 - Aug 2022, May 2024 - Dec 2024

ACADEMIC SERVICE

- ACM CCS 2026, subreviewer
- IEEE ICDCS 2026, PC member

HONORS & AWARDS

IEEE S&P 2025 Student Travel Grants	*May 2025*
USENIX Security 2024 Student Grants	*Jul 2024*

Additional: First Prize – NEU Math Modeling (2016), NEU Scholarship (2018)

TECHNICAL SKILLS

Programming Languages: C, C++, Rust, Python, Java
Security & Analysis Tools: LLVM, SVF, AFL, LibFuzzer, IDA Pro, etc.
Build & Deployment: Docker, CMake, Make