Qingkai Shi

ADDRESS

School of Computer Science
Nanjing University
163 Xianlin Avenue
Nanjing, Jiangsu Province 210023, PRC
Email: qingkaishi@nju.edu.cn
URL: https://qingkaishi.github.io
ORCID: 0000-0002-8297-8998

SHORT BIO

Qingkai Shi is an associate professor in the School of Computer Science at Nanjing University. His research focuses on the use of compiler techniques, especially static program analysis, to rigorously ensure software security. He has published extensively at premium venues of programming languages (PLDI, OOPSLA), cybersecurity (SP, CCS), and software engineering (ICSE, ESEC/FSE). His research received many awards, including two ACM SIGPLAN Distinguished Paper Awards, two ACM SIGSOFT Distinguished Paper Awards, a Google Research Paper Reward, and the Hong Kong Ph.D. Fellowship.

Qingkai obtained his Ph.D. from the Hong Kong University of Science and Technology. He co-founded Sourcebrella LLC, where his research was commercialized. He then moved to Ant Group as Sourcebrella was acquired. Qingkai also enjoyed a wonderful period as a postdoctoral researcher at Purdue University.

EXPERIENCES

Nanjing University — 2023 -
- *Associate Professor*

Purdue University — 2021 - 2023
- *Postdoctoral Researcher*

Sourcebrella LLC, Ant Group — 2020 - 2021
- *Technical Expert*

Sourcebrella LLC — 2015 - 2020
- *Co-founder of Sourcebrella*

Hong Kong University of Science and Technology — 2015 - 2020
- *Ph.D. in Computer Science*

SELECTED PUBLICATIONS

1. **Qingkai Shi**, Xiaoheng Xie, Xianjin Fu, Peng Di, Huawei Li, Ang Zhou, and Gang Fan. Datalog-Based Language-Agnostic Change Impact Analysis for Microservices. *Proceedings of the ACM/IEEE International Conference on Software Engineering (ICSE'25)*. 2025.
2. **Qingkai Shi**, Junyang Shao, Yapeng Ye, Mingwei Zheng, and Xiangyu Zhang. Lifting Network Protocol Implementation to Precise Format Specification with Security Applications. *Proceedings of the ACM Conference on Computer and Communications Security (CCS'23)*. 2023.
3. **Qingkai Shi**, Xiangzhe Xu, and Xiangyu Zhang. Extracting Protocol Format as State Machine via Controlled Static Loop Analysis. *Proceedings of the USENIX Security Symposium (SEC'23)*. 2023.

4. **Qingkai Shi**, Yongchao Wang, Peisen Yao, and Charles Zhang. Indexing the Extended Dyck-CFL Reachability for Context-Sensitive Program Analysis. *Proceedings of the ACM on Programming Languages (OOPSLA'22)*. 2022.

5. **Qingkai Shi**, Peisen Yao, Rongxin Wu, and Charles Zhang. Path-Sensitive Sparse Analysis without Path Conditions. *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI'21)*. 2021.

6. **Qingkai Shi** and Charles Zhang. Pipelining Bottom-up Data Flow Analysis. *Proceedings of the ACM/IEEE International Conference on Software Engineering (ICSE'20)*. 2020.

7. **Qingkai Shi**, Rongxin Wu, Gang Fan, and Charles Zhang. Conquering the Extensional Scalability Problem for Value-Flow Analysis Frameworks. *Proceedings of the ACM/IEEE International Conference on Software Engineering (ICSE'20)*. 2020.

8. **Qingkai Shi**, Xiao Xiao, Rongxin Wu, Jinguo Zhou, Gang Fan, and Charles Zhang. Pinpoint: Fast and Precise Sparse Value Flow Analysis for Million Lines of Code. *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI'18)*. 2018.

Other Publications

9. Yi Sun, Chengpeng Wang, Gang Fan, **Qingkai Shi**, and Xiangyu Zhang. Fast and Precise Static Null Exception Analysis with Synergistic Preprocessing. *IEEE Transactions on Software Engineering (TSE'25)*. 2025.

10. Mingwei Zheng, **Qingkai Shi**, Xuwei Liu, Xiangzhe Xu, Le Yu, Congyu Liu, Guannan Wei, and Xiangyu Zhang. ParDiff: Practical Static Differential Analysis of Network Protocol Parsers. *Proceedings of the ACM on Programming Languages (OOPSLA'24)*. 2024. **ACM SIGPLAN Distinguished Paper Award**

11. Peisen Yao, Jinguo Zhou, Xiao Xiao, **Qingkai Shi**, Rongxin Wu, and Charles Zhang. Falcon: A Fused Approach to Path-Sensitive Sparse Data Dependence Analysis. *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI'24)*. 2024.

12. Wuqi Zhang, Zhuo Zhang, **Qingkai Shi**, Lu Liu, Lili Wei, Yepang Liu, Xiangyu Zhang, and Shing-Chi Cheung. Nyx: Detecting Exploitable Front-Running Vulnerabilities in Smart Contracts. *Proceedings of the IEEE Symposium on Security and Privacy (SP'24)*. 2024.

13. Shiwei Feng, Yapeng Ye, **Qingkai Shi**, Zhiyuan Cheng, Xiangzhe Xu, Siyuan Cheng, Hongjun Choi, and Xiangyu Zhang. ROCAS: Root Cause Analysis of Autonomous Driving Accidents via Cyber-Physical Co-mutation. *Proceedings of the ACM/IEEE International Conference on Automated Software Engineering (ASE'24)*. 2024. **ACM SIGSOFT Distinguished Paper Award**

14. Rongxin Wu, Y. He, J. Huang, C. Wang, W. Tang, **Qingkai Shi**, X. Xiao, and Charles Zhang. A Two-Layer Persistent Summary Design for Taming Third-Party Libraries in Static Bug-Finding Systems. *Proceedings of the ACM/IEEE International Conference on Software Engineering (ICSE'24)*. 2024.

15. Xizhe Yin, Yang Feng, **Qingkai Shi**, Zixi Liu, Hongwang Liu, and Baowen Xu.

FRIES: Fuzzing Rust Library Interactions via Efficient Ecosystem-Guided Target Generation. *Proceedings of the ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA'24)*. 2024.

16. Xinmeng Xia, Yang Feng, **Qingkai Shi**, James Jones, Xiangyu Zhang, and Baowen Xu. Enumerating Valid Non-Alpha-Equivalent Programs for Interpreter Testing. *ACM Transactions on Software Engineering and Methodology (TOSEM'24)*. 2024.

17. Yapeng Ye, Zhuo Zhang, **Qingkai Shi**, Yousra Aafer, and Xiangyu Zhang. D-ARM: Disassembling ARM Binaries by Lightweight Superset Instruction Interpretation and Graph Modeling. *Proceedings of the IEEE Symposium on Security and Privacy (SP'23)*. 2023.

18. Xiangzhe Xu, Z. Xun, S. Feng, S. Chen, Y. Ye, **Qingkai Shi**, G. Tao, L. Yu, Z. Zhang, and Xiangyu Zhang. PEM: Representing Binary Program Semantics for Similarity Analysis via A Probabilistic Execution Model. *Proceedings of the ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE'23)*. 2023.

19. Xiangzhe Xu, S. Feng, Y. Ye, G. Shen, Z. Su, S. Chen, G. Tao, **Qingkai Shi**, Z. Zhang, and Xiangyu Zhang. Improving Binary Code Similarity Transformer Models by Semantics-driven Instruction Deemphasis. *Proceedings of the ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA'23)*. 2023.

20. Heqing Huang, Hung-Chun Chiu, **Qingkai Shi**, Peisen Yao, and Charles Zhang. Balancing Seed Scheduling via Monte Carlo Planning. *IEEE Transactions on Dependable and Secure Computing (TDSC'23)*. 2023.

21. Chengpeng Wang, Wenyang Wang, Peisen Yao, **Qingkai Shi**, Jinguo Zhou, Xiao Xiao, and Charles Zhang. Anchor: Fast and Precise Value-Flow Analysis for Containers via Memory Orientation. *ACM Transactions on Software Engineering and Methodology (TOSEM'23)*. 2023.

22. Chengpeng Wang, Peisen Yao, Wensheng Tang, **Qingkai Shi**, and Charles Zhang. Complexity-Guided Container Replacement Synthesis. *Proceedings of the ACM on Programming Languages (OOPSLA'22)*. 2022. **ACM SIGPLAN Distinguished Paper Award**

23. Yuandao Cai, Chengfeng Ye, **Qingkai Shi**, and Charles Zhang. Peahen: Fast and Precise Static Deadlock Detection via Context Reduction. *Proceedings of the ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE'22)*. 2022.

24. Yiyuan Guo, Jinguo Zhou, Peisen Yao, **Qingkai Shi**, and Charles Zhang. Precise Divide-By-Zero Detection with Affirmative Evidence. *Proceedings of the ACM/IEEE International Conference on Software Engineering (ICSE'22)*. 2022.

25. Heqing Huang, Yiyuan Guo, **Qingkai Shi**, Peisen Yao, Rongxin Wu, and Charles Zhang. Beacon: Directed Grey-Box Fuzzing with Provable Path Pruning. *Proceedings of the IEEE Symposium on Security and Privacy (SP'22)*. 2022. **Google Research Paper Reward**

26. Peisen Yao, **Qingkai Shi**, Heqing Huang, and Charles Zhang. Program Analysis

via Efficient Symbolic Abstraction. *Proceedings of the ACM on Programming Languages (OOPSLA'21)*. 2021.

27. Peisen Yao, Heqing Huang, Wensheng Tang, **Qingkai Shi**, Rongxin Wu, and Charles Zhang. Skeletal Approximation Enumeration for SMT Solver Testing. *Proceedings of the ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE'21)*. 2021.

28. Peisen Yao, Heqing Huang, Wensheng Tang, **Qingkai Shi**, Rongxin Wu, and Charles Zhang. Fuzzing SMT Solvers via Two-Dimensional Input Space Exploration. *Proceedings of the ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA'21)*. 2021.

29. Heqing Huang, Peisen Yao, Rongxin Wu, **Qingkai Shi**, and Charles Zhang. Pangolin: Incremental Hybrid Fuzzing with Polyhedral Path Abstraction. *Proceedings of the IEEE Symposium on Security and Privacy (SP'20)*. 2020.

30. Peisen Yao, **Qingkai Shi**, Heqing Huang, and Charles Zhang. Fast Bit-Vector Satisfiability. *Proceedings of the ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA'20)*. 2020.

31. Gang Fan, Chengpeng Wang, Rongxin Wu, Xiao Xiao, **Qingkai Shi**, and Charles Zhang. Escaping Dependency Hell: Finding Build Dependency Errors with the Unified Dependency Graph. *Proceedings of the ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA'20)*. 2020.

32. Yang Feng, **Qingkai Shi**, Xinyu Gao, Jun Wan, Chunrong Fang, and Zhenyu Chen. DeepGini: Prioritizing Massive Tests to Enhance the Robustness of Deep Neural Networks. *Proceedings of the ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA'20)*. 2020.

33. Chunrong Fang, Zixi Liu, Yangyang Shi, Jeff Huang, and **Qingkai Shi**. Functional Code Clone Detection with Syntax and Semantics Fusion Learning. *Proceedings of the ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA'20)*. 2020.

34. Gang Fan, Rongxin Wu, **Qingkai Shi**, Xiao Xiao, Jinguo Zhou, and Charles Zhang. Smoke: Scalable Path-Sensitive Memory Leak Detection for Millions of Lines of Code. *Proceedings of the ACM/IEEE International Conference on Software Engineering (ICSE'19)*. 2019. **ACM SIGSOFT Distinguished Paper Award**

35. **Qingkai Shi**, Jeff Huang, Zhenyu Chen, and Baowen Xu. Verifying Synchronization for Atomicity Violation Fixing. *IEEE Transactions on Software Engineering (TSE'16)*. 2016.

36. **Qingkai Shi**, Zhenyu Chen, Chunrong Fang, Yang Feng, and Baowen Xu. Measuring the Diversity of a Test Set with Distance Entropy. *IEEE Transactions on Reliability (TR'16)*. 2016.

PATENTS

Defect detection method, device, system, and computer readable medium.

- US Patent No. 20190108003
- China Patent No. 201811013103, 201811013000, 201811015751, 2018110146864

SERVICE

Journal Reviewer

- *ACM Computing Surveys*
- *ACM Transactions on Software Engineering and Methodology*
- *IEEE Transactions on Software Engineering*
- *IEEE Transactions on Dependable and Secure Computing*
- *IEEE Transactions on Reliability*
- *IEEE Transactions on Emerging Topics in Computing*
- *Springer Automated Software Engineering*

Member of Program Committee

- *IEEE Symposium on Security and Privacy (SP'25)*
- *ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA'25)*
- *ACM Conference on Computer and Communications Security (CCS'24, CCS'25)*
- *IEEE/ACM International Conference on Automated Software Engineering (ASE'24)*

Member of External Review Committee

- *ACM International Conference on the Foundations of Software Engineering (FSE'25)*
- *European Conference on Object-Oriented Programming (ECOOP'23)*

Member of Artifact Evaluation Committee

- *ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA'24)*
- *European Conference on Object-Oriented Programming (ECOOP'23)*
- *ACM Symposium on Principles of Programming Languages (POPL'22, POPL'23)*

TEACHING

Data Structure (Spring 2024)

Principles of Compiler Design (Spring 2024)