Jinseong Jeon

425 Urban Plz, Kirkland, WA 98033

jsjeon@cs.umd.edu

https://jsjeon.github.io/

Education

- **University of Maryland, College Park** — College Park, MD
  Ph.D. in Computer Science — *Feb 2016*
- **KAIST** — Daejeon, Korea
  M.S. in Computer Science — *Feb 2007*
- **KAIST** — Daejeon, Korea
  B.S. in Computer Science — *Feb 2005*

Experience

- **Google LLC** — Kirkland, WA
  Tech Lead / Staff Software Engineer — *Mar 2023 / Nov 2024 – Present*
  - *Kotlin compiler at Google*: Leading Kotlin frontend and JVM compiler efforts within Google
    * Spearheaded K2 adoption in Google monorepo
    * Guided Android Studio migration to K2 and K2 IDE plugin
    * Drove tooling migrations: Android Lint and Metalava in AndroidX

- **Google LLC** — Kirkland, WA
  Senior Software Engineer — *May 2019 – Oct 2024*
  - *IntelliJ IDEA* (contributions): Designed and implemented K2 UAST from scratch:
    * Unified AST for Java and Kotlin, backbone of Android Lint
    * Landed to AndroidX: average 1.31x, up to 1.72x faster
    * Adopted by external companies: Meta, Square, and Mozilla
  - *Kotlin compiler* (contributions): Contributed to K2, the new Kotlin compiler frontend, end-to-end:
    * Deserialization: to load Kotlin stdlib, other libraries or modules
    * (old-school) parsing: frontend IR generation
    * Resolution: types, declarations, call targets, SAM/suspend conversion, etc.
    * Static analyses: control-flow / data-flow analysis, diagnostics
    * Conversion to backend IR
    * Serialization: to generate @Metadata for reflection

- **Google LLC** — Kirkland, WA
  Software Engineer — *Feb 2016 – Apr 2019*
  - *Android Compiler Toolchain*: D8 dexer and R8 shrinker (contributions): Researched; designed; implemented; and deployed optimizations and obfuscations, such as:
    * local type/nullability analysis, call-site optimization (e.g., remove Kotlin intrinsics calls),
    * `StringBuilder` optimization, compile-time reflection simplification, constant/call canonicalization,
    * Kotlin @Metadata rewriting, identifier string obfuscation, package obfuscation
  - *Google Compute Engine*: Sole-tenant nodes, Committed use discounts

- **University of Maryland, College Park** — College Park, MD
  Research Assistant — *Jun 2011 – Feb 2016*
  - PASKET: *Synthesizing Framework Models for Symbolic Execution* [1, 4, 6, 7, 14, 15]: Researched and developed scalable synthesis of models for object-oriented, event-driven frameworks, such as Java Swing and Android, that enable other static analysis tools to analyze real-world apps effectively and efficiently
  - *SymDroid: Symbolic Execution for Dalvik Bytecode* [5, 16, 18]: Developed a symbolic execution engine for Android apps, which explores apps' possible behaviors and discovers under what circumstance apps may disclose which sensitive user data

- *Brahmastra: Driving Apps to Test the Security of Third-Party Components* [8]: Designed and implemented a static analysis that extracts program execution paths from app's entry points to method calls of interest, e.g., authorization APIs in Facebook SDK, which are in turn used to steer apps so as to test security properties of third-party components in a large scale

- *Troyd: Integration Testing Framework for Android* [17]: Developed a script-based testing framework for Android that allows testers to run Android apps via command-line interface; record testing scenarios; and replay recorded scenarios as regression tests

- *Redexer: Dalvik Bytecode Instrumentation Framework* [9, 19, 20]: Developed a general-purpose bytecode rewriting framework for Android, which is composed of a rich set of utilities that let programmers parse, manipulate, and generate Dalvik bytecode from scratch

Google Inc. — Mountain View, CA
Software Engineering Intern — *May – Aug 2015*

- *Espresso Test Recorder*: Designed and prototyped an Android Studio plugin that records user interactions via instrumentation and synthesizes repeatable Espresso test code from the logs

Microsoft Research — Redmond, WA
Research Intern — *May – Aug 2014*

- *AppFormer: Synthesizing Cross-Platform Mappings from Examples*: Researched automatic creation of platform-to-platform mappings (e.g., Android to Windows Phone) by logging example apps' behaviors and summarizing them via template-based program synthesis

University of Maryland, College Park — College Park, MD
Teaching Assistant — *Aug 2010 – May 2011*

- *CMSC 330 Organization of Programming Languages by Dr. Jeffrey S. Foster*: Designed meta-scripts that generate testing scripts that conduct regression tests over hundreds of students' projects

- *CMSC 132 Object-Oriented Programming II by Larry Herman*: Assisted in teaching basic algorithms and data structures in Java

Agency for Defense Development — Daejeon, Korea
Researcher — *Feb 2007 – Jul 2010*

- Worked with an ELINT (ELectric INTelligence) team to build a pod-style ELINT system,

- Researched an adaptive way to test the system with a minimal number of input RF signals [10],

- Designed a compact, hard-to-decompose data format for ELINT system missions, and

- Developed a passive geolocation algorithm for RF signals

KAIST — Daejeon, Korea
Research Assistant — *Mar 2006 – Feb 2007*

- *RTFA: Layout Transformation for Heap Objects* [2, 12]: Developed a compiler optimization that infers data structure access patterns and transforms heap layouts to improve program performance by increasing cache hit ratios (*won an Outstanding Master's Thesis Award from the department*).

- *Raccon: Buffer Overrun Analyzer for C Programs* [3, 13]: Modified a buffer overrun analyzer for C programs so that it can distinguish k different call contexts during analysis

KAIST — Daejeon, Korea
Teaching Assistant — *Mar 2005 – Feb 2006*

- *CS220 Programming Principles by Dr. Taisook Han*: Assisted in teaching functional programming via Dr. Scheme, which is now Racket

- *CS230 System Programming by Dr. Jinsoo Kim*: Led a small group of students and helped them learn shell programming (*won an Excellent Teaching Assistant Award from the department*)

Tmax Soft — Seongnam, Korea
Internship — *Jul – Aug 2004*

- Programmed a simple bulletin board system (BBS) based on Java Servlet and JSP

Awards and Activities

Scholarships and Awards:

- CAV15 Student Scholarship 2015
- Google Fellowship Nominee, Dept. of Computer Science, UMD 2013
- Outstanding Master's Thesis Award, Dept. of Computer Science, KAIST 2007
- Excellent Teaching Assistant Award, Dept. of Computer Science, KAIST 2005
- *Magna Cum Laude*, KAIST 2005

Professional Activities:

- Artifact Evaluation Committee, PLDI '15, PLDI '20 2015, 2020
- Reviewer, IEEE Transactions on Mobile Computing (TMC) 2015, 2018
- Reviewer, POPL '15, ICSE '15 2014
- Reviewer, Journal of Information Security and Applications (JISA) 2014
- Reviewer, IEEE Transactions on Dependable and Secure Computing (TDSC) 2013

Extracurricular Activities:

- Chair of Academic Affair, Washington Metro Chapter, Korean-American Scientist and Engineer's Association (KSEA) 2014 – 2015
- Staff of Korea Graduate Student Associates (KGSA), UMD 2013 – 2015
- Vice president of Korea Graduate Student Associates (KGSA), UMD 2011 – 2013
- Vice president of the senior class in Dept. of Computer Science, KAIST 2004

Publications

Journal Articles

[1] Jinseong Jeon, Xiaokang Qiu, Armando Solar-Lezama, and Jeffrey S. Foster. An Empirical Study of Adaptive Concretization for Parallel Program Synthesis. *Formal Methods in System Design (FMSD)*, 50(1):75–95, Mar 2017.

[2] Jinseong Jeon, Keoncheol Shin, and Hwansoo Han. Abstracting Access Patterns of Dynamic Memory Using Regular Expressions. *ACM Transactions on Architecture and Code Optimization (TACO)*, 5(4):18:1–18:28, Mar 2009.

[3] Youil Kim, Jinseong Jeon, and Hwansoo Han. Development of Cost-Effective Buffer Overrun Analyzer for C Programs. *KIISE SIGPL Transactions on Programming Languages*, 19(2):1–9, 2005.

Conference/Workshop Proceedings

[4] Jinseong Jeon, Xiaokang Qiu, Jonathan Fetter-Degges, Jeffrey S. Foster, and Armando Solar-Lezama. Synthesizing Framework Models for Symbolic Execution. In *38th International Conference on Software Engineering (ICSE '16)*, May 2016.

[5] Kristopher K. Micinski, Jonathan Fetter-Degges, Jinseong Jeon, Jeffrey S. Foster, and Michael R. Clarkson. Checking Interaction-Based Declassification Policies for Android Using Symbolic Execution. In *the 20th European Symposium on Research in Computer Security (ESORICS '15)*, Sep 2015.

[6] Jinseong Jeon, Xiaokang Qiu, Jeffrey S. Foster, and Armando Solar-Lezma. JSKETCH: Sketching for Java. In *10th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE '15)*, Sep 2015.

[7] Jinseong Jeon, Xiaokang Qiu, Armando Solar-Lezama, and Jeffrey S. Foster. Adaptive Concretization for Parallel Program Synthesis. In *the 27th International Conference on Computer Aided Verification (CAV '15)*, Jul 2015.

[8] Ravi Bhoraskar, Seungyeop Han, Jinseong Jeon, Tanzirul Azim, Shuo Chan, Jaeyeon Jung, Suman Nath, Rui Wang, and David Wetherall. Brahmastra: Driving Apps to Test the Security of Third-Party Components. In *the 23rd USENIX Security Symposium (Security '14)*, Aug 2014.

[9] Jinseong Jeon, Kristopher K. Micinski, Jeffrey A. Vaughan, Ari Fogel, Nikhilesh Reddy, Jeffrey S. Foster, and Todd Millstein. Dr. Android and Mr. Hide: Fine-grained Permissions in Android Applications. In *ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM '12)*, pages 3–14, Oct 2012.

[10] Jinseong Jeon, Sangwon Kim, and Dongkeun Lee. SAT-Based On-Line Fault Isolation in Serial Systems. In *Advances in System Testing and Validation Lifecycle (VALID '09)*, pages 25–30, Sep 2009.

[11] Yunkyung Ahn, Jinseong Jeon, and Taisook Han. Taking Environments into account for ESTEREL Program Verification. In *2007 Fall Korea Information Science Society Conference (KCC '07)*, pages 540–545, Oct 2007.

[12] Jinseong Jeon, Keoncheol Shin, and Hwansoo Han. Layout Transformations for Heap Objects Using Static Access Patterns. In *Compiler Construction (CC '07)*, pages 187–201, Mar 2007.

[13] Jinseong Jeon, Gunwoo Kim, Hwansoo Han, and Taisook Han. Practical Buffer Overrun Vulnerabilities Detection using Static Analysis. In *Korea Computer Congress (KCC '06)*, pages 391–393, Jun 2006.

Posters

[14] Jinseong Jeon. Program Synthesis for Program Analysis: Models and Drivers. In *ACM Symposium on Principles of Programming Languages (POPL '14)*, Jan 2014.

Technical Reports

[15] Jinseong Jeon, Xiaokang Qiu, Armando Solar-Lezama, and Jeffrey S. Foster. JSKETCH: Sketching for Java. Technical Report arXiv:1507.03577, CoRR, Jul 2015.

[16] Kristopher K. Micinski, Jonathan Fetter-Degges, Jinseong Jeon, Jeffrey S. Foster, and Michael R. Clarkson. Checking Interaction-Based Declassification Policies for Android Using Symbolic Execution. Technical report, CS-TR-5043, Department of Computer Science, University of Maryland, College Park, Jul 2015.

[17] Jinseong Jeon and Jeffrey S. Foster. Troyd: Integration Testing for Android. Technical report, CS-TR-5013, Department of Computer Science, University of Maryland, College Park, Aug 2012.

[18] Jinseong Jeon, Kristopher K. Micinski, and Jeffrey S. Foster. SymDroid: Symbolic Execution for Dalvik Bytecode. Technical report, CS-TR-5022, Department of Computer Science, University of Maryland, College Park, Jul 2012.

[19] Jinseong Jeon, Kristopher K. Micinski, Jeffrey A. Vaughan, Nikhilesh Reddy, Yixin Zhu, Jeffrey S. Foster, and Todd Millstein. Dr. Android and Mr. Hide: Fine-grained Security Policies on Unmodified Android. Technical report, CS-TR-5006, Department of Computer Science, University of Maryland, College Park, Dec 2011.

[20] Nikhilesh Reddy, Jinseong Jeon, Jeffrey A. Vaughan, Todd Millstein, and Jeffrey S. Foster. Application-centric Security Policies on Unmodified Android. Technical report, UCLA TR #110017, Department of Computer Science, University of California, Los Angeles, Jul 2011.