# CHA, SANG KIL

KAIST N5 #2319, 291 Daehak-ro, Yuseong-gu, Daejeon 34141, Korea

+82-(0)42-350-3569 | *sangkilc@kaist.ac.kr* | *sangkilc*

https://softsec.kaist.ac.kr/~sangkilc

## Executive Summary

Sang Kil Cha is an experimental computer scientist whose research is at the intersection of computer security and software engineering. His research mainly involves building and evaluating systems that can analyze programs for enhancing their security. He has received several awards from top-tier conferences by developing such systems. He received his Ph.D. in Electrical and Computer Engineering from Carnegie Mellon University in 2015. He is currently an associate professor at KAIST, where he leads the Software Security Lab. He is also the director of the Cyber Security Research Center at KAIST. He is currently an editorial board member of IEEE Transactions on Dependable and Secure Computing (TDSC).

## Education

2010–2015 **Ph.D. in Electrical and Computer Engineering**
*Carnegie Mellon University, Pittsburgh, PA*

Advisor: David Brumley
Thesis Title: Towards Resource-Aware Security Testing of Software
Thesis Committee: Lujo Bauer, David Molnar, and Vyas Sekar

2008–2009 **M.S. in Electrical and Computer Engineering**
*Carnegie Mellon University, Pittsburgh, PA*

2001–2008 **B.S. in Electrical and Computer Engineering**
*Korea University, Seoul, Korea*

## Research Interests

Software security. systems security, software engineering, and program analysis.

## Professional Experience

| | | |
|---|---|---|
| 2023–Present | Chief Professor of Graduate School of Information Security | Daejeon, Korea |
| 2020–Present | Director of Cyber Security Research Center, KAIST | Daejeon, Korea |
| 2020–Present | Associate Professor, KAIST | Daejeon, Korea |
| 2015–2020 | Assistant Professor, KAIST | Daejeon, Korea |
| Summer 2013 | Research Intern, Microsoft Research | Redmond, WA |
| Summer 2011 | Research Intern, Microsoft Research | Redmond, WA |

## Honors and Awards

| | | |
|---|---|---|
| 2023 | USENIX Security Distinguished Paper Award | Anaheim, CA |
| 2022 | TSE Best Paper Award | IEEE Computer Society |
| 2022 | ACM Distinguished Paper Award (*ASE* 2022) | Oakland Center, MI |
| 2022 | Test-of-Time Award (IEEE Security & Privacy) | San Francisco, CA |
| 2021 | Technology Innovation Award (College of Engineering) | Daejeon, Korea |
| 2021 | MSIT Minister Prize (Cybersecurity Education) | Seoul, Korea |
| 2020 | ACM Distinguished Paper Award (*FSE* 2020) | Sacramento, CA |
| 2019 | Best Paper Award (*NDSS BAR* 2019) | San Diego, CA |
| 2018, 2019 | MSIT Minister Prize (*Data Challenge* 2018, 2019) | Seoul, Korea |
| 2018 | Best Paper Award (*WISA* 2018) | Jeju, Korea |
| 2014 | ACM Distinguished Paper Award (*ICSE* 2014) | Hyderabad, India |
| 2013 | Student Travel Grant Award (*CCS* 2013) | Berlin, Germany |
| 2011 | Ann and Martin McGuinn Graduate Fellowship (CMU) | Pittsburgh, PA |
| 2010 | Student Travel Grant Award (*CCS* 2010) | Chicago, IL |
| 2010 | Moon-Jung Chung Scholarship (*KOCSEA* 2010) | Vienna, VA |
| 2010 | Carnegie Institute of Technology Dean's Tuition Fellowship | Pittsburgh, PA |

## Publications

### Refereed Conference and Workshop Papers

[1] Soomin Kim, Hyungseok Kim, and **Sang Kil Cha**. FunProbe: Probing Functions from Binary Code through Probabilistic Analysis. In *Proceedings of the International Symposium on Foundations of Software Engineering*, pages 1419–1430, 2023

[2] Haeun Lee, Hee Dong Yang, Su Geun Ji, and **Sang Kil Cha**. On the Effectiveness of Synthetic Benchmarks for Evaluating Directed Grey-box Fuzzers. In *Proceedings of the Asia-Pacific Software Engineering Conference*, pages 11–20, 2023

[3] Hyungseok Kim, Soomin Kim, Junoh Lee, Kangkook Jee, and **Sang Kil Cha**. Reassembly is Hard: A Reflection on Challenges and Strategies. In *Proceedings of the USENIX Security Symposium*, pages 1469–1486, 2023

[4] Tae Eun Kim, Jaeseung Choi, Kihong Heo, and **Sang Kil Cha**. DAFL: Directed Grey-box Fuzzing Guided by Data Dependency. In *Proceedings of the USENIX Security Symposium*, pages 4931–4948, 2023

[5] Minyeop Choi, Gihyuk Ko, and **Sang Kil Cha**. BotScreen: Trust Everybody, but Cut the Aimbots Yourself. In *Proceedings of the USENIX Security Symposium*, pages 481–498, 2023

[6] Haeun Lee, Soomin Kim, and **Sang Kil Cha**. Fuzzle: Making a Puzzle for Fuzzers. In *Proceedings of the IEEE/ACM International Conference on Automated Software Engineering*, pages 1–12, 2022

[7] Hyungseok Kim, Junoh Lee, Soomin Kim, SeungIl Jung, and **Sang Kil Cha**. How'd Security Benefit Reverse Engineers? The Implication of Intel CET on Function Identification. In *Proceedings of the International Conference on Dependable Systems and Networks*, pages 559–566, 2022

[8] Jaeseung Choi, Doyeon Kim, Soomin Kim, Gustavo Grieco, Alex Groce, and **Sang Kil Cha**. Smartian: Enhancing Smart Contract Fuzzing with Static and Dynamic Data-Flow Analyses. In *Proceedings of the IEEE/ACM International Conference on Automated Software Engineering*, pages 227–239, 2021

[9] Jaeseung Choi, Kangsu Kim, Daejin Lee, and **Sang Kil Cha**. NTFuzz: Enabling Type-Aware Kernel Fuzzing on Windows with Static Binary Analysis. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 1973–1989, 2021

[10] Marcel Böhme, Valentin J. M. Manès, and **Sang Kil Cha**. Boosting Fuzzer Efficiency: An Information Theoretic Perspective. In *Proceedings of the International Symposium on Foundations of Software Engineering*, pages 678–689, 2020

[11] Valentin J. M. Manès, Soomin Kim, and **Sang Kil Cha**. Ankou: Guiding Grey-box Fuzzing towards Combinatorial Difference. In *Proceedings of the International Conference on Software Engineering*, pages 1024–1036, 2020

[12] Suyoung Lee, HyungSeok Han, **Sang Kil Cha**, and Sooel Son. Montage: A Neural Network Language Model-Guided JavaScript Engine Fuzzer. In *Proceedings of the USENIX Security Symposium*, pages 2613–2630, 2020

[13] Jaeseung Choi, Joonun Jang, Choongwoo Han, and **Sang Kil Cha**. Grey-box Concolic Testing on Binary Code. In *Proceedings of the International Conference on Software Engineering*, pages 736–747, 2019

[14] HyungSeok Han, DongHyeon Oh, and **Sang Kil Cha**. CodeAlchemist: Semantics-Aware Code Generation to Find Vulnerabilities in JavaScript Engines. In *Proceedings of the Network and Distributed System Security Symposium*, 2019

[15] Minkyu Jung, Soomin Kim, HyungSeok Han, Jaeseung Choi, and **Sang Kil Cha**. B2R2: Building an Efficient Front-End for Binary Analysis. In *Proceedings of the NDSS Workshop on Binary Analysis Research*, 2019

[16] Jina Hong, JinKi Lee, HyunKyu Lee, YoonHa Chang, KwangHo Choi, and **Sang Kil Cha**. AlertVision: Visualizing Security Alerts. In *Proceedings of the World Conference on Information Security*, 2018

[17] Chanhee Lee, Changhoon Yoon, Seungwon Shin, and **Sang Kil Cha**. INDAGO: A New Framework For Detecting Malicious SDN Applications. pages 220–230, 2018

[18] SeongIl Wi, Jaeseung Choi, and **Sang Kil Cha**. Git-based CTF: A Simple and Effective Approach to Organizing In-Course Attack-and-Defense Security Competition. In *Proceedings of the USENIX Workshop on Advances in Security Education*, 2018

[19] HyungSeok Han and **Sang Kil Cha**. IMF: Inferred Model-based Fuzzer. In *Proceedings of the ACM Conference on Computer and Communications Security*, pages 2345–2358, 2017

[20] Soomin Kim, Markus Faerevaag, Minkyu Jung, SeungIl Jung, DongYeop Oh, JongHyup Lee, and **Sang Kil Cha**. Testing Intermediate Representations for Binary Analysis. In *Proceedings of the IEEE/ACM International Conference on Automated Software Engineering*, pages 353–364, 2017

[21] Weidong Cui, Marcus Peinado, **Sang Kil Cha**, Yanick Fratantonio, and Vasileios P. Kemerlis. RETracer: Triaging Crashes by Reverse Execution from Partial Memory Dumps. In *Proceedings of the International Conference on Software Engineering*, pages 820–831, 2016

[22] **Sang Kil Cha**, Maverick Woo, and David Brumley. Program-Adaptive Mutational Fuzzing. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 725–741, 2015

[23] Alexandre Rebert, **Sang Kil Cha**, Thanassis Avgerinos, Jonathan Foote, David Warren, Gustavo Grieco, and David Brumley. Optimizing Seed Selection for Fuzzing. In *Proceedings of the USENIX Security Symposium*, pages 861–875, 2014

[24] Thanassis Avgerinos, Alexandre Rebert, **Sang Kil Cha**, and David Brumley. Enhancing Symbolic Execution with Veritesting. In *Proceedings of the International Conference on Software Engineering*, pages 1083–1094, 2014

[25] Maverick Woo, **Sang Kil Cha**, Samantha Gottlieb, and David Brumley. Scheduling Black-box Mutational Fuzzing. In *Proceedings of the ACM Conference on Computer and Communications Security*, pages 511–522, 2013

[26] **Sang Kil Cha**, Thanassis Avgerinos, Alexandre Rebert, and David Brumley. Unleashing Mayhem on Binary Code. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 380–394, 2012

[27] Thanassis Avgerinos, **Sang Kil Cha**, Brent Lim Tze Hao, and David Brumley. AEG: Automatic Exploit Generation. In *Proceedings of the Network and Distributed System Security Symposium*, pages 283–300, 2011

[28] **Sang Kil Cha**, Brian Pak, David Brumley, and Richard J. Lipton. Platform-Independent Programs. In *Proceedings of the ACM Conference on Computer and Communications Security*, pages 547–558, 2010

[29] **Sang Kil Cha**, Iulian Moraru, Jiyong Jang, John Truelove, David Brumley, and David G. Andersen. SplitScreen: Enabling Efficient, Distributed Malware Detection. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation*, pages 377–390, 2010

## Refereed Journal Papers

[30] Marcel Böhme, Valentin J. M. Manès, and **Sang Kil Cha**. Boosting Fuzzer Efficiency: An Information Theoretic Perspective. *Communications of the ACM*, 66(11):89–97, 2023

[31] Dongkwan Kim, Eunsoo Kim, **Sang Kil Cha**, Sooel Son, and Yongdae Kim. Revisiting Binary Code Similarity Analysis Using Interpretable Feature Engineering and Lessons Learned. *IEEE Transactions on Software Engineering*, 49(4):1661–1682, 2023

[32] Valentin J. M. Manès, HyungSeok Han, Choongwoo Han, **Sang Kil Cha**, Manuel Egele, Edward J. Schwartz, and Maverick Woo. The Art, Science, and Engineering of Fuzzing: A Survey. *IEEE Transactions on Software Engineering*, 47(11):2312–2331, 2021

[33] Sang-Ok Park, Ohmin Kwon, Yonggon Kim, **Sang Kil Cha**, and Hyunsoo Yoon. Mind Control Attack: Undermining Deep Learning with GPU Memory Exploitation. *Computers & Security*, 102 (102115):1–16, 2021

[34] **Sang Kil Cha** and Zhenkai Liang. Asia's Surging Interest in Binary Analysis. *Communications of the ACM*, 63(4):86–88, 2020

[35] Thanassis Avgerinos, Alexandre Rebert, **Sang Kil Cha**, and David Brumley. Enhancing Symbolic Execution with Veritesting. *Communications of the ACM*, 59(6):93–100, 2016

[36] Thanassis Avgerinos, **Sang Kil Cha**, Alexandre Rebert, Edward J. Schwartz, Maverick Woo, and David Brumley. Automatic Exploit Generation. *Communications of the ACM*, 57(2):74–84, 2014

[37] **Sang Kil Cha**, Iulian Moraru, Jiyong Jang, John Truelove, David Brumley, and David G. Andersen. SplitScreen: Enabling Efficient, Distributed Malware Detection. *Journal of Communications and Networks*, 13(2):187–200, 2011

## Patents

[38] **Sang Kil Cha**, Homook Cho, Jungho Lee, and MyeongGeun Shin. Providing Method of Hacking Platform based on Open Community and Providing System Using the Same. Korea Patent 10-2384717-0000, 2022

[39] **Sang Kil Cha**, Valentin J. M. Manès, and Soomin Kim. Method and Apparatus for Grey-box Fuzzing with Distance-based Fitness Function. Korea Patent 10-2289574-0000, 2021

[40] **Sang Kil Cha** and Jiwon Choi. Technology and System for Improving the Accuracy of Binary Reassembly System with Lazy Symbolization. Korea Patent 10-2104198-0000, 2020

[41] **Sang Kil Cha**, Soomin Kim, DongYeop Oh, SeungIl Jung, and Minkyu Jung. Method and Apparatus For Testing Intermediate Language for Binary Analysis. Korea Patent 10-2117165-0000, 2020

[42] **Sang Kil Cha**, Sang-Ok Park, Ohmin Kwon, and Yonggon Kim. Mechanisms for CUDA Code Memory Page Scanning and CUDA Function Scanning. Korea Patent 10-2112929-0000, 2020

[43] **Sang Kil Cha**, SeongIl Wi, Jaeseung Choi, and HyungSeok Han. Git-based System and Method for Providing Attack and Defense Hacking Contests. Korea Patent 10-2152989-0000, 2020

[44] **Sang Kil Cha**, Soomin Kim, and Minsu Lee. System and Method for Visualizing Binary Structure. Korea Patent 10-1974069-0000, 2019

[45] Seungwon Shin, Chanhee Lee, Changhoon Yoon, and **Sang Kil Cha**. Apparatus, Method, and Computer Program for Malware Detection of Software Defined Network. Patent App. US 15811248, 2017

[46] David Brumley, **Sang Kil Cha**, Thanassis Avgerinos, and Alexandre Rebert. Detecting Exploitable Bugs in Binary Code. US Patent 9542559, 2017

[47] David Brumley, Thanassis Avgerinos, and **Sang Kil Cha**. Automated Exploit Generation. US Patent 9135405, 2015

[48] Weidong Cui, David Molnar, and **Sang Kil Cha**. Computer Memory Access Monitoring and Error Checking. Patent App. US 20130152053, 2013

# Professional Services

## Conference Program Committee

| | |
|---|---|
| 2022–2023 | ACM CCS |
| 2021–2024 | USENIX Security |
| 2024 | ICSE |
| 2024 | NDSS |
| 2020–2023 | The Web Conference (WWW) |
| 2021–2023 | ACSAC |
| 2018–2022 | ACM ASIACCS |
| 2022,2024 | ISSTA |
| 2022–2023 | FUZZING |
| 2022 | FSE Artifact |
| 2020 | IEEE Euro S&P |
| 2021–2023 | SVCC |
| 2021 | ACNS |
| 2020–2021 | SoftEng |
| 2021 | CHECKMATE |

| | |
|---|---|
| 2020 | ISSTA Doctoral Symposium |
| 2020 | ICISS |
| 2020 | LCTES |
| 2013–2021 | WISA |
| 2018 | APSys |
| 2018–2019, 2022–2023 | NDSS BAR |
| 2016, 2018 | USENIX Enigma |
| 2018 | Netsec-KR |

## Selected Talks

| | | |
|---|---|---|
| 2023 | Evaluating Fuzzers: A Grand Challenge for Software Testing Research<br>*Invited Speaker at UNIST CSE Seminar* | Ulsan, Korea |
| 2022 | Security Fundamentals<br>*Invited Speaker at Samsung MX Seminar* | Suwon, Korea |
| 2020 | Cybersecurity in Contact-free Era<br>*Invited Speaker at Korea Council of CISO* | Seoul, Korea |
| 2020 | AI-based Cyber Warfare: Its Present and Future<br>*Invited Speaker at GIST EECS Colloquium* | Gwangju, Korea |
| 2019 | Binary Analysis and Vulnerability Discovery<br>*Invited Speaker at Korea University EE Seminar* | Seoul, Korea |
| 2019 | Binary Analysis and Finding Security Vulnerabilities<br>*Invited Speaker at Korea University Seminar* | Seoul, Korea |
| 2019 | Finding Vulnerabilities in Large and Complex COTS Software<br>*Invited Speaker at Qualcomm Seminar* | San Diego, CA |
| 2018 | Git-based CTF and Automated Hacking Competition<br>*Invited Speaker at Cyber Conflict Exercise 2018* | Jeju, Korea |
| 2018 | Binary-level Vulnerability Discovery: Past, Present, and Future<br>*Invited Speaker at POSTECH CS Seminar* | Pohang, Korea |
| 2017 | Binary Analysis and Hacking<br>*Invited Speaker at Cyber Conflict Exercise 2017* | Seoul, Korea |
| 2017 | Cyberwarfare and Artificial Intelligence<br>*Advisory Panel at KTV Science Talk Show* | Sejong, Korea |
| 2017 | Automated Bug Finding<br>*Invited Speaker at Korea University Seminar* | Seoul, Korea |
| 2016 | Towards Resource-Aware Fuzzing<br>*Invited Speaker at International Symposium on Mobile Security* | Seoul, Korea |
| 2016 | AI & Security | Seoul, Korea |

| | | |
|---|---|---|
| | *Keynote Speaker at International Cyber Security Conference* | |
| 2016 | Automatic Exploit Generation<br>*Keynote Speaker at SECUINSIDE* | Seoul, Korea |
| 2016 | Towards Resource-Aware Fuzzing<br>*Invited Speaker at Samsung Seminar* | Suwon, Korea |
| 2016 | Towards Resource-Aware Fuzzing<br>*Invited Speaker at KAIST CS Colloquium* | Daejeon, Korea |

# Teaching

| | | |
|---|---|---|
| Spring 2022 | CS492E - Introduction to Software Security | KAIST |
| Fall 2021 | IS561 - Binary Code Analysis and Secure Software Systems | KAIST |
| Spring 2021 | CS220 - Programming Principles | KAIST |
| Fall 2020 | IS521 - Information Security Laboratory | KAIST |
| Spring 2020 | IS521 - Information Security Laboratory<br>CS220 - Programming Principles | KAIST<br>KAIST |
| Fall 2019 | IS561 - Binary Code Analysis and Secure Software Systems | KAIST |
| Spring 2019 | IS521 - Information Security Laboratory<br>IS511 - Introduction to Information Security (co-taught with 5 others)<br>CS220 - Programming Principles | KAIST<br>KAIST<br>KAIST |
| Fall 2018 | IS561 - Binary Code Analysis and Secure Software Systems | KAIST |
| Spring 2018 | IS521 - Information Security Laboratory<br>IS511 - Introduction to Information Security (co-taught with 5 others)<br>CS408E - Computer Science Project (mentored 1 team) | KAIST<br>KAIST<br>KAIST |
| Fall 2017 | IS561 - Binary Code Analysis and Secure Software Systems | KAIST |
| Spring 2017 | IS521 - Information Security Laboratory<br>IS511 - Introduction to Information Security (co-taught with 4 others) | KAIST<br>KAIST |
| Fall 2016 | IS893 - Advanced Software Security | KAIST |
| Spring 2016 | IS561 - Binary Code Analysis and Secure Software Systems | KAIST |

# Advising

## Current Students

| | | |
|---|---|---|
| 2018.3– | Soomin Kim | Ph.D. |

| | | |
|---|---|---|
| 2019.9– | Hyungseok Kim | Ph.D. |
| 2021.8– | JungHyun Kim | Ph.D. |
| 2023.3– | HyungJoon Yoon | M.S. |
| 2023.3– | Sanghyun Park | M.S. |
| 2023.3– | Jungwoo Lee | M.S. |
| 2023.8– | Steve Gustaman | M.S. |
| 2023.8– | Geonwoo Park | M.S. |

## Selected Grants

| | | |
|---|---|---|
| 2021– | Developing Next-Generation Binary Decompiler<br>2,325,000,000 KRW from IITP | PI |
| 2018 | Developing JavaScript Mutation Algorithm for Fuzzing<br>50,000,000 KRW from LIG | PI |
| 2018–2019 | Model-based Fuzzing for Finding Kernel Vulnerabilities<br>180,000 USD (90,000 USD from ONRG and 90,000 USD from ONR) | PI |
| 2017 | Binary Smart Fuzzing<br>98,000,000 KRW from Samsung | PI |
| 2017 | Next-Generation SIEM Development<br>60,000,000 KRW from AhnLab | PI |
| 2017–2018 | Cyber Immune System<br>225,000,000 KRW from IITP | PI |
| 2017 | Smart Fuzzing for Emerging Vehicle Platform<br>50,000 USD from Ford Motor Company | PI |
| 2016 | Search-based Fuzzing<br>37,000,000 KRW from KAIST | PI |
| 2016–2018 | Building a Platform for Automated Reverse Engineering and Vulnerability Detection with Binary Code Analysis<br>1,510,000,000 KRW from IITP | PI |
| 2015–2018 | Discovering Vulnerabilities with Binary Code Analysis<br>100,000,000 KRW from KAIST | PI |