

Jinbum Park

Email: jinb.park7@gmail.com

South Korea

Affiliation: [Samsung Research, Security & Privacy Team](#)

Website: <https://jinb-park.github.io>

Blog: <https://jinb-park.github.io/blog>

Google Scholar: <https://scholar.google.com/citations?user=e-o2O2IAAAAJ>

Research Interest

- Trusted execution environments (TrustZone, ARM CCA, SGX, Secure processors)
- Confidential computing
- OS kernel security
- Side-channel attacks and defenses
- Bug finding and exploitations
- Machine learning security (e.g., federated learning, LLM security)
- Applied cryptography (e.g., Zero Knowledge Proof)

Education

2006 – 2013 **BSc in Department of Software**, Gachon University, South Korea

Projects

All projects listed below were done in Samsung Research.

2023.04 – on **Islet: An on-device confidential computing platform**

- **Role.** A developer and researcher
- **Type.** Open source project ([an official Confidential Computing Consortium \(CCC\) project](#))
- Developing a whole software stack, fully written in Rust, to power ARM CCA. (based on the ARM CCA specification)
- Developed an end-to-end confidential AI demo scenario (for details, see [here](#)).
- Implemented an integration with [the certifier framework](#) to build an end-to-end heterogeneous CC (Confidential Computing) protection.
- An academic research towards privacy-preserving CC framework (work in progress internally as a leading author).

2022.05 – 2023.04 a period of time for parental leave

2021 – 2022 A federated learning framework for mobile devices

- **Role.** Lead developer
- **Type.** Proof-of-concept project (not deployed in production)
- Developed an android based (Java) on-device federated learning framework built on top of a TensorFlowLite library modified to be able to do training on devices.
- Developed a federated learning server (Python) that communicates with devices through gRPC.
- Did a field test with 20 android devices on a location-based service deep learning model.

2020 – 2021 Rust-based full-stack OS for secure processor

- **Role.** Lead kernel developer and one of the application layer developers
- **Type.** In development while aiming to be in production (but not yet released)
- Developed a Rust-based kernel from scratch, which targets ARM Cortex-M boards and doesn't rely on Rust's std library.
- Developed an application layer (a set of system calls and libraries) and an async backend that allows applications to use Rust's async capability.

2019 – 2020 A TrustZone-based secure enclave

- **Role.** Lead developer (one-man project)
- **Type.** Proof-of-concept project (not deployed in production)
- Designed and developed an SGX-like enclave architecture on top of ARM TrustZone, thereby allowing mobile developers to take SGX's programming model. (Rust and C++)
- Developed a new small Rust compiler toolchain for this architecture.

2018 – 2019 A real-time kernel protection

- **Role.** One of the core developers
- **Type.** Developed for autonomous platforms but not deployed
- Designed and developed a Type-1 hypervisor on ARMv8-A, which ensures that Linux's non-writable memory regions are not corrupted. This is similar in concept to [KNOX RKP](#) in galaxy devices.
- Written in C and ARM assembly.








2014 – 2017 System Integrity Monitor (SIM) version 1.0–3.0





- **Role.** Lead developer
- **Type.** Deployed as the key part of [GAIA](#) which is Samsung SMART TV's security solution.
- Designed and developed a Linux kernel monitoring system that utilizes ARM TrustZone and a proprietary memory bus snooping system. It aims to prevent and detect corruptions on non-writable memory regions and security-critical kernel read-write data. Also, it plays a crucial role in the secure boot and attestations of Samsung SMART TVs. (C and C++)
- Developed device drivers for Linux kernel and TrustZone secure kernel. (C and ARM assembly)
- Developed a daemon service that runs as a system service of the Tizen TV platform and takes local/remote attestation requests from other processes. (C++)
- Designed PKI (Public Key Infrastructure) and cryptographic key operations for this system.
- Designed attestation servers and supported server developers.

2013 – 2014 **Samsung DRM (SDRM)**







- **Role.** Associate developer
- **Type.** Deployed in Samsung SMART TVs to protect 4k contents.
- Migrated the existing SDRM codes into ARM TrustZone. (C and C++)
- Developed the SDRM media plugin for the Tizen TV platform.
- Managed PKI (Public Key Infrastructure) and cryptographic key operations for this system.






Publications

- 2025 EdgeShield: A Security Monitor Framework For On-Device Confidential Computing
- **Jinbum Park**, Bokdeuk Jeong, Sunwook Eom, Taesoo Kim
 - Under submission
- 2025 TikTag: Breaking ARM's Memory Tagging Extension with Speculative Execution 
- Juhee Kim, **Jinbum Park**, Sihyeon Roh, Jaeyoung Chung, Youngjoo Lee, Taesoo Kim, Byoungyoung Lee
 - IEEE Security and Privacy 2025 (*top-tier conference*) (to appear)
- 2024 PeTAL: Ensuring Access Control Integrity against Data-only Attacks on Linux 
- Juhee Kim, **Jinbum Park**, Yoochan Lee, Chengyu Song, Taesoo Kim, Byoungyoung Lee
 - ACM CCS 2024 (*top-tier conference*)
- 2022 In-Kernel Control-Flow Integrity on Commodity OSES using ARM Pointer Authentication  
- Sungbae Yoo(*), **Jinbum Park**(*), Seolheui Kim, Yeji Kim, Taesoo Kim (*: co-leading authors)
 - The 31st USENIX Security Symposium (USENIX Security 2022) (*top-tier conference*)
- 2022 ViK: Practical Mitigation of Temporal Memory Safety Violations through Object ID Inspection 
- Haehyun Cho, **Jinbum Park**, Adam Oest, Tiffany Bao, Ruoyu Wang, Yan Shoshitaishvili, Adam Doupe, Gail-Joon Ahn
 - The 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '22) (*top-tier conference*)
- 2020 Exploiting Uses of Uninitialized Stack Variables in Linux Kernels to Leak Kernel Pointers  
- Haehyun Cho, **Jinbum Park**, Joonwon Kang, Tiffany Bao, Ruoyu Wang, Yan Shoshitaishvili, Adam Doupe, Gail-Joon Ahn
 - The 14th USENIX Workshop on Offensive Technologies (WOOT '20)



- 2020 SmokeBomb: Effective Mitigation Method against Cache Side-channel Attacks on the ARM Architecture  
- Haehyun Cho, **Jinbum Park**, Donguk Kim, Ziming Zhao, Yan Shoshitaishvili, Adam Doupe, Gail-Joon Ahn
 - The 18th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys 2020) (*top-tier conference*)
- 2018 Prime+Count: Novel Cross-world Covert Channels on ARM TrustZone  
- Haehyun Cho, Penghui Zhang, Donguk Kim, **Jinbum Park**, Choong-Hoon Lee, Ziming Zhao, Adam Doupe, and Gail-Joon Ahn
 - Annual Computer Security Applications Conference (ACSAC) 2018
- 2016 A Snoop-Based Kernel Introspection System against Address Translation Redirection Attack
- Donguk Kim, Jihoon Kim, **Jinbum Park**, Jinmok Kim
 - Journal of The Korea Institute of Information Security & Cryptology VOL.26, NO.5, Oct. 2016
- 2015 An Efficient Kernel Introspection System using a Secure Timer on TrustZone
- Jinmok Kim, Donguk Kim, **Jinbum Park**, Jihoon Kim, Hyounghshick Kim
 - Journal of The Korea Institute of Information Security & Cryptology VOL.25, NO.4, Aug. 2015

Talks (industry conferences)

- 2024 On-Device Confidential Computing: Updates on Our Activities and Future Potential 
- **Jinbum Park**, Heeill Wang
 - Samsung Security Tech Forum 2024 (SSTF 2024)
- 2024 Breaking ARM MTE with Speculative Execution
- **Jinbum Park**
 - Zer0Con 2024
- 2022 Taking Kernel Hardening to the Next Level  
- **Jinbum Park**, Haehyun Cho, Sungbae Yoo, Seolheui Kim, Yeji Kim, Bumhan Kim, Taesoo Kim
 - Blackhat ASIA 2022
- 2020 Cache Attacks on Various CPU Architectures  
- **Jinbum Park**
 - POC 2020
- 2019 Micro-architectural attack and defense on Linux kernel 
- **Jinbum Park**, Joonwon Kang
 - Samsung Open Source Conference (SOSCON) 2019

- 2019 Leak kernel pointer by exploiting uninitialized uses in Linux kernel  
 - **Jinbum Park**
 - Zer0Con 2019
- 2018 Attack and Defense on Linux kernel 
 - **Jinbum Park**
 - Samsung Open Source Conference (SOSCON) 2018
- 2018 Exploit Linux kernel eBPF with side-channel  
 - **Jinbum Park**
 - KIMCHICON 2018

Open sources

- KSPP Study: Analysis on Kernel Self-Protection: Understanding Security and Performance Implication 
 - Analyzed security and performance analysis for kernel self-protection projects
- CSCA: Crypto Side Channel Attack 
 - Developed cache-based crypto side-channel attacks on both x86_64 and ARM64 (e.g., recovering a full AES-128 key)
- Linux kernel contributions (selected)
 - Fix vulnerable gadgets to spectre-variant1 attack (patch [0,1](#))
 - arm: Makes ptdump reusable and add WX page checking ([patch](#))
 - arm: Add ARCH_HAS_FORTIFY_SOURCE (patch [0,1](#))
- Ubuntu kernel contributions
 - Revert barrier-patch which turns out be vulnerable to variant4 attack (patch [0,1](#))

Skills

Languages.

- Korean, English

Programming Languages.

- C, C++, Python, Rust, Assembly (x86_64 and ARM)

Hardware.

- ARM: ARM Cortex-A, ARM Cortex-M, ARM TrustZone, ARM CCA, ARM pointer authentication, ARM memory tagging extension
- Intel: x86_64, SGX
- Developed several security-relevant arch-specific codes and cache attacks/defenses on both architectures.

Low-level software.

- Kernel: Linux, FreeBSD
- Hypervisor: KVM, a light-weight security monitor (e.g., RMM in ARM CCA)

Compiler.

- LLVM, GCC (developed several static analysis passes on LLVM and GCC)

Domain knowledge.

- System and software security
- Operating system kernel and hardware architectures
- Offensive techniques (kernel exploits and bug findings)
- Mobile platforms (Tizen and Android)
- Applied cryptography
- Machine learning and deep learning
- Zero-Knowledge Proof