**Physical Model-Assisted Secure Software Execution   
Against Fault-Injection Attacks**

Mridha Md Mashahedur Rahman, Arunabho Basu, Pantha Protim Sarker, Dr Rakibul Hassan,   
Dr Mark Tehranipoor, Dr Farimah Farahmandi

*Electrical and Computer Engineering Dept,*

*University of Florida, Gainesville, Florida, 32611*

*Contact : mrahman1@ufl.edu*

**Abstract**— Modern integrated circuits, like Systems-on-Chip (SoCs) and micro-processors, are increasingly being targeted by both cyber and physical attacks. Software running on compromised hardware is not secure and often requires expensive hardware recall or redesign. While major strides have been made in mitigating cyberattacks through secure software development, compilation, and execution practices; physical attacks such as fault-injection attacks exploit hardware vulnerabilities, thereby bypassing software level protections. Given physical access to the target device, these attacks are highly effective in compromising the confidentiality, integrity, and availability of modern computing systems. Since the hardware is already fabricated, the cost of redesign to address physical vulnerabilities is very high. Thus, software and firmware based counter-measures against physical vulnerabilities provide a more cost-effective and adaptable solution. However, existing compiler frameworks lack built-in considerations for physical threats and do not implement targeted hardening techniques against physical attacks. In this work, we analyze physical threats from fault-injection at the software compilation stage. Subsequently, we model the attack based on physical design characteristics which can be addressed at compilation stage. Our proposed approach enables secure software compilation by integrating low-cost, physically aware hardening techniques, effectively enhancing system resilience against fault-injection attacks.

*Index Terms—*fault injection; physical attacks; fault model; compiler; software countermeasures

References

[1] GCC, the GNU Compiler Collection, “Program instrumentation options”, https://gcc.gnu.org/onlinedocs/gcc/Instrumentation-Options.html.

[2] L. Dureuil, G. Petiot, M.-L. Potet, T.-H. Le, A. Crohen, and P. de Choudens, “FISSC: a fault injection and simulation secure collection,” in SAFECOMP 2016 - The 35th International conference on Computer Safety, Reliability and Security, vol. 9922, 2016, pp. 3–11.

[3] T. Farheen, S. Tajik, and D. Forte, “Spred: Spatially distributed laser fault injection resilient design,” in 2023 24th International Symposium on Quality Electronic Design (ISQED), 2023, pp. 1–8.

[4] F. Lu, G. D. Natale, M.-L. Flottes, B. Rouzeyre, and G. Hubert, “Layout aware laser fault injection simulation and modeling: From physical level to gate level,” in 2014 9th IEEE International Conference on Design Technology of Integrated Systems in Nanoscale Era, 2014, pp. 1–6.