International Journal of Parallel Programming (2019) 47:535–537
https://doi.org/10.1007/s10766-019-00636-z

EDITORIAL

Editorial: Special Issue on Side-Channel and Fault Analysis of High-Performance Computing Platforms

Nahid Farhady Ghalaty[1]

Published online: 20 April 2019

Classically, Side Channel Attack (SCA) and Fault Attack (FA) are the two passive and active forms of physical attacks geared toward secret key extraction and access control mechanisms bypass. In the past decade, side channel attacks have been more on the center of attention, due to the practicality of launching the attack. More recently, due to new forms of such attacks applicable beyond the domain of cryptography and cryptographic hardware, SCA and FA have grown to be a threat for the semiconductor industry [1–4].

In SCA for cryptography, the adversary is an observer of the physical information leaked during the execution of a target cryptographic software or hardware. This information could be leaked via data dependent fluctuation of power, voltage, temperature, electromagnetic radiations, execution time. The adversary correlates the observations to the operations executed in the target to reveal assets or other confidential information, e.g., secret keys. In FA for cryptography, the adversary has very accurate mathematical assumptions on the fault model. The fault model is the mathematical assumption on the effect of fault injection on the target, on which the accuracy and success of the fault attack post-processing step depends. Due to all these assumptions without considering the fault injection tool, the target device and its accuracy, fault attacks have been behind in the race of security threats. However, over the past decade, an effort began towards more practical fault attacks. Examples of these efforts are Fault Sensitivity Analysis [5], Differential Fault Intensity Analysis (DFIA) [6], and DERA [7]. These attacks tried to gradually eliminate mathematical assumptions from the fault attack process, making it more practical. As an example, DFIA considered the natural behavior of the circuit under fault injection as expected behavior for further post-processing. These attacks rely on the concept of fault biasness. As a result of the above-mentioned efforts, FA too have become a viable threat and several research ideas towards countermeasures have been also progressed over the past years. The research community have gone through several phases for physical attacks.

✉ Nahid Farhady Ghalaty
farhady@gwu.edu

[1] Department of Computer Science, The George Washington University, Washington, DC, USA

Two categories of mitigation techniques, masking and hiding countermeasures, have been proposed to mitigate the problem of SCA for cryptography. These countermeasures are effective since they tend to remove the correlation and bond between the secret values and the leaked information. The reason behind such an attack is non-uniformity and the fact that any changes in the side channel leakage (more in general, in the side channel) reveal a change in the data values. By using Masking and Hiding techniques, we can bring some randomness into the picture to break this bond. The countermeasures against FA mostly rely on redundancy [8]. There are several forms of redundancy such as spatial redundancy, time redundancy, etc. The duplication in this case can be applied in different levels in hardware and software. In case of embedded software, it can be applied to algorithm level, or instruction level. A major problem with redundancy techniques at higher levels of abstraction is that due to the advances in technology, the fault injection devices are accurate enough that are capable of bypassing the redundancy by injecting similar faults in the original and the redundant copy. As a consequence, the countermeasures are mostly moving towards more granular redundancy. Recent publications propose granular techniques both at software and hardware level. Conor et al. proposed to use bit-slicing on LED algorithm as a countermeasure against fault attacks using intra-instruction redundancy [9]. Their proposed technique also reduces the performance overhead compared to instruction duplication. In another effort on the embedded software, Chen et al. have proposed using vectorization in modern microprocessors [10]. The benefit of their proposed method is that it is automated as an independent LLVM pass. Since the proposed technique is using vectorization, it will much difficult for the adversary to be able to apply same fault on one line of data.

Observing this long journey of countermeasures against fault attacks and side channel attacks, we have come a long way. But we are still not at the destination. On one hand, based on my research, the next focus must be on high performance combined countermeasures considering the applications. On the other hand, more work has to be done in terms of mitigation in the space of SCA and FA beyond cryptography and cryptographic hardware. Specifically, high-performance architectures and their compilers traditionally focus on maximizing the system performance through micro-architectural optimizations such as pipelining, caches, and hyper-threading, bootstrapped by the corresponding compilation and run-time supports. However, such performance considerations introduce a stochastic behavior in the execution time which statistic is biased toward data and control dependent operations in the micro-architecture. Once the biases and their dependency on the instruction and data are observed by an adversary and exploited, critical or secret information can be leaked, and the system confidentiality and security will be subverted. The problem does not exclusively affect cryptography and paves the way for new, remote attack vectors—a huge departure from the classic SCA and FA setting. For the first time, this Special Issue of IJPP gathers leading contributions on modern SCA and FA for high-performance computing. The contributions bring to the attention of the high-performance computing and parallelizing compilers challenges and opportunities in both attack and defense mechanisms in the context of side channel, covert channel and fault analysis applied to modern computing systems. The Special Issue Contributions includes examples of attacks and prospective defenses to high-performance processors, instruction set

extensions, deep memory hierarchy, and FPGA embodiments, from initial contributions appeared in top conferences in security/side-channel analysis, such as CT-RSA, HOST and high-performance computing such as HPCA. This Special Issue also calls for an urgent and synergic approach with both the security and high-performance computing communities, to come together and address the emerging challenges related to SCA and FA on modern systems.

References

1. SIA SRC Vision Report. https://www.semiconductors.org/resources/semiconductor-research-opportunities-an-industry-vision-and-guide-2/sia-src-vision-report-3-30-17-2/
2. Kocher, P., et al.: Spectre Attacks: Exploiting Speculative Execution. CoRR abs/1801.01203 (2018)
3. Inci, M.S., et al.: Cache Attacks Enable Bulk Key Recovery on the Cloud. In: CHES, pp. 368–388 (2016)
4. Yuce, B., et al.: Fault attacks on secure embedded software: threats, design, and evaluation. J. Hardw. Syst. Secur. **2**(2), 111–130 (2018)
5. Li, Y., et al.: Fault sensitivity analysis. In: International Workshop on Cryptographic Hardware and Embedded Systems, pp. 320–334. Springer, Berlin (2010)
6. Ghalaty, N.F., et al.: Differential fault intensity analysis. In: 2014 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC), pp. 49–58. IEEE (2014)
7. Liu, Y., et al.: DERA: yet another differential fault attack on cryptographic devices based on error rate analysis. In: Proceedings of the 52nd Annual Design Automation Conference, p. 31. ACM (2015)
8. Barenghi, A., et al.: Countermeasures against fault attacks on software implemented AES: effectiveness and cost. In: Proceedings of the 5th Workshop on Embedded Systems Security, p. 7. ACM (2010)
9. Conor, P., et al.: Lightweight fault attack resistance in software using intra-instruction redundancy. In: International Conference on Selected Areas in Cryptography, pp. 231–244. Springer, Cham (2016)
10. Chen, Z., et al.: CAMFAS: a compiler approach to mitigate fault attacks via enhanced SIMDization. In: 2017 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC), pp. 57–64. IEEE (2017)