Research work:
Modeling Control-Flow Attackers of a Secure Cryptoprocessor

Supervisors: Sylvain Boulmé and Marie-Laure Potet

mailto:Sylvain.Boulme@univ-grenoble-alpes.fr

Our pervasive computing world (with Internet of things, Cyber-physical systems, etc) makes us very vulnerable to computer attacks. In particular, *control-flow attacks* makes the chip run another code than the one intended by its designer (for instance, see Return-oriented programming as detailed in Buchanan et al. [2008]). Control-flow attacks may exploit software weaknesses (e.g. a lack of protection against buffer overflows), or they may also "create" the weakness by hardware fault injections Bukasa et al. [2018].

Several techniques, under the umbrella of Control-Flow Integrity (CFI) defined by Abadi et al. [2005a], have been proposed to detect and prevent such attacks, by monitoring program branching behaviors. In particular, we consider a processor, called "IntrinSec", and designed by Oliver Savry and his team at CEA-Leti. IntrinSec provides CFI from instruction-level encryption, following ideas of a previous experiment described in Hiscock et al. [2019].

In order to ensure CFI, IntrinSec requires that the compiler generate some special instructions before each branching instructions. At Verimag, Sylvain Boulmé and Paolo Torrini have implemented these modifications on the CompCert compiler[1]. CompCert is actually a *certified* compiler, meaning that the functional correctness of the compiler is mathematically proved, and that this proof is automatically checked within a proof assistant (here Coq). While, the functional correctness of our compiler is still a proof in progress, they have already defined a functional model of IntrinSec and of its special instructions to protect the control-flow.

This research work will tackle the issue of evaluating the protection of IntrinSec against control-flow attacks. This requires to define *models* of control-flow attackers, such as de Clercq and Verbauwhede [2017], Potet et al. [2014] or Jacomme et al. [2017]. Hence, the goal of this work is to define/adapt such models in order to evaluate CFI of IntrinSec. For example, following ideas of Abadi et al. [2005b], can we extend our CompCert model of IntrinSec in order to include models of attackers? Can we prove that these attacks make the processor aborts? Indeed, ideally, we expect to be able to mathematically prove that some kinds of attacks are not possible (or difficult). Such mathematical proofs are now very frequent on security protocols : for instance, see Scerri [2015].

[1] http://compcert.inria.fr/

References

M. Abadi, M. Budiu, Ú. Erlingsson, and J. Ligatti. Control-flow integrity. In *Proceedings of the 12th ACM Conference on Computer and Communications Security, CCS 2005, Alexandria, VA, USA, November 7-11, 2005*, pages 340–353. ACM, 2005a. doi: 10.1145/1102120.1102165. URL https://doi.org/10.1145/1102120.1102165.

M. Abadi, M. Budiu, Ú. Erlingsson, and J. Ligatti. A theory of secure control flow. In *Formal Methods and Software Engineering, 7th International Conference on Formal Engineering Methods, ICFEM 2005, Manchester, UK, November 1-4, 2005, Proceedings*, volume 3785 of *Lecture Notes in Computer Science*, pages 111–124. Springer, 2005b. URL https://users.soe.ucsc.edu/~abadi/Papers/cfitheory_submit.pdf.

E. Buchanan, R. Roemer, H. Shacham, and S. Savage. When good instructions go bad: generalizing return-oriented programming to RISC. In *Proceedings of the 2008 ACM Conference on Computer and Communications Security, CCS 2008, Alexandria, Virginia, USA, October 27-31, 2008*, pages 27–38. ACM, 2008. doi: 10.1145/1455770.1455776. URL https://doi.org/10.1145/1455770.1455776.

S. Bukasa, L. Claudepierre, R. Lashermes, and J.-L. Lanet. When fault injection collides with hardware complexity. In *FPS 2018 - 11th International Symposium on Foundations & Practice of Security*, pages 1–16, Montréal, Canada, Nov. 2018. URL https://hal.inria.fr/hal-01950931.

R. de Clercq and I. Verbauwhede. A survey of hardware-based control flow integrity (CFI). *CoRR*, abs/1706.07257, 2017. URL http://arxiv.org/abs/1706.07257.

T. Hiscock, O. Savry, and L. Goubin. Lightweight instruction-level encryption for embedded processors using stream ciphers. *Microprocessors and Microsystems - Embedded Hardware Design*, 64:43–52, 2019. doi: 10.1016/j.micpro.2018.10.001. URL https://doi.org/10.1016/j.micpro.2018.10.001.

C. Jacomme, S. Kremer, and G. Scerri. Symbolic models for isolated execution environments. *IACR Cryptology ePrint Archive*, 2017:70, 2017. URL http://eprint.iacr.org/2017/070.

M.-L. Potet, L. Mounier, M. Puys, and L. Dureuil. Lazart: A Symbolic Approach for Evaluation the Robustness of Secured Codes against Control Flow Injections. In *Seventh IEEE International Conference on Software Testing, Verification and Validation*, Cleveland, United States, Mar. 2014. doi: 10.1109/ICST.2014.34. URL http://hal.univ-grenoble-alpes.fr/hal-01229274.

G. Scerri. *Proofs of security protocols revisited. (Les preuves de protocoles cryptographiques revisitées)*. PhD thesis, École normale supérieure de Cachan, France, 2015. URL https://tel.archives-ouvertes.fr/tel-01133067.