Improving the Design and Evaluation of Cryptographic Implementations against Leakage

Can Open Source Help and How?

François-Xavier Standaert*

June 6, 2022

Two decades after the publication of the first side-channel attacks by Kocher et al. [6, 7], the definition of what is a secure implementation remains a topic of intense discussion. We posit that the reason of this situation lies in the origin of the attacks, which took both the cryptographic research community and the embedded security industry by surprise. Without good formal solutions to prevent them, the industry first reacted by combining countermeasures with a certain level of security by obscurity. As a result, certification schemes have been established in order to try characterizing the "practical" security of a product. Defining the practicality of an adversary is hard because practicality is a somewhat subjective notion which tends to change over time [1]. In parallel, researchers started to understand the most powerful attacks paths and to identify countermeasures to circumvent them. Progresses over the years made clear that security without obscurity is possible: we refer to [8] for a technical overview. As a result, academic works have increasingly utilized a definition of worst-case adversary in their evaluations, which aims to bound the measurement and computational limits of the adversary rather than its practicality, and is therefore more in line with cryptographic research. Among other benefits, this approach comes with the possibility to extend the cryptographic separation of duties between unambiguous assumptions on primitives and security reductions from complex schemes to simpler assumptions.

In general, such a maturity in security and cryptographic research indicates a point in time where standardization can be beneficial. However, while standardization has been successfully applied to cryptographic algorithms multiple times, generalizing this approach to implementations raises additional challenges. First, side-channel security has to rely on physical assumptions which, to be connected with cryptographic reductions, must be unambiguous and quantifiable. Typical examples include limiting the side-channel signal (e.g., in the case of leakage-resilient constructions [5]) or ensuring a sufficient level of noise in the leakage for the masking countermeasure [4]. This excludes the (hard to define) secrecy of an implementation as a valid assumption, and therefore requires an evolution in certification practices that value such a secrecy qualitatively. Second, physical assumptions are technology-dependent and must be re-evaluated with technological progresses.

In this talk, I will (1) argue that taking full advantage of research progresses in embedded security through (ideally standardized) countermeasures may strongly benefit from open source implementations maintained and publicly evaluated over time, and (2) describe a model of development that can serve such purposes by complementing the industrial ecosystem rather than competing with it, enabling a gradual integration of open source solutions when they become sufficiently stable

* Crypto Group, ICTEAM Institute, UCLouvain, Belgium.

over time. On the one hand, it is expected that combining the longer-term quantitative evaluations that open source designs enable with shorter-term certifications to assess their integration will give rise to stronger technological building blocks in a foreseable future. On the other hand, it is expected that identifying some open source designs as practically-relevant targets can serve as a constructive interface between academic research and industrial developments, limiting the need of hardly productive discussions about research being unpractical and, as a result, the need to target deployed products as a counter-argument (with all the responsible disclosure issues that it raises).

Note that while the examples in the talk will primarily focus on side-channel security, the general ideas put forward could be applicable to other physical (e.g., fault) attacks [3, 2].

References

[1] M. Azouaoui, D. Bellizia, I. Buhan, N. Debande, S. Duval, C. Giraud, É. Jaulmes, F. Koeune, E. Oswald, F. Standaert, and C. Whitnall, *A systematic appraisal of side channel evaluation strategies*, in SSR, vol. 12529 of Lecture Notes in Computer Science, Springer, 2020, pp. 46–66.

[2] E. Biham and A. Shamir, *Differential fault analysis of secret key cryptosystems*, in CRYPTO, vol. 1294 of Lecture Notes in Computer Science, Springer, 1997, pp. 513–525.

[3] D. Boneh, R. A. DeMillo, and R. J. Lipton, *On the importance of checking cryptographic protocols for faults (extended abstract)*, in EUROCRYPT, vol. 1233 of Lecture Notes in Computer Science, Springer, 1997, pp. 37–51.

[4] A. Duc, S. Faust, and F. Standaert, *Making masking security proofs concrete (or how to evaluate the security of any leaking device), extended version*, J. Cryptol., 32 (2019), pp. 1263–1297.

[5] S. Dziembowski and K. Pietrzak, *Leakage-resilient cryptography*, in FOCS, IEEE Computer Society, 2008, pp. 293–302.

[6] P. C. Kocher, *Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems*, in CRYPTO, vol. 1109 of Lecture Notes in Computer Science, Springer, 1996, pp. 104–113.

[7] P. C. Kocher, J. Jaffe, and B. Jun, *Differential power analysis*, in CRYPTO, vol. 1666 of Lecture Notes in Computer Science, Springer, 1999, pp. 388–397.

[8] F. Standaert, *Towards and Open Approach to Secure Cryptographic Implementations (Invited Talk)*, in EUROCRYPT (1), vol. 11476 of Lecture Notes in Computer Science, Springer, 2019, p. xv.