TUTORIAL: How to Digitally Construct and Validate TRNG and PUF Primitives which are based on Physical Phenomenon ?

Jean-Luc Danger
Full Professor at Telecom ParisTech, University Paris-Saclay
Scientific Advisor at Secure-IC

September 23, 2017

1 Abstract

In digital devices, the cryptographic functions are dependant on peripheral primitives, like the True Random Number Generation (TRNG) and Physically Unclonable Function (PUF) which generates a random number and an identifier respectively. The source of these primitives is not defined by a digital algorithm but comes from physical phenomenon, notably the noise. Consequently a conversion is necessary to output a digital random number or identifier. Indeed, these two types of primitives exploit the noise, but at different stage. At the manufacturing stage, the variance of the manufacturing process creates mismatches between transistors. These slight differences are fixed once the chip is fabricated, they should be transformed by the PUF to a digital variable when an identifer is called by the application. When the chip is in used, the environmental noise is extracted by the TRNG to generate a digital random number. In case of PUF, we can say that the entropy is "static", whereas the entropy for the TRNG is "dynamic". The dynamic entropy is a major problem for the PUF which is natively not steady because of the environmental noise. The TRNG is very sensitive to an external noise, which can be malevolently generated by an attacker, and can bias the TRNG output. Consequently, it is necessary to add to the primitives an evaluation or correction block to detect or enhance their behavior. This means that some tests and metrics have to be be specified to define what is a good identifier and a good random number.

We will see in this tutorial, the different constructions of PUF and TRNG, but also the methods to validate their quality to ensure a minimum level of trust.