# Leakage Assessment Report for Xoodyak_R3_first_order

Cankun Zhao Hang Zhao Bohan Yang Wenping Zhu Leibo Liu

May 11, 2022

1. Target implementation

   (a) Algorithm: **Xoodyak.**

   (b) Team: **Ruhr-University Bochum, Germany.**

   (c) Variant name: **Xoodyak_R3_first_order.**

   (d) URL: https://github.com/Chair-for-Security-Engineering/LWC-Masking/tree/main/Xoodyak/Xoodyak_R3_first_order.

   (e) GitHub commit hash: **4244b255e282e2d309aa270960bcd5a594c2db03.**

   (f) Protection method: **Hardware Private Circuits 2.**

   (g) Protection order: **1.**

   (h) Note: **A typo in line 104 of "xoodoo.vhd" should be fixed before assessment as discussed in the Issue.**

2. Experimental setup

   (a) Measurement platform and device-under-evaluation: **Design-under-evaluation was instantiated on the Xilinx Spartan-6 (XC6SLX75-2CSG484C) FPGA on SAKURA-G board. The other Xilinx Spartan-6 (XC6SLX9-2CSG225C) FPGA on SAKURA-G was used for control.**

   (b) Description of measurements: **The design-under-evaluation power consumption is measured at the output of the SAKURA-G's on-board amplifier (AD8000YRDZ), that amplifies the voltage drop across the on-board 1 Ω shunt resistor.**

   (c) Usage of bandwidth limiters, filters, amplifiers, etc. and their specification: **N/A.**

   (d) Frequency of operation: **3 MHz.**

   (e) Oscilloscope and its major characteristics: **Teledyne LeCroy WaveRunner 8404M with 4 GHz bandwidth was used to collect traces.**

   (f) Sampling frequency and resolution: **Sampling rate of 100 MS/s and 8-bit sample resolution were used.**

   (g) Are sampling clock and design-under-evaluation clock synchronized? **No.**

3. Leakage assessment characteristics

   (a) Leakage assessment type: **Fixed vs. random t-test at first order** [GGR11].

   (b) Number of traces used: **100,000.**

   (c) Data inputs and performed operations: **Tested operation is the Xoodoo permutation with 12 rounds. Input test vectors are initially shared on the control FPGA.**

   (d) Source of random and pseudorandom inputs: **Trivium-based DRBG.**

   (e) Trigger location relative to the execution start time of the algorithm: **Scope trigger is set at the beginning of the Xoodoo permutation.**

   (f) Time required to collect data for a given leakage assessment: **About 9 minutes.**

   (g) Total time of the assessment: **About 10 minutes.**

   (h) Total size of all traces: **766 MB.**

   (i) Availability of raw measurement results: **Per request.**

4. Results of leakage assessment

   (a) Graphs illustrating the obtained results: **The t-test result is shown in Figure 1. The raw waveform of power consumption is provided in Figure 2. A simulation waveform of a Xoodoo permutation is provided in Figure 3 to help understand the power consumption and the leakage.**

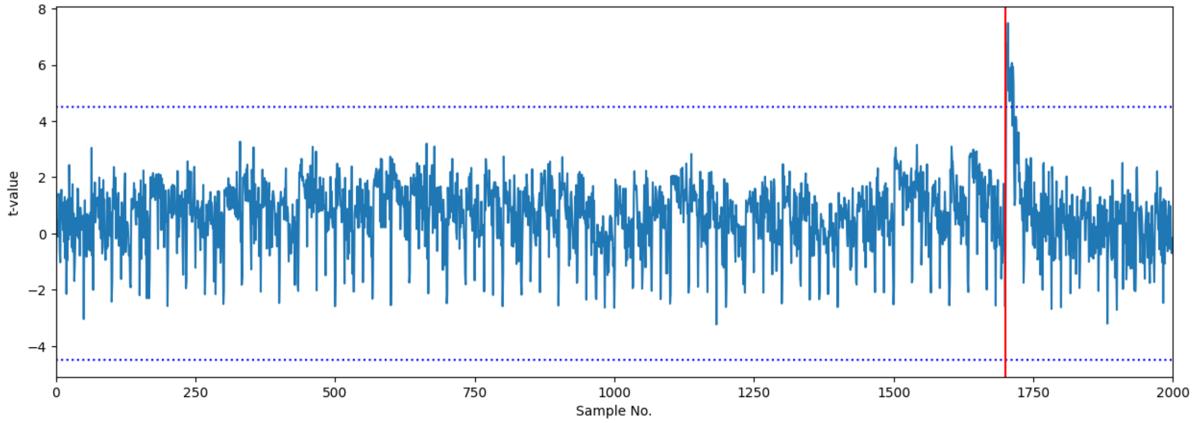

Figure 1: First-order t-test results (100,000 traces).

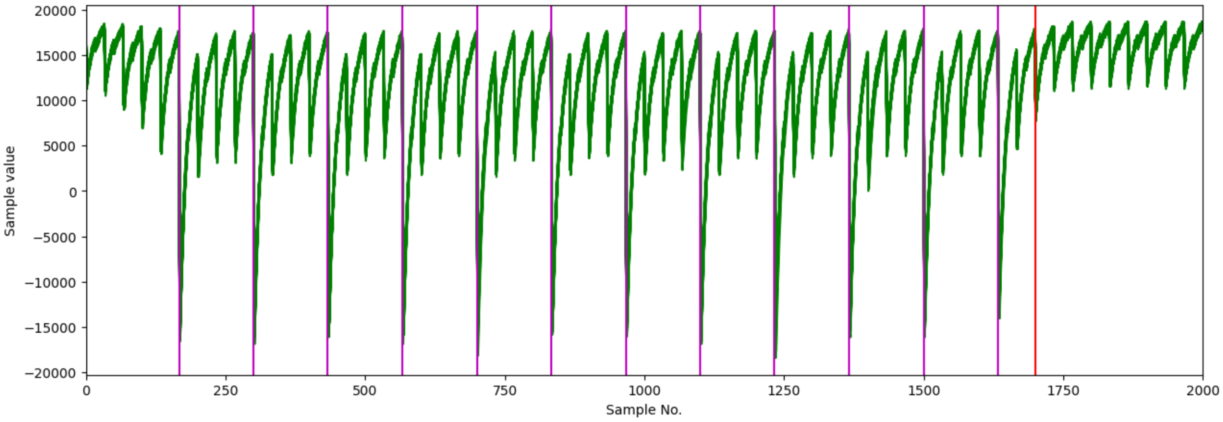

Figure 2: Waveform of power consumption.

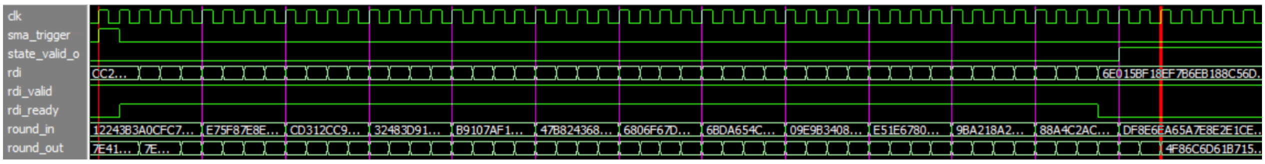

Figure 3: Simulation waveform of one Xoodoo permutation.

   (b) Leakage analysis: **Figure 1 shows that a significant leakage occurs at the moment marked by the red vertical line. This moment is also marked in Figure 2 and Figure 3 with red lines. To help understand the power consumption waveform, we mark the transition moments of the 768-bit state register (signal "*round_in*" in Figure 3) with magenta vertical lines in Figure 2 and Figure 3, which also correspond to the ending moments of twelve rounds. As shown in Figure 3, the leakage occurs two cycles after the permutation is done (state_valid_o == 1). The reason of this leakage is that the contents of the state register ("*round_in*" in Figure 3) are unintentionally sent to the round function module for another round**

**of calculation after all 12 rounds are done, i.e., 13 rounds are actually calculated in a permutation. Besides, this 13th round is calculated without the protection of fresh random number, as the signal "*rdi_ready*" is inactive after the permutation is done.**

(c) Discussion: **This leakage can be avoided by adding a control signal of the round function calculation or by not storing the permutation result in the state register. It shows that just replacing the core calculation circuit with secure gadgets does not automatically guarantee the security of the whole circuit.**

## References

[GGR11] Josh Jaffe Gilbert Goodwill, Benjamin Jun and Pankaj Rohatgi. A testing methodology for side-channel resistance validation. In *NIST Non-Invasive Attack Testing Workshop*, Nara, Japan, 2011.