Leakage Assessment on TinyJAMBU_first_order

10/7/2022

Abubakr Abdulgadir Jens-Peter Kaps Kris Gaj

1. Target

a) Algorithm: **TinyJAMBU**
b) Implementer: **Ruhr-University Bochum, Germany**
c) Variant: **TinyJAMBU_first_order**
d) URL: **https://github.com/Chair-for-Security-Engineering/LWC-Masking**
e) Commit hash: **a2c2e26d33c851697dc03ba97b92b7746bae74aa**
f) Protection method: **Hardware Private Circuits 2 (HPC2)**
g) Protection order: **1**

2. Equipment and Software Used

(a) General type of the evaluation platform: **Control board is FOBOS3 board for control. This board uses a PYNQ-Z1 board with a Zynq SoC (XC7Z020-1CLG400C) and a custom board that hosts an ADC for power measurement. The target board is NewAE CW305 Aritx7 (xc7a100tftg256)**
(b) Oscilloscope and its major characteristics: **We utilized OpenADC with 40 MHz bandwidth and a maximum sampling rate of 100 M Sample/sec.**
(c) Current and electromagnetic probes: **N/A**
(d) Usage of bandwidth limiters, filters, amplifiers, etc. and their specification: **Power was measured at the output of the CW305's onboard amplifier which amplifies the voltage drop across the board's 0.1 Ω shunt resistor.**
(e) Are sampling clock and design-under-evaluation clock synchronized? **Yes**
(f) Names and versions of programs used for evaluating side-channel resistance: **FOBOS3 analysis software.**
g) Clock frequency of target: **10 MHz**
h) Sampling frequency and resolution: **50 M Sample/sec sampling frequency and 10 bit resolution.**

3. Leakage Assessment Method

a) Type of the method: **Fixed-vs-random Test Vector Leakage Assessment [GJJR11 ,SM15].**
b) Number of traces: **10 million traces.**
c) Source of randomness: **Trivium-based DRBG.**
Before each test vector is processed, the DRBG is run to store the required number of random words in a FIFO. The design-under-evaluation then consume randomness from the FIFO and the DRBG is disabled while the design-under-evaluation is running. The goal of this is to eliminate the effect of the DRBG power consumption on the measurements.

d) Trigger location relative to the execution start time of the algorithm: **Triggered ADC at the start of the algorithm.**
e) Time required to collect data for a given attack/leakage assessment: **About 28 hours.**
f) Total time of the attack/assessment: **About 28 hours.**

g) Total size of all traces (if stored): **91 GB.**
h) Availability of raw measurement results: **N/A.**
i) Test vector generation:

The test vector generation procedure is as follows:

1- Generate a short unshared test vector using cryptotvgen.

2- Convert it into a shared format using gen_shared.py.

3- Generate the FOBOS-ready fixed-vs-random test vectors using lwc_2_fobos_tv.py. The SDI is kept similar in all test vectors, and the PDI section is fixed in fixed test vectors and random in the random test vectors. Fresh sharing on the PDI section is generated in all test vectors.

4. Results:

a) Documentation of results:

Test on the original implementation

We performed a first TVLA test on the implementation in Github and there was a significant leakage visible as shown in Figure 1.

By looking into the code, we observed that the input and output shares are combined in the tinyjambu_control.vhd file. We removed the lines where the shares are combined and performed more tests as shown below. Note that the resulting implementation ignores tag comparison.

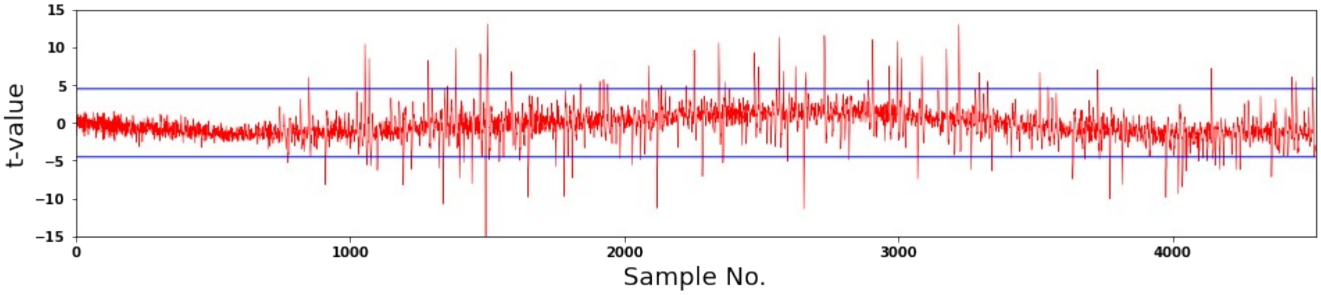

Figure 1: First test on the TinyJAMBU_first_order t-values (100,000 traces)

Test on the implementation with no tag comparison

We performed another TVLA test on the implementation with tag comparison disabled as mentioned above. The result of this test is shown in Figure 2 and 3. We observe that there is one spike above the threshold so we performed another test to confirm the leakage. The results of the third test is shown in Figure 4 and 5. Again one sample is sightly above the threshold but not at the same location as the previous test and for this reason it is likely that this is a false positive. This is consistent with the rule the device fails the test only when a t-value exceeds the threshold in the same time sample and the same direction in two independent experiments [GJJR11] .

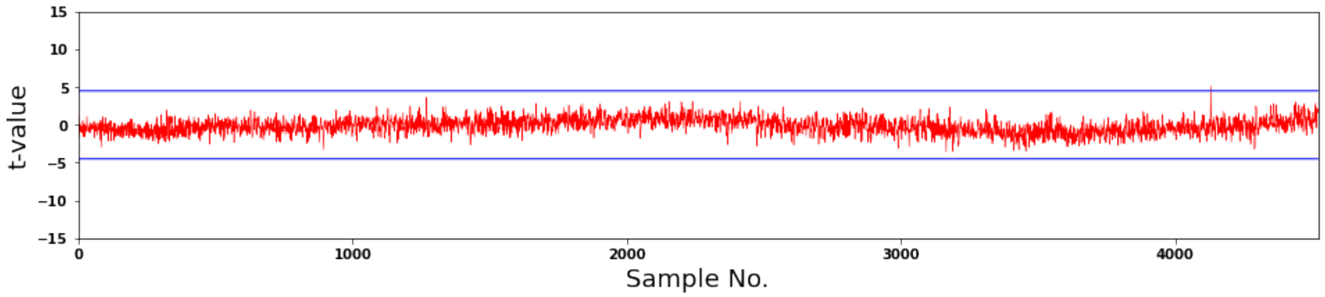

Figure 2: Second test on TinyJAMBU_first_order with no tag comparison: t-values (10 million traces)

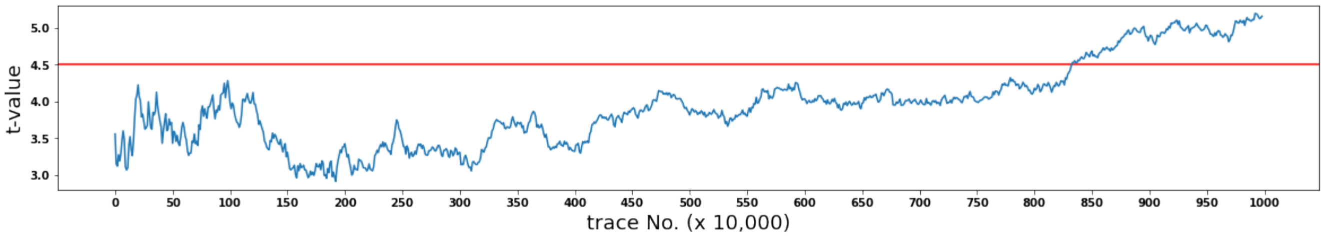

Figure 3: Second test on TinyJAMBU_first_order with no tag comparison : maximum t-value vs. number of traces (x 10,000)

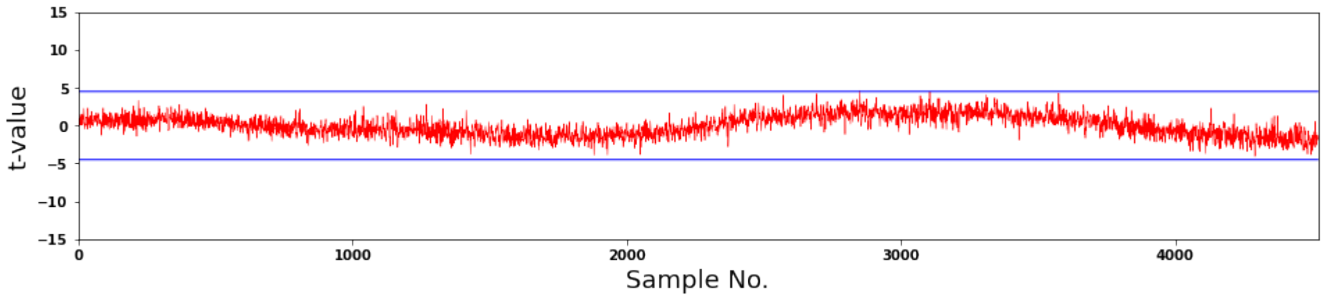

Figure 4: Third test on TinyJAMBU_first_order with no tag comparison: t-values (10 million traces)

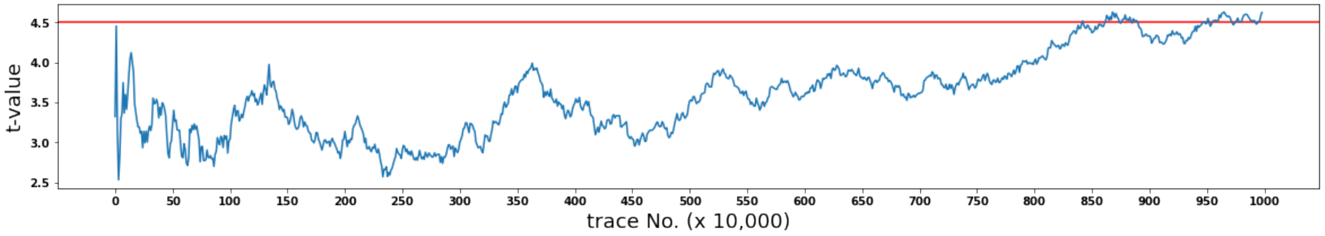

Figure 5: Third test on TinyJAMBU_first_order with no tag comparison : maximum t-value vs. number of traces (x 10,000)

References

[GJJR11] G. Goodwill, B. Jun, J. Jaffe, and P. Rohatgi, “A testing methodology for side-channel resistance validation,” Nara, Japan, 2011.

[SM15] T. Schneider and A. Moradi, “Leakage Assessment Methodology - a clear roadmap for side-channel evaluations,” Cryptology ePrint Archive 2015/207, Jun. 2015. Accessed: Dec. 31, 2021. [Online]. Available: https://eprint.iacr.org/2015/207