Late Breaking Results: Configurable Ring Oscillators as a Side-Channel Countermeasure

Hassan Nassar, Simon Pankner, Lars Bauer, Jörg Henkel
Chair for Embedded Systems, Karlsruhe Institute of Technology, Germany
{hassan.nassar, lars.bauer, henkel}@kit.edu

***Abstract*—Side-channel attacks are a threat to computing devices. In this work, we propose a novel countermeasure against power analysis side-channel attacks. This countermeasure uses ring oscillators with runtime-configurable chain lengths to generate noise to hide the effects of the secret intermediate values on the device's power consumption. We develop our countermeasure to be compatible with a state-of-the-art of side-channel-attack detection mechanism. Therefore, our solution does not incur any extra area overhead as it uses a subset of the circuit needed for detection. We evaluate our countermeasure using the test vector leakage assessment test (TVLA test). When our countermeasure is active no side-channel leakage could be detected.**

***Index Terms*—Side-Channel Analysis, Hardware Security**

I. Introduction

The number of computing devices of any size and computational capability is continuing to rapidly grow. So do the non-secure environments that give a chance for physical attacks such as Side-Channel Attacks (SCAs) where the attacker uses unintended leakages like temperature, power consumption, or electromagnetic radiation to deduce the processed data [1].

One of the most discussed SCA is Power Analysis (PA). All that is needed for a PA is a small shunt resistor to collect power traces, e.g., using an oscilloscope [2]. Using several thousands of power traces the attacker is able to perform a statistical analysis to deduce the processed data. Several countermeasures against PA have been proposed such as using parallel implementations of the same circuit, increasing the noise within the circuit, or using dynamic voltage frequency scaling [1].

The countermeasures have a considerable power overhead, hence, they rely on a detection mechanism to enable them only when an attack is detected. The detection is usually very fast (order of microseconds) and can be applied periodically [3]. The periodic application does not harm the detection ability as a successful attack requires several thousand traces over a long period of time (order of minutes or hours) [1]. Thus, running the detection every couple of minutes is sufficient and does not have a significant power overhead.

One detection mechanism is to use Ring Oscillators (ROs) as a detection sensor [3]. The frequency of the ROs is monitored and changes in the frequency might indicate the attack. This is due to the fact that the shunt resistor decreases the voltage that is supplied to the device, which consequently affects the RO frequency.

This work was partially funded by the 'Federal Ministry for Economic Affairs and Climate Action' as part of the 'Central Innovation Program for SMEs'.

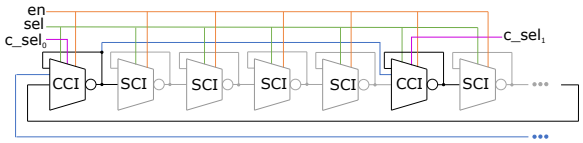


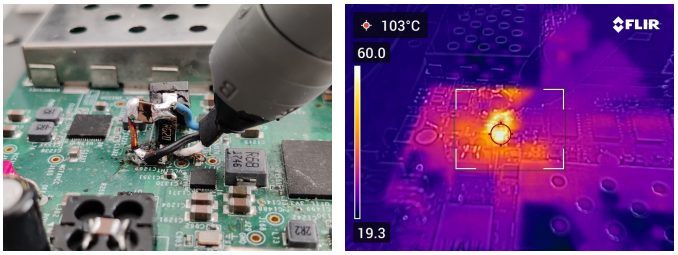
Fig. 1: Runtime-Configurable Ring Oscillator.

We develop our countermeasure using ROs as well, hence, it can be integrated with the detection mechanism without any additional area overhead. We implement the ROs with a runtime-configurable chain length. Using a random number generator, the chain length is constantly adjusted to a new random value, thus varying the frequency of the noise. This generates a random noise consisting of different frequencies and makes it much more difficult for an attacker to remove the noise from the measurements using a simple frequency filter.

II. Runtime-Configurable Ring Oscillators

Both, the detection mechanism and the countermeasure need several ROs to function. Exact numbers are detailed in Section III. The countermeasure is to be used only when a side-channel attack has been detected by the detection mechanism. Hence, we propose to use the ROs for both detection and countermeasure. To achieve this, the ring oscillators are implemented as a chain of runtime-configurable ring oscillators (RCROs) as shown in Fig. 1. The building blocks of the RCRO are the Configurable Inverters (CIs). CIs have two types: Simple CIs (SCIs) and Complex CIs (CCIs). A chain of RCROs consists of 65 CIs (13 CCIs and 52 SCIs). When used for detection, each CI is connected in a chain of length one, i.e., its output is directly fed back as its input. The deviations in frequency over a **large number** of ROs can then be tracked and fed to an ML model to detect if a PA is performed [3].

Once an attack is detected, we change the operation of the CIs to generate random noise. For SCIs, if the sel bit is set, the output of the previous inverter in the chain is used as the input, instead of its own output being applied to the input as immediate feedback. For CCIs, not only the output signal of the previous SCI or its own feedback can be used as an input signal, but also the output signal of the previous CCI in the chain. For this additional case, a second c_sel control bit is required for each CCI. If the c_sel control bit is not set, the CCI behaves like an SCI. However, if both the sel



(a) PA measurement shunt resistor (b) countermeasure thermal effect

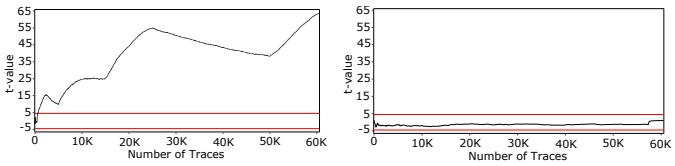
Fig. 2: Experimental setup

control bit and the c_sel control bit are set, the output signal of the preceding CCI in the chain is selected as the input signal and the SCIs that are between the two CCIs are skipped to reduce the length of the chain by four inverters. Since the c_sel control bit can be set individually for each CCI, it is possible to change the chain length at any time by changing the value of c_sel. Depending on the configuration, the chain can have a variable length between 13 (all SCIs are skipped) and 65 CIs (all SCIs are used) with a step size of 4 CIs.

The switch between detection and countermeasure is controlled by a central control module. When in countermeasure mode, the control module obtains a random number R from a Random Number Generator (RNG) and then enables all CIs R clock cycles, and then disables them for R clock cycles. A new number R is then generated and the process is repeated. This generated signal is forwarded by the control module to the ring oscillators as an enable signal when the countermeasure is activated so that these are always active or inactive for a random number of cycles. This generates random noise in the time domain. To generate random noise in the power domain, the c_sel signals for each CCI are also controlled by the RNG, Thus, a random number of SCIs is skipped at each run to vary both the power consumption and the frequency of the noise. Consequently, the noise does not contain any regular patterns that could be filtered out. This is in contrast to using all RCROs all the time which would have a regular pattern.

III. Evaluation

To evaluate the design, the countermeasure was prototyped on a Xilinx ZCU104 board. Each CI is implemented using one Look Up Table (LUT). Furthermore, a measurement setup was developed that consists of the evaluation board with the countermeasure and a reference AES implementation, an oscilloscope, and a PC responsible for controlling the measurement process. We use a pseudo RNG in our prototype as proof of concept. However, for a real system, a true RNG should be used. Additionally, a 620 mΩ shunt resistor is inserted into the FPGA power supply (VCC_INT) to collect the power traces as shown in Fig. 2a. Figure 2b shows the challenging thermal effect of our countermeasure. As our countermeasure will run continuously once an attack is detected, it can overheat our system. Hence, we had to fine-tune the number of RCROs used for the countermeasure to the minimum needed in order not to damage the whole system.



(a) TVLA of unprotected AES (b) TVLA of protected AES

Fig. 3: Test vector leakage assessment (TVLA) results

We further evaluate the effectiveness of the countermeasure using the test vector leakage assessment method (TVLA) [4]. It looks for leakage that depends on the secret key or the input data in cryptographic operations. It uses two sets of power traces, the first set is generated by always choosing the same, fixed input data. In the second set, the plain texts are chosen randomly. Then a Welch's t-test is applied to the two data sets to determine whether they differ significantly from each other. If the t-value stays in the range of $-4.5 < t < 4.5$ the system is considered secure, otherwise leakage exists [4].

We run the TVLA method on our system, once with the countermeasure activated and another with the countermeasure deactivated. For each, we collect 120K traces, 60K are with fixed input and 60K with random input. Based on the traces, we evaluate the t-value. Figure 3 shows the TVLA results. For the unprotected case (shown in Fig. 3a) the t-value rapidly grows out of the secure range. However, for the protected case (shown in Fig. 3b), the t-value stays in the secure range even when using all 60K traces. The countermeasure needs only **32 RCRO chains (using 2080 LUTs)** to be effective on the ZCU104 board while the detection on the same board needs **256 RCRO chains (using 16640 LUTs)**. Therefore, the countermeasure does not need any extra area overhead as it uses only a subset of the RCROs needed for the detection.

IV. Conclusion

We develop a countermeasure in this work to prevent side-channel attacks. We first developed a concept of how noise with random and constantly changing frequencies can be generated using ring oscillator chains of varying lengths. We prototype our solution on an FPGA board and the power traces of the device are collected using an oscilloscope. Based on the power traces, we use the TVLA method to assess the leakage. Using our countermeasure, the t-value never goes outside the range of $-4.5 < t < 4.5$, i.e., our countermeasure successfully stops the leakage of critical data. Our solution does not incur any area overhead as it combines with the state-of-the-art detection mechanism of side-channel attacks.

References

[1] E. De Mulder, T. Eisenbarth, and P. Schaumont, "Identifying and eliminating side-channel leaks in programmable systems," *IEEE Design & Test*, vol. 35, no. 1, pp. 74–89, 2018.

[2] M. Zhao and G. E. Suh, "FPGA-based remote power side-channel attacks," in *Symp. on Security and Privacy*, 2018, pp. 229–244.

[3] N. Gattu, M. N. I. Khan, A. De, and S. Ghosh, "Power side channel attack analysis and detection," in *ICCAD*, 2020, pp. 1–7.

[4] F. Bache, C. Plump, and T. Güneysu, "Confident leakage assessment - A side-channel evaluation framework based on confidence intervals," in *DATE*, 2018, pp. 1117–1122.