Saidoyoki: Evaluating side-channel leakage in pre- and post-silicon setting

Pantea Kiaei
WPI
pkiaei@wpi.edu

Zhenyuan Liu
WPI
zliu12@wpi.edu

Ramazan Kaan Eren
WPI
reren@wpi.edu

Yuan Yao
Virginia Tech
yuan9@vt.edu

Patrick Schaumont
WPI
pschaumont@wpi.edu

***Abstract*—Predicting the level and exploitability of side-channel leakage from complex SoC design is a challenging task. We present Saidoyoki, a test platform that enables the assessment of side-channel leakage under two different settings. The first is pre-silicon side-channel leakage estimation in SoC, and it requires the use of fast side-channel leakage estimation from a high-level design description. The second is post-silicon side-channel leakage measurement and analysis in SoC, and it requires a hardware prototype that reflects the design description. By designing an in-house SoC and next building a side-channel leakage analysis environment around it, we are able to evaluate design-time (pre-silicon) side-channel leakage estimates as well as prototype (post-silicon) side-channel leakage measurements. The Saidoyoki platform hosts two different SoC, one based on a 32-bit RISC-V processor and a second based on a SPARC V8 processor. In this contribution, we highlight our design decisions and design flow for side-channel leakage simulation and measurement, and we present preliminary results and analysis using the Saidoyoki platform. We highlight that, while the post-silicon setting provides more side-channel leakage detail than the pre-silicon setting, the latter provides significantly enhanced test resolution and root cause analysis support. We conclude that pre-silicon side-channel leakage assessment can be an important tool for the security analysis of modern Security SoC.**

***Index Terms*—Side-channel Leakage Verification, Secure IC Design, Hardware Security**

I. Introduction

Power-based side-channel leakage is a known vulnerability in security SoC, yet it is hard to predict the amount of side-channel leakage at design-time. The fundamental reason is that the source of the vulnerability, namely data-dependency in the power dissipation of a design, is found at every abstraction level in the system stack [1]. For example, a design may appear perfectly side-channel resistant at RTL level, yet due to imperfections of the implementation at gate-level or below, the ideal side-channel leakage properties of RTL break down and cause side-channel leakage in the form of glitches [2] or cross-talk [3]. This strongly suggests that extensive verification of side-channel leakage properties, at every abstraction level of the design, is crucial. In fact, contemporary *provably secure* countermeasures against side-channel leakage always assume a leakage model, a set of assumptions that must be supported by the implementation to deliver the security properties claimed. These leakage models (and their correctness for a given implementation) are an ongoing area of research [4].

Design-time verification of power-based side-channel leakage can be supported through power modeling and simulation. But at lower abstraction levels, power modeling is complex, and it comes with steep trade-offs between simulation time and resolution. Therefore, in the absence of comprehensive leakage modeling and/or efficient power simulation, the current design practice in side-channel resistant design in many cases still relies on prototyping. A prototype provides a real-life design test-case that can be measured and evaluated from a power side-channel leakage perspective. Field Programmable Gate Arrays (FPGA) are often selected as a prototyping target. However, the low level structure of FPGA does not reflect the gate-level netlist that is mapped on it, and therefore the FPGA may not be the best choice for the study of ASIC side-channel leakage behavior at low level. To compare the power side-channel leakage of a gate level netlist model to that of a prototype, ASIC technology with standard cells provides a better match.

In our recent research, we have designed several ASICs as a byproduct (and often as a proof of concept) of our experimental work. FAMEv2 (Fault Aware Microprocessor Extensions) is an SoC with fault-sensing capability based around a LEON-3 core. PICO is a similar SoC based around a 32-bit RISC-V core. Both SoCs contain several coprocessors as well as on-chip RAM, and they are built in 180nm standard cells. Most importantly, they are in-house designs, so we have access to all design information down to layout level.

To study the problem of pre-silicon vs post-silicon side-channel leakage modeling, we integrated both of these SoCs in a test platform. The Saidoyoki board provides a programming interface to download applications to either chip, and supports high-bandwidth power measurement of each individual chip. The purpose of Saidoyoki is to validate a design flow for pre-silicon side-channel leakage estimation, by providing estimations next to actual side-channel leakage measurements. Our long term objectives are to understand and address two crucial shortcomings of side-channel leakage estimation from high-level models: (a) *false positives*, where the side-channel leakage estimation on the pre-silicon design model indicates side-channel leakage that cannot be confirmed by measurements; (b) *false negatives*, where the side-channel leakage estimation on the pre-silicon design does not show side-channel leakage while the measurements confirm the opposite. Both problems are hard and, to our knowledge, still unsolved.

The remainder of this paper is organized as follows. In Section II, we introduce the Saidoyoki platform at system-level,

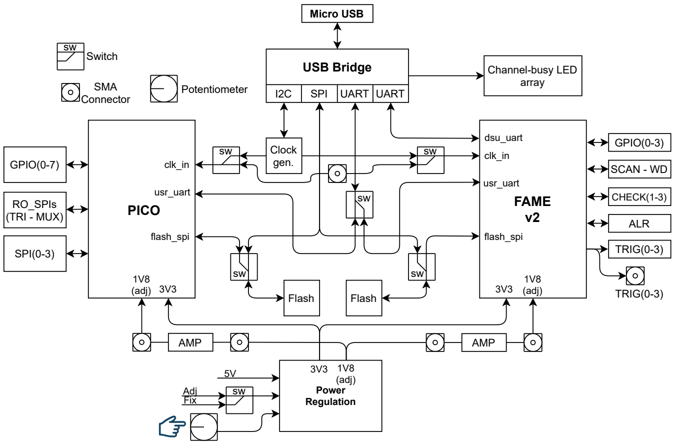

Fig. 1: Block diagram of Saidoyoki Board.

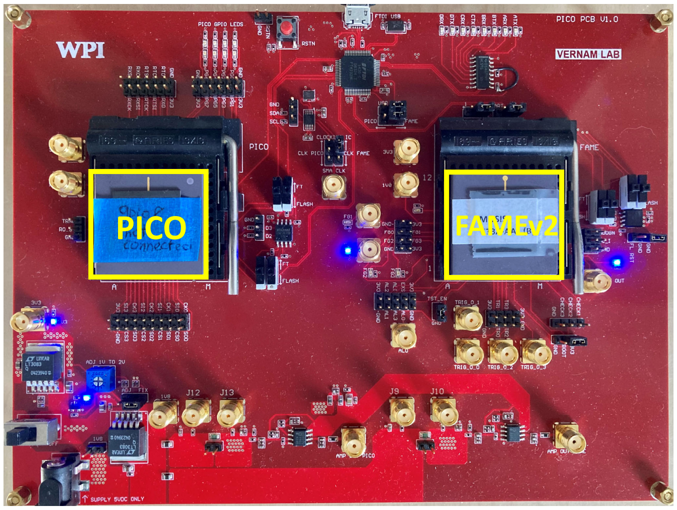

Fig. 2: Photo of the Saidoyoki Board.

including the design decisions on the PCB to instrument it for side-channel leakage measurement, as well as a brief overview of the ASIC designs. Section III describes pre-silicon side-channel leakage estimation techniques. We describe the design flow used for the SoC power estimation, and handle several practical challenges related to design complexity. Section IV describes post-silicon side-channel leakage measurement using the Saidoyoki board. Section V describes several experiments using Saidoyoki, and the analysis of the results obtained so far. Section VI concludes the paper.

II. Saidoyoki Platform

This section describes the features of the Saidoyoki platform, including the platform architecture, and the system architecture of the chips. We also compare with related work.

A. Saidoyoki PCB

A side-channel measurement setup includes a test target, a means to digitize power side-channel leakage, and a side-channel campaign controller. The controller exchanges stimuli with the test target, and while the target executes the stimuli, the target's power signature is captured. Afterwards, the campaign controller collects the power signatures and performs side-channel analysis. A PCB to support a test target thus has to provide multiple functions, including (a) providing easy access to the power consumption of the target, (b) real-time data input/output to exchange test stimuli with the campaign controller, (c) debugging of programmable/reconfigurable targets, and optionally, (d) adjusting target voltage/clock to study the impact of environmental factors. The Saidoyoki PCB was developed to support the PICO and FAME chips as targets for side-channel measurement campaigns, and it supports all functions enumerated above.

Figure 1 shows the block diagram of the board. The board is connected to the side-channel campaign controller with a single USB connection to multiplex multiple data and control channels used in a campaign (FT4232HL USB Bridge IC). These control channels include a UART debug connection for FAMEv2, a shared user UART, a shared SPI to program flash memories, and an I2C control channel for clock configuration.

The side-channel leakage of each ASIC is captured by measuring voltage drop over a shunt resistor. The signal is also amplified through a differential broadband opamp, one for each ASIC. Saidoyoki uses a single 5V power supply that is regulated into a fixed 3V3 supply and an adjustable 1V8 supply. The adjustable supply feeds the ASIC core and can be varied between 1V and 2V. This rail is also split into two using ferrite beads and then connected to each chip independently. A precise shunt resistor is inserted into each branch of the 1.8V rail for power measurement.

The flash chips connected to PICO and FAMEv2 are externally programmable through USB. The board has several 3-point slider switches to physically switch the flash chips from external configuration to ASIC configuration. In the ASIC configuration, the ASIC will boot from the firmware configured in the flash chips.

Saidoyoki also includes a clock generation IC (SI5351A-B-GTR), which is able to generate configurable clock frequencies for the ASICs between 2.5KHz and 200MHz. The clock can also be provided from an external source through an SMA connector.

Finally, the Saidoyoki board supports low-level debugging tasks by bringing every chip pin out on a jumper header or an SMA connector. Additionally, the GPIO ports in each ASIC are visible through LEDs.

B. FAMEv2 ASIC

The FAMEv2 ASIC is a 180nm SoC with a LEON3 core and 128 kByte of internal memory, and several coprocessors. The program can execute either from on-chip SRAM or else from off-chip flash through an SPI flash ROM. A debug unit, controlled through an on-chip Debug UART, provides program loading, monitoring, breakpoints. The coprocessors are isolated from the processor through a bus bridge. All coprocessors exclusively operate as bus slaves, and communicate with the software through memory-mapped registers.

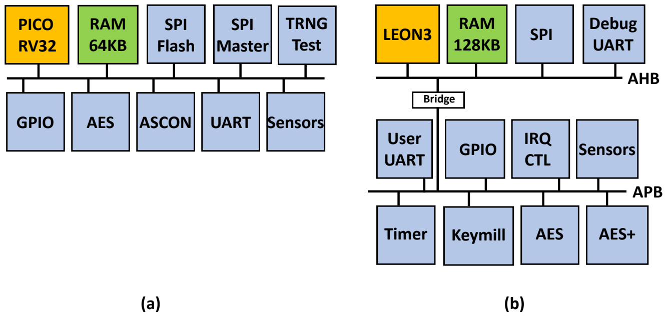

Fig. 3: (a) PICO Block diagram and (b) FAMEv2 Block diagram.

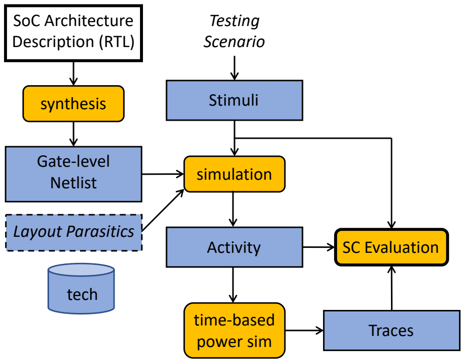

Fig. 4: Flow for SCL assessment of hardware

FAME contains cryptographic accelerators for symmetric-key encryption (AES and AES+, a hardened version of AES) and pseudo-random stream generation (KeyMill). The sensors in FAME detect timing faults injected through clock glitching and voltage glitching. A detailed description of the fault detection mechanisms, and their integration with software, was presented earlier [5].

C. Pico ASIC

The PICO ASIC is a 180nm SoC with a RISCV (RV32) core and 64 kByte of internal memory, and several coprocessors. The program exclusively runs from off-chip flash through a Quad-SPI flash ROM. The system is integrated on a single bus. All coprocessors run as bus slaves and communicate with the RISC-V software through memory-mapped registers. PICO contains cryptographic accelerators for symmetric-key encryption (AES), authenticated encryption (ASCON), and hardware testing of true random bitstreams (TRNG test). The sensors in PICO detect fault injection as well as side-channel leakage. The FPGA prototype design of the sensors was presented earlier [6]. Furthermore, the high-level description of the sensors on PICO ASIC was presented earlier [7].

D. Related Work

Several other solutions have been proposed for high-bandwidth power monitoring of hardware. The SASEBO series of side-channel analysis boards [8], originally developed by AIST, is the oldest and arguably best known implementation. One version of SASEBO, SASEBO-R, is the only open source board to support an ASIC. The SAKURA series of boards [9], also by AIST, are based on Kintex-7/Spartan-6 FGPA or chip card microcontroller. The FOBOS from GMU [10] is an FPGA-based board oriented towards benchmarking. The HAHA board from U Florida [11] is a hybrid FPGA/microcontroller board oriented towards education. The Chip Whisperer is a low-cost measurement environment with FPGA-based and microcontroller based target boards [12]. The majority of these boards are oriented at studying side-channel leakage in a (configurable or programmable) test target. With Saidoyoki, since we have full knowledge of the test target's internal design, we can evaluate side channel leakage in either pre-silicon or else post-silicon side-channel leakage evaluation scenario's.

III. Pre-silicon Side-channel Leakage Estimation

In a pre-silicon setting, power-based side-channel leakage is estimated through a time-based simulation of the power consumption of the target.

A. Design flow for Hardware Targets

There are many solutions towards capturing side-channel leakage by simulation [1]. Figure 4 describes the steps for a hardware target such as for example a coprocessor in one of the SoCs. The Saidoyoki flow starts from a gate-level netlist, obtained from the FAME/PICO chip design or else through RTL synthesis of the design flows. Through a suitable testing scenario, a set of testing stimuli are defined. For classic DPA/CPA analysis of cryptographic hardware, for example, one selects a fixed key and a set of random plaintext/ciphertext. Specific or non-specific TVLA tests, on the other hand, require a combination of random and fixed key/plaintext inputs [13]. The activity files created from hardware simulation are used by a time-based power simulation tool, which provides a trace for every input test vector. Finally, the stimuli, activity files and traces are used as input for the side-channel evaluation. Our prototype implementation supports Cadence Genus or Synopsys Design Compiler, Cadence XCelium or Mentor ModelSim, and Cadence Joules respectively as synthesis, simulation, and power estimation tool. The side-channel evaluation is a customized Chipwhisperer script.

There is a trade-off between the level of simulation detail, and the ability of a design model to capture different causes and sources of side channel leakage. The gate-level abstraction provides reasonable accuracy. Because of the post-silicon artifacts available in Saidoyoki, we prefer lower simulation abstraction levels to investigate and verify lower-level effects.

B. Design flow for Software Targets

To ensure sufficient accuracy with software and SoC firmware targets, we extend the hardware based flow and simulate the microprocessor as a gate-level design as well. The main challenge is to convert the firmware into a form that can

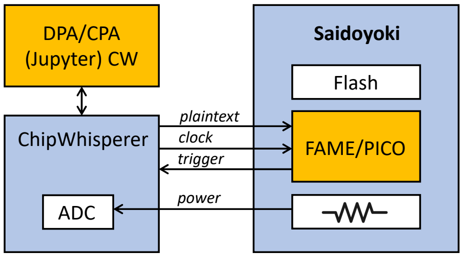

Fig. 5: Integration of Chipwhisperer and Saidoyoki

be integrated into the SoC hardware simulation. We compile the firmware into a binary that is used to initialize the on-chip memory at the start of the simulation. By compiling input test vectors (key, input) as hard-coded constants in the firmware binary, we also eliminate complex input/output schemes during simulation.

To enable testing only specific parts of the firmware, we make use of the GPIOs on the SoC as triggers. We set a GPIO pin high right before the point of interest and reset it to low right after. In the test bench module, during simulation, we monitor the trigger signal and log its set and reset time stamps. Later, in power trace generation phase, we generate the power trace only for the logged time period.

IV. Post-silicon Side-channel Leakage Measurement

Measuring side-channel leakage in the post-silicon setting requires a campaign controller to measure power from the PICO or FAME chips on Saidoyoki while they execute a test application. We use a Chipwhisperer kit [12] integrated as in Figure 5. Chipwhisperer uses synchronous sampling and generates a clock signal for the target. This allows the side-channel leakage to be captured with low overhead of one to four times the target clock. A campaign executes a large collection of encryption operations on the target, with different input stimuli. For each operation, Chipwhisperer sends an input plaintext. The target responds with a trigger when the cryptographic operation starts. Chipwhisperer then collects the power signal and finally performs side-channel leakage analysis or assessment.

V. Results

In this section, we describe three experiments performed using the Saidoyoki board. The first is a post-silicon side-channel leakage analysis on FAME; the second and third are a pre-silicon side-channel leakage analysis on PICO. In each case, we used AES encryption as the target algorithm, with the SubBytes output as the leakage model.

A. Post-silicon evaluation of FAME SoC firmware

Figure 6 (top) shows a power signal captured from the first round of AES encryption running on the LEON3 core in the FAME chip. Typically, post-silicon traces are very noisy and it's not easy to visually recognize the different portions of the algorithm. However, since the code on the target is fully known, it's easy to determine when each function executes. In this campaign, the FAME target runs at 4MHz and the side-channel is sampled at 16 MHz. Each trace contains 24,400 points. Figure 6 (bottom) shows a correlation plot obtained from running CPA on key byte 7 on the traces of 25,000 encryptions. The correlation plot shows two spikes: one of them when the SubBytes output is computed and stored in memory, and a second when ShiftRows reads that result and moves it to another memory location. While this type of side-channel analysis is a standard operation, it is not without its pitfalls. The black-box nature of the power signal, as well as the high level of noise, requires careful tuning of the measurement parameters.

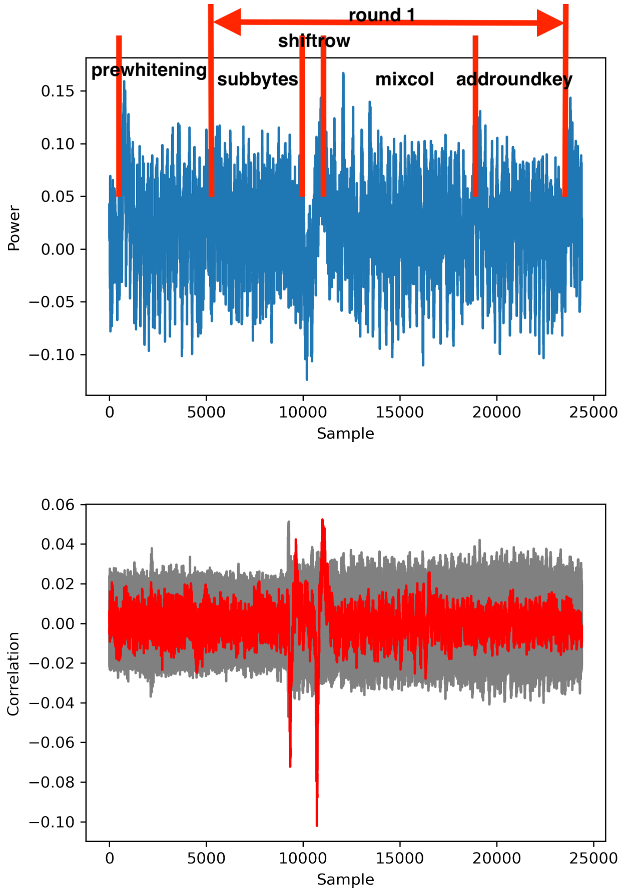

Fig. 6: CPA HW on FAME executing AES firmware: (top) power traces identifying portions of the first round (bottom) outcome of Correlation Power Analysis

B. Pre-silicon evaluation of PICO SoC coprocessor

A significant advantage of Saidoyoki is its ability to support pre-silicon side-channel leakage assessment using gate-level power simulation. There is a trade-off between the speed of a campaign (determined by the speed of gate-level simulation) and the noise level of the side-channel leakage. In a pre-silicon setting, we capture only a fraction of the traces that are collected in a post-silicon setting. On the other hand, the

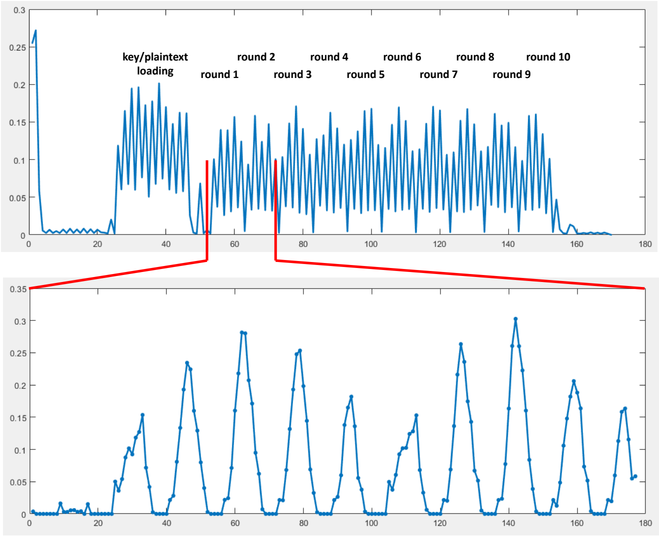

Fig. 7: Gate-level Power simulation of an AES hardware coprocessor: (top) entire encryption at two samples per cycle (bottom) zoom on first and second round at 16 samples per cycle

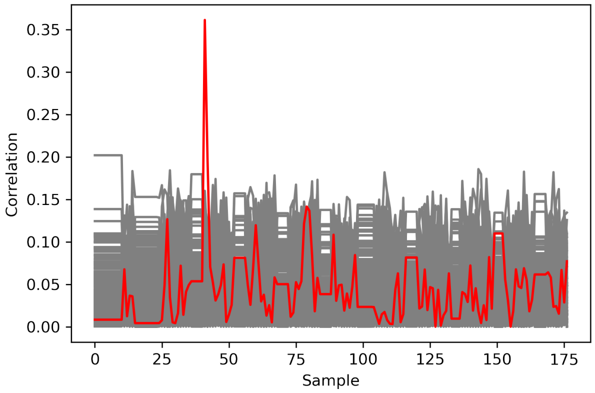

Fig. 8: Correlation on the PICO HD AES pre-silicon trace @16 samples per cycle

absence of noise implies that side-channel leakage assessment or analysis will converge much quicker.

Figure 7 shows the result of a gate-level simulation of the AES coprocessor in the PICO chip. The top plot is a simulation at two power samples per clock cycle, while the bottom plot is a simulation at 16 power samples per clock cycle. The resolution of the power trace can thus be easily adjusted without penalty on the noise level. Increasing the resolution of a simulated power trace has a different effect as well: at higher resolutions, fewer gates contribute to a single power sample. This is because a gate-level simulation properly models the switching time of each gate, and at sub-cycle resolution, different gates will switch at different time instants.

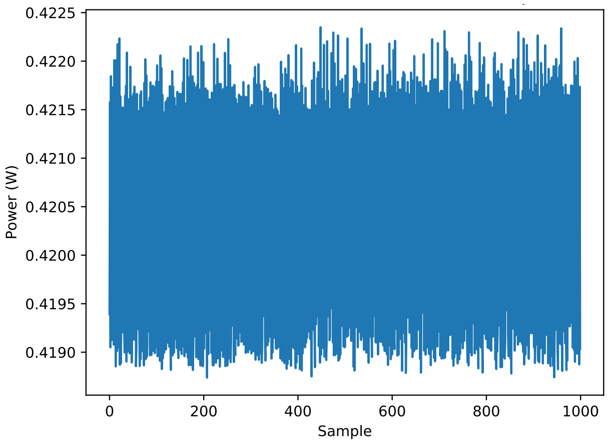

Fig. 9: Sample simulated power trace of software AES running on PICO chip. This trace includes only the first add round key and SBox in the first round of encryption.

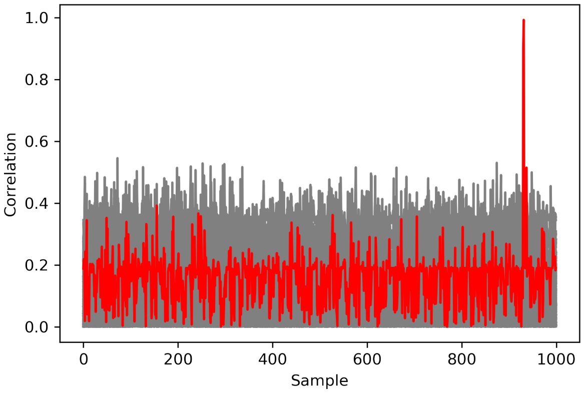

Fig. 10: Correlation outcome on the PICO HW AES pre-silicon trace

Figure 8 shows a CPA result on 400 simulated traces, showing a sharp correlation peak in cycle 2 of round 1. While the length of a simulation trace is in principle unbounded (when compared to the limited length of a sample buffer in a post-silicon setup), in practice we aim to make the traces as short as possible to minimize the simulation time overhead. Choosing the proper time window of simulation can be a challenge when the source of side-channel leakage is unknown. On the other hand, in a pre-silicon simulation, we can undersample the power consumption, as the simulator will accumulate power over multiple cycles without adding (physical) noise or precision. The third experiment, discussed next, builds on this property.

C. Pre-silicon evaluation of PICO SoC firmware

We experiment with the software implementation of AES encryption in ECB mode running on the PicoRV32 core in PICO chip. As we aim to target the SubBytes in the first round, we set a GPIO pin high before the first key addition and reset

TABLE I: Performance factors for each of the case studies

	V.A	**V.B**	**V.C**
Pre/Post Silicon	Post	Pre	Pre
ASIC	FAME	PICO	PICO
Target	AES SW	AES HW	AES SW
Correlation Peak	0.1	0.36	0.99
Number of Traces	25,000	400	60
Samples per Cycle	4	16	1/80
Samples per Trace	24,400	200	1000
Capture Time per Trace (s)	0.06	0.55	260
Assessment Time per Byte (s)	660	<1	<1

it to low after the SBox in first round. We use a Python script that generates random plaintexts and automatically generates the C code with hard-coded palintext values. Furthermore, we store the plaintext values in a text file for later use in side-channel analysis. We simulate the AES software running on the synthesized netlist of PICO with 180nm CMOS standard cell library. We run the simulation with a clock frequency of 80MHz and store the switching activity information in Value Change Dump (VCD) format. Using Joules, we compute the power trace for each VCD file. As an example, Figure 9 shows the plot of one of the generated power traces. As PicoRV32 is a non-pipelined architecture, even a small portion of the AES algorithm (first AddRoundKey and SubBytes of the first round) takes about 79k clock cycles to execute. Our power computation tool, Joules, can generate a maximum of 1000 samples per power trace. Each VCD file expands over around 990us, therefore, generating one power trace per VCD file results in a power trace with a sample rate of around 1MHz. While this is a strong under-sampled power trace for a device running at 80MHz (one sample per eighty clock cycles), CPA is able to find the private key with only 60 traces. Figure 10 shows the CPA result with Hamming Weight as the power model on 60 simulated traces, showing a sharp correlation peak (close to perfect correlation) in the first round SubBytes region.

With the simulation and VCD generation taking about 30 seconds and the power computation taking about 4 minutes, it took us less than five minutes to generate one power trace for our software implementation of AES. This experiment was performed as a single-thread process on an Intel Xeon Gold 6248 server.

D. Performance Evaluation

Table I summarizes our experiments. As this table shows, the correlation peaks in a software implementation are significantly higher than that of hardware implementations while also requiring orders of magnitude fewer traces even for an under-sampled trace. The capture time per trace in a post-silicon measurement is much faster than that of pre-silicon simulation, however, because of the reduced SNR in a physical implementation, a successful attack requires much more traces and thus takes more time to reveal each byte of the secret key.

VI. Conclusion

The pre-silicon tooling of side-channel leakage is rapidly catching up with the more traditional post-silicon prototyping strategy. The main advantage of pre-silicon techniques is that design flaws can be fixed at a low cost. There are two open challenges, both of them related to the accuracy by which pre-silicon modeling can reflect post-silicon measurement. False positive errors occur when pre-silicon tooling identifies side-channel leaks that are practically unexplotable in post-silicon context. False negative errors occur when pre-silicon tooling fails to identify exploitable leaks. Both of them are practical challenges, and will require a detailed comparison of pre-silicon models with post-silicon measurement. This research was supported in part by NSF grant 1931639.

References

[1] Ileana Buhan, Lejla Batina, Yuval Yarom, and Patrick Schaumont, “Sok: Design tools for side-channel-aware implementations”, Cryptology ePrint Archive, Report 2021/497, 2021, https://ia.cr/2021/497.

[2] Stefan Mangard and Kai Schramm, “Pinpointing the side-channel leakage of masked AES hardware implementations”, in *Cryptographic Hardware and Embedded Systems - CHES 2006, 8th International Workshop, Yokohama, Japan, October 10-13, 2006, Proceedings*, Louis Goubin and Mitsuru Matsui, Eds. 2006, vol. 4249 of *Lecture Notes in Computer Science*, pp. 76–90, Springer.

[3] Ilias Giechaskiel, Ken Eguro, and Kasper Bonne Rasmussen, “Leakier wires: Exploiting FPGA long wires for covert- and side-channel attacks”, *ACM Trans. Reconfigurable Technol. Syst.*, vol. 12, no. 3, pp. 11:1–11:29, 2019.

[4] Siemen Dhooghe and Svetla Nikova, “My gadget just cares for me - how NINA can prove security against combined attacks”, in *Topics in Cryptology - CT-RSA 2020 - The Cryptographers’ Track at the RSA Conference 2020, San Francisco, CA, USA, February 24-28, 2020, Proceedings*, Stanislaw Jarecki, Ed. 2020, vol. 12006 of *Lecture Notes in Computer Science*, pp. 35–55, Springer.

[5] Bilgiday Yuce, Chinmay Deshpande, Marjan Ghodrati, Abhishek Bendre, Leyla Nazhandali, and Patrick Schaumont, “A secure exception mode for fault-attack-resistant processing”, *IEEE Trans. Dependable Secur. Comput.*, vol. 16, no. 3, pp. 388–401, 2019.

[6] Yuan Yao, Pantea Kiaei, Richa Singh, Shahin Tajik, and Patrick Schaumont, “Programmable RO (PRO): A multipurpose countermeasure against side-channel and fault injection attack”, *CoRR*, vol. abs/2106.13784, 2021.

[7] Pantea Kiaei, Yuan Yao, and Patrick Schaumont, “Real-time detection and adaptive mitigation of power-based side-channel leakage in soc”, *arXiv preprint arXiv:2107.01725*, 2021.

[8] Research Institute for Secure Systems (RISCEC/AIST), “Evaluation environment for side-channel attacks”, https://www.risec.aist.go.jp/project/sasebo/, Accessed 7/26/2021.

[9] Hendra Guntur, Jun Ishii, and Akashi Satoh, “Side-channel attack user reference architecture board sakura-g”, in *2014 IEEE 3rd Global Conference on Consumer Electronics (GCCE)*, 2014, pp. 271–274.

[10] Abubakr Abdulgadir, William Diehl, and Jens-Peter Kaps, “An open-source platform for evaluation of hardware implementations of lightweight authenticated ciphers”, in *2019 International Conference on ReConFigurable Computing and FPGAs, ReConFig 2019, Cancun, Mexico, December 9-11, 2019*, David Andrews, René Cumplido, Claudia Feregrino, and Marco Platzner, Eds. 2019, pp. 1–5, IEEE.

[11] Shuo Yang, Shubhra Deb Paul, and Swarup Bhunia, “Hands-on learning of hardware and systems security”, *ASEE*, vol. 9, no. 2, pp. 1–25, 2021.

[12] Colin O’Flynn and Zhizhang (David) Chen, “Chipwhisperer: An open-source platform for hardware embedded security research”, in *Constructive Side-Channel Analysis and Secure Design - 5th International Workshop, COSADE 2014, Paris, France, April 13-15, 2014. Revised Selected Papers*, Emmanuel Prouff, Ed. 2014, vol. 8622 of *Lecture Notes in Computer Science*, pp. 243–260, Springer.

[13] François-Xavier Standaert, “How (not) to use welch’s t-test in side-channel security evaluations”, *IACR Cryptol. ePrint Arch.*, vol. 2017, pp. 138, 2017.