



Improving Side-channel Leakage Assessment Using Pre-silicon Leakage Models

Dillibabu Shanmugam^(✉) and Patrick Schaumont

Worcester Polytechnic Institute, Worcester, MA 01609, USA
{[dshanmugam](mailto:dshanmugam@wpi.edu), [pschaumont](mailto:pschaumont@wpi.edu)}@wpi.edu

Abstract. Side-channel leakage assessment is an essential tool in the security evaluation of new chip designs. Pre-silicon side-channel analysis tools have made significant progress in delivering assessment results early in the chip design flow. However, a gap remains with actual implementations where measurements are affected by noise and distortions. These measurement imperfections degrade the assessment of the physical prototype and may lead to false negatives. In this contribution, we present a transfer learning technique to improve the assessment of physical prototypes using pre-silicon side-channel leakage simulation of the same implementation. The noiseless simulation traces are used for initial profiling to train a convolutional neural network (CNN). The trained CNN is then used in the assessment of measured traces. We apply this idea to ASCON and XOODYAK, two different sponge-based cryptographic primitives proposed in the NIST Lightweight Crypto competition. The target platform is a software implementation on a RISC-V (RV32IMC) microcontroller realized using 180 nm CMOS technology. Side-channel leakage is first captured using gate-level power simulation and then measured from a chip prototype of the same design. We investigate different side-channel analysis strategies under simulated and measured scenarios and demonstrate that, in each case, machine-learning-based side-channel leakage assessment outperforms other profiled and non-profiled analysis. However, using the proposed transfer learning technique, we can improve the side-channel leakage assessment even further. With the proposed transfer learning technique, we need approximately 2.87 less measured traces compared to the previous best profiled attack. We conclude that the proposed transfer learning using pre-silicon leakage models can improve the side channel leakage assessment of post-silicon implementations.

Keywords: Transfer learning · ASCON · XOODYAK

1 Introduction

Side-channel leakage assessment, a critical step in the security evaluation of an IC, quantifies the amount of side-channel leakage from the implementation. There are multiple methodologies to characterize side-channel leakage of a cryptographic implementation [15]. However, all of them rely on data measurements.

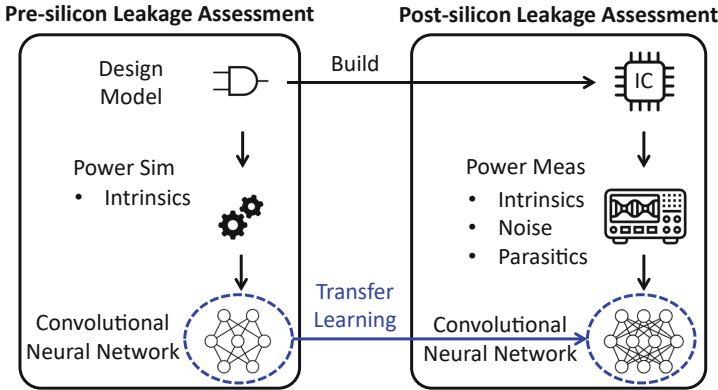


Fig. 1. Post-silicon leakage assessment can be improved with transfer learning from a pre-silicon leakage model.

This practical aspect of measurement requires engineering skills as well as insight into the cipher design, the target hardware technology, and the methodology of trace measurement. Practical side-channel measurement also faces challenges of reproducibility, because IC performance characteristics and power consumption are affected by voltage and temperature. For example, a side-channel campaign that gathers millions of traces takes days or even weeks to complete, requiring environmental controls on the test setup.

Pre-Silicon Side-Channel Leakage Assessment. During the design of a new IC, side-channel leakage assessment (SLA) can be directly implemented on the design descriptions of the hardware or the firmware [5]. A designer then uses simulation to create power traces for a design. Pre-silicon power simulation is noiseless and does not suffer from the imperfections suffered by physical measurement. In contrast to traditional SLA pre-silicon SLA is implemented in a white-box scenario with full knowledge of the design implementation details. Thus, pre-silicon SLA helps a designer to understand the weak parts of a design before committing it to silicon. In addition, pre-silicon SLA is able to support root-cause analysis of side-channel leakage to the single gate or the single instruction [12].

Improving Post-Silicon Side-Channel Leakage Assessment. In this contribution, we investigate how pre-silicon design knowledge can be applied to improve post-silicon SLA. We want to use the knowledge of simulated side-channel leakage properties on the evaluation of measured side-channel leakage. So far, this problem was studied only as a cross-device attack between different *physical* implementations [8, 22]. Instead, we are using a portability threat model [16] from simulation to implementation. Architectural abstracts as a predictive leakage model were explored in PARAM [1] and ROSITA [19].

Figure 1 shows our strategy which makes use of deep learning. A pre-silicon leakage assessment uses simulated power traces to map design-intrinsic leakage

properties into a CNN. The simulated traces are noiseless and without distortion. A post-silicon leakage assessment of the same design uses measured power traces to map design-intrinsic leakage properties to a threat model. Measured power traces may be corrupted by noise and measurement parasitics. Because of these distortions, the post-silicon CNN has a harder time to learn the intrinsic leakage properties of the design. To improve the post-silicon training, we apply a *transfer learning* technique, which carries over some of the properties of the pre-silicon CNN to the post-silicon CNN. Earlier work in transfer learning to support the portability threat model was presented by Thapar *et al.* for the case of cross-FPGA analysis [21], and by Paguada *et al.* as a generic toolbox for deep-learning based side-channel analysis [14]. We believe our work is the first to demonstrate the use of transfer learning techniques for SLA between the pre-silicon (simulated) and post-silicon (measured) environment.

Use Scenario. Since our proposed transfer learning for SLA assumes that both the design files and the physical implementation of a design are available, we motivate the practical meaning of this assumption. First, we observe that for new designs, the pre-silicon design phase always transitions into a post-silicon phase after tape-out. Hence, it is helpful to transfer the pre-silicon SLA results to the chip prototype evaluation, for the same reason pre-silicon test vectors are beneficial to test the prototype’s functionality.

Second, intellectual property modules for cryptography can benefit from a mechanism to transfer side-channel leakage properties from design to implementation. In current practice, only high-level (algorithmic) leakage models, such as the Hamming Distance on a specific intermediate variable, capture the side-channel leakage properties of an intellectual property (IP) module. In contrast, our pre-silicon CNN is developed from gate-level power simulation and reflects the specific leakage characteristics in much greater detail. This model is, therefore, of practical use to the system integrator of the IP module.

Hence, we see the practical use of the proposed SLA for both in-house IC design and external IP modules. Finally, we emphasize that the proposed technique is an SLA method and *not* an attack method; the assumption that an attacker needs access to detailed design information is too impractical.

Analysis Targets. In our experiments with side-channel leakage assessment, we target ASCON [9] and XOODYAK [7], two sponge-based ciphers that have been proposed as part of the NIST Lightweight Crypto competition. In contrast to standard block-ciphers, only a limited number of side-channel analysis have been published on sponge-based ciphers.

- A Differential Power Analysis (DPA) on ASCON was demonstrated by Samwel on a Spartan-6 FPGA and required around 40K traces [18]. A machine-learning based attack by Ramazanpour on ASCON required around 24K traces on a Artix-7 FPGA [17].
- A simulated CPA on XOODYAK was demonstrated by Batina *et al.* using 30K traces [2].

In our work we substantially improve upon these earlier results and find a correct key *within a few hundred traces*. The authors of XOODYAK argue that the design

has several built-in features against DPA, including slow absorption of the nonce, key rolling, and ratchetting of the internal state [7]. Our assessment only assumes that the XOODYAK design can be restarted, each time with a different controlled nonce.

We use a software implementation of ASCON and XOODYAK on RISC-V (RV32IMC) processor implemented in 180nm CMOS standard cells with on-chip memory. Because this chip is an in-house design, we have access to the netlist of the chip and we can establish a precise cycle-by-cycle correspondence between gate-level simulated (pre-silicon) and measured (post-silicon) power traces.

To evaluate the SLA on our ASCON and XOODYAK implementations, we use a combination of non-profiled and profiled techniques [15]. In addition to the proposed transfer-learning technique, we use signal-to-noise ratio (SNR) analysis, correlation power analysis (CPA), template attack (TA) and standard deep learning analysis with a CNN. We measure the efficiency of the SLA through the key rank or the measurements to disclosure (MTD) for a known key. We acknowledge that test vector leakage assessment (TVLA) is a popular side-channel leakage assessment technique, but we use an assessment that also shows how efficiently the key can be recovered (which is not possible using TVLA alone).

Contributions of the Paper. We perform side-channel leakage trace collection for ASCON and XOODYAK using power simulation (pre-silicon) and measurement (post-silicon). We then present a side-channel leakage assessment using SNR, CPA, TA and CNN. For each case, we compare the pre-silicon simulation result to the post-silicon measurement result. We present a novel transfer learning technique from the pre-silicon threat model to the post-silicon threat model to improve the deep learning assessment. We analyze the assessment complexity and time complexity for all of the above cases.

Organization of the Paper. In Sect. 2 we summarize the implementation details of ASCON and XOODYAK for the RISC-V processor. Section 3 presents a traditional side-channel vulnerability analysis of ASCON and XOODYAK in terms of SNR, CPA and TA. Section 4 describes the CNN assessment and our new transfer learning technique. Section 5 summarizes and analysis the experimental results. We then conclude the paper in Sect. 6.

2 Preliminaries

In this section, we define the metrics used for SLA, and we describe to test setup of pre- and post-silicon SLA.

2.1 Side-channel Leakage Assessment Metrics

We rely on the following well known metrics [15].

- The SNR for simulated SLA is defined as the ratio of the data variance to the algorithmic noise variance, whereas the SNR for measured SLA is defined as the ratio of the data variance to the algorithmic and measurement noise variance.

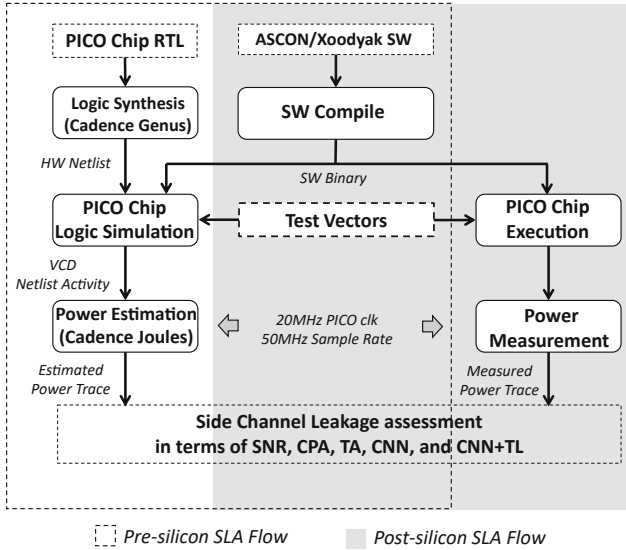


Fig. 2. Implementation flow and test set-up for Pre- and Post-Silicon side channel leakage assessment.

- The *key rank* of a key $k \in K^m$ is defined as the number of keys with a probability greater than k [13]. In SLA, the key rank of the known key k_0 reflects how much information is disclosed under a given assessment method.
- MTD denotes the number of traces required to reduce the key rank of a known key k_0 to 1.
- Pearson’s Correlation coefficient is used to correlate measured and hypothetically modelled power consumption (P_{msd} and P_{hyp}) and compute a correlation for each key k . In SLA, the MTD is reached when the known key k_0 ’s correlation coefficient becomes maximal among all $k \in K^m$.

2.2 Target Platform for SLA

The transfer learning is based on the combination of pre-silicon simulation results with post-silicon measurements of the same design. The target is a small SoC based on the open-source PicoRV RISC-V core. The chip uses 180nm TSMC standard cells and includes 64 KB of on-chip RAM to hold variables. The instructions are fetched from an off-chip serial flash chip (QSPI). For this implementation, we have created an SLA flow that can analyze the implementation either in pre-silicon context starting from the design files, or else in post-silicon context starting from a prototype chip implementation (Fig. 2). Both flows lead to traces that can be compared regardless of their origin. The simulation and measurement setup use a common chip clock (20 MHz) and a common power sample rate clock (50 MHz).

Pre-Silicon SLA Flow. In a pre-silicon setting, power-based side-channel leakage is simulated on a post-synthesis netlist of the design. Initially, we write

the target as a C program for the PicoRV core, and compile the program using `riscv32-unknown-elf-gcc` (v 10.2.0) compiler without optimization into a binary image. The design is then simulated at gate-level accuracy while collecting toggle traces (VCD) for every net. We then use Cadence Joules (RTL Power Solution, Version v20.11-s001_1) and a Skywater 130nm standard cell library to compute frame-based power estimation for the complete netlist using the toggle traces and the post-synthesis netlist. This simulation and power estimation is repeated for every test vector in the side-channel measurement campaign.

Post-Silicon SLA Flow. In a post-silicon setting, the same binary is run on the actual chip while we captured power-based side-channel leakage through a Lecroy Waverunner 7 oscilloscope. We filtered the side channel leakage signal using a 100 KHz - 30 MHz minicircuits bandpass filter before digitizing. To mark the region of interest for side-channel analysis, we instrumented the C program with GPIO triggers. The same method is used for simulation so that all traces can be aligned.

On the SLA Accuracy of Gate-Level Power Simulation. A power simulation is never fully accurate, so an important question relates to the similarity of simulated and measured power traces. Indeed, a power simulation must make a trade-off between the simulation accuracy and the simulation speed of a model. By increasing modeling detail, the estimated power consumption will be a better approximation of the physical power consumption, while the power simulation speed will drastically decrease. Side-channel leakage originates from *any* data-dependency in the power consumption. As we go down in abstraction level from RTL to transistor, each new abstraction level uncovers additional dependencies. For example, gate-level power models can capture gate drive strength, static power leakage, and IR-drop effects, all of which are invisible at the RTL power model yet contribute data-dependent power dissipation. We rely on gate-level power modeling but accept that some power details, such as parasitic coupling, will be ignored by the simulation. At the time of writing, transistor-level power simulation of a complete cryptographic side-channel assessment cannot yet be completed using a reasonable amount of design power [20].

3 Traditional Side-Channel Vulnerability Analysis

In this section, we capture the SLA of ASCON and XOODYAK using common side-channel leakage assessment tools. We use the analysis of the SNR to establish the leakage point of interest for each target. Then, we perform a CPA and a TA.

3.1 Results Summary

Table 1 summarizes the results for *all* assessment techniques investigated in this contribution, including a non-profiled technique (CPA) and several profiled techniques (TA, CNN). For each of ASCON and XOODYAK, we analyze three cases: SLA using simulated traces, SLA using measured traces, and SLA with the proposed transfer learning technique (TL). We will elaborate on individual result entries in the following subsections.

Table 1. Assessment using MTD metric for all targets. The number of traces shown is the *average* needed to retrieve a key byte. For profiled attacks, the number of traces used for profiling are listed separately.

Primitive	SLA flow	CPA MTD	TA		CNN	
			Profiling (x 1,000)	MTD	Profiling (x 1,000)	MTD
ASCON	Simulated	8	9	2	9	2
	Measured	2,000	90	573	90	500
	TL	–	–	–	19	176
Xoodyak	Simulated	91	19	84	19	60
	Measured	700k	90	520	90	490
	TL	–	–	–	60	170

3.2 Traditional SLA on ASCON

Ascon ASCON-128 is an authenticated-encryption with associated-data primitive which is selected as a finalist in the NIST Lightweight Cryptography competition [10]. ASCON-128 is a duplex-sponge-based construction with four phases of operation: initialization, associated data, plaintext/ciphertext, and finalization. All phases use the same permutation function which includes a constant addition, a substitution layer, and a linear layer. ASCON-128 has 320 bits of state, divided into five double words that hold the 64-bit initialization vector (X_0), the 128-bit key (X_1, X_2) and the 128-bit nonce (X_3, X_4) respectively.

In ASCON’s SLA we aim to demonstrate that the 128-bit key can be recovered at a given number of traces. The controlled variable, required to drive differential power analysis, is the nonce (X_3, X_4). We focus on the non-linear operations in the S-box of ASCON that compute X_1 and X_4 , as expressed in the following Boolean equations. In these equations, the nonce is loaded in (X_3, X_4) and the key is loaded in (X_1, X_2).

$$\begin{aligned}
 X_4 &= (X_4 \oplus X_3) \oplus ((255 \oplus (X_0 \oplus X_4)) \& X_1) \\
 X_1 &= ((X_1 \oplus ((255 \oplus (X_2 \oplus X_1)) \& X_3)) \\
 &\quad \oplus ((X_0 \oplus X_4) \oplus ((255 \oplus X_1) \& (X_2 \oplus X_1))))
 \end{aligned} \tag{1}$$

The target implementation of ASCON is an 8-bit reference implementation in software. Listing 1.1 shows the assembly code to compute X_4 as a byte-wise operation. The point where key and control inputs merge is sensitive to side-channel leakage. The `and` operation on line 16 is the first line where that happens. Subsequent operations, such as on line 18 and 20, are potential targets as well. To understand which of these operations is the best candidate to mount a CPA, we perform SNR analysis on 500 simulated traces (Fig. 3, top) [15]. This analysis shows that the store instruction contributes a greater data-dependent power variation and, therefore, is the proper target for the side-channel leakage assessment.

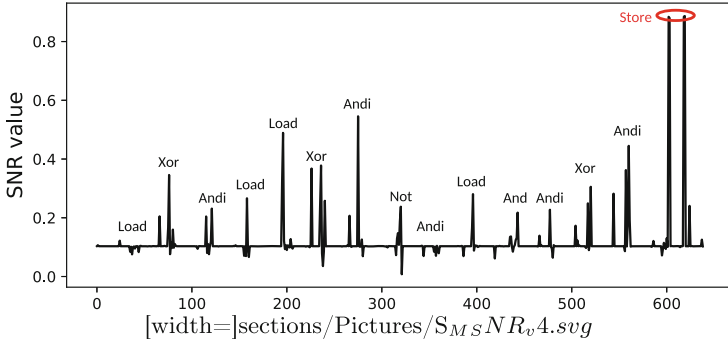


Fig. 3. SNR Analysis of X_4 : (top) SNR on 500 simulated traces to identify leaky instructions (bottom) SNR on 2K simulated traces (black) and 200K measured traces (grey) to estimate illustrate by practical measurement. (Color figure online)

Ascon SNR Analysis. The SNR analysis of Fig. 3, top, demonstrates an important major advantage of simulation-based traces, namely the absence of measurement noise. Figure 3, bottom, compares the SNR of 2K simulated traces to the SNR of 200K measured traces. Both the measured and simulated traces are aligned by making use of a GPIO trigger in the real and simulated ASCON software. The range of the X axis is roughly equivalent to the execution of Listing 1. The X axis spans 640 sample points, which corresponds to $12.8 \mu s$ or 256 cycles. The simulated SNR shows two sharp peaks corresponding to the memory-store operation (Fig. 3, top). However, the SNR on measured traces is much noisier and shows leakage over the last 64 samples of the curve. We attribute these extra leaky points to measurement noise, trigger signal jitter, and possibly an unexplained effect from the off-chip QSPI flash.

Ascon Correlation Power Analysis. Fig. 4 shows the outcome of CPA on ASCON for both a simulated assessment (black) and a measured assessment (grey). Both cases converge at the same key value, although the simulated CPA requires only 8 traces while the measured CPA needs 2,000 traces (Table 1).

The power model of the CPA is the Hamming Weight of X_4 , whose update depends on both the lower half of the secret key K_1 and the controlled nonce. Specifically, with i representing the test vector index, and j denoting the key byte index 0 to 7, we find the following power model.

$$\begin{aligned} X_k^{i,j} &= (N2^{i,j} \oplus N1^{i,j}) \oplus ((255 \oplus (IV^{i,j} \oplus N2^{i,j})) \& K1_k^{i,j}) \\ P_{hyp} &= HW[(X_k^{i,j})] \end{aligned} \quad (2)$$

The correlation of the power model with the power traces then leads to the value of K_1 . After K_1 is found, its value is used to mount a CPA on the value of X_1 which combines both the upper half K_2 and the lower half K_1 of the secret key. This leads to the value of K_2 .

Listing 1.1. Portion of the SBOX computation of ASCON. Instructions highlighted in blue are potential targets for CPA.

```

1  lui      a5,0x30005
2  addi    a5,a5,8
3  li      a4,1
4  sw      a4,0(a5)          // GPIO trigger up
5  lbu     a4,-52(s0)
6  lbu     a5,-60(s0)
7  xor     a5,a5,a4          // a4 <- X3^X4
8  andi    a4,a5,255
9  lbu     a3,-28(s0)
10 lbu     a5,-52(s0)
11 xor     a5,a5,a3          // a5 <- (X4^X0)
12 andi    a5,a5,255
13 not     a5,a5            // a5 <- (255^(X4^X0))
14 andi    a3,a5,25
15 lbu     a5,-36(s0)
16 and     a5,a5,a3          // a5 <- (255^(X4^X0))&X1
17 andi    a5,a5,255
18 xor     a5,a5,a4          // a5 <- (X3^X4)^(255^(X4^X0))&X1
19 andi    a5,a5,255
20 sb      a5,-52(s0)        // store X4
21 lui     a5,0x3000
22 addi    a5,a5,8
23 sw      zero,0(a5)        // GPIO trigger down

```

Ascon Template Attack. A template attack is a well known profiled attack [6]. It uses a profiling phase to compute a template, a set of probability distributions that describe how the power traces vary for many different keys. Then, in the testing phase, it estimates the probability distribution of the target and finds the best matching distribution from the template. This leads to the unknown key. The template is computed over a limited number of point of interest (POI) in the trace. In our ASCON Template Attack, we select 15 POIs among 640 possible trace points. We build the profile on the Hamming Weight of X_4 , computing the mean and covariance matrix for each Hamming Weight Value. Because of the profiling phase, a template attack can outperform a CPA. Table 1 demonstrates that the ASCON key is extracted using just 2 simulated power traces, or 573 measured power traces.

3.3 Traditional SLA on XOODYAK

XOODYAK is an authenticated-encryption with associated-data primitive which is also selected as a finalist in the NIST Lightweight Cryptography Competition [7]. Like ASCON, XOODYAK is based on duplex-sponge construction which allows its use in multiple symmetric-key applications. The XOODYAK design is inspired by the KECCAK round permutation. The assessment target in XOODYAK is the θ function which adds the key K , the nonce N and a counter C as $X = K \oplus N \oplus C$. In this expression, the nonce and the counter are the controlled variables. The assessment of XOODYAK is harder than that of ASCON for two reasons. First, the

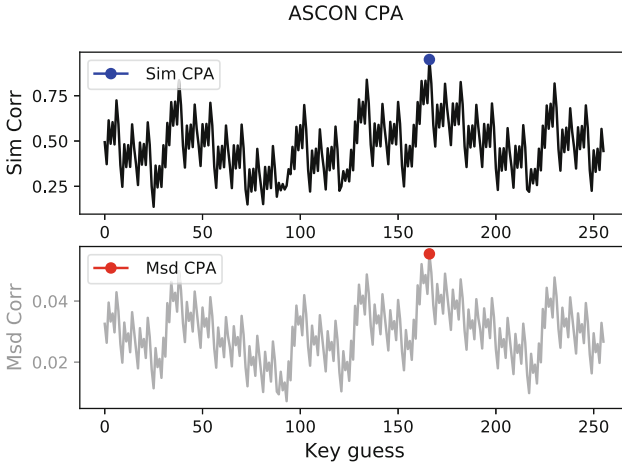


Fig. 4. ASCON: Correlation Power Analysis on simulated (black) and measured (grey) traces. (Color figure online)

XOR operation which combines the controlled variables with the key is linear. Since $A \oplus B = \bar{A} \bar{B}$, this leads to so-called ghost-peaks of equally-likely keys in the assessment [4]. Second, our specific implementation of XOODYAK is implemented on a 32-bit wordlength which combines 4 different key bytes in a single 32-bit RISC-V instruction. Hence, the XOODYAK traces will have a higher level of algorithmic noise. Listing 2 shows the relevant portion of the XOODYAK implementation under consideration for SLA. The `xor` operation on line 7 is a potential target, as well as the dependent `xor` on line 9 and the store-word instruction on line 10.

Xoodyak SNR Analysis. Because a single execution of Listing 2 computes on four different key bytes, one can compute four different SNR curves for a single set of power traces. Figure 5a shows the SNR on 10K simulated traces. Its X-axis corresponds roughly to the execution of Listing 2, and we find that leakage is concentrated in a few power samples. Similar to the analysis on ASCON, we find the store-word instruction to be a dominant contributor to data-dependent power dissipation. The same SNR curve is also computed on 1500K measured traces as shown in Fig. 5b. Using a common GPIO trigger, we are able to align the SNR analysis of the simulated traces to the measured traces. Because of the high level of algorithmic noise, the resulting SNR is extremely noisy. We mark the last 100 samples of the measurement window as containing leaky samples in SLA.

Xoodyak Correlation Power Analysis. XOODYAK’s CPA uses a Hamming Weight power model on $P_x^{i,j}$, where x denotes a word index range from 0 to 3, i represents the test vector, and j denotes the key byte index range from 0 to 3. $P_x^{i,j}$ depends on the lower half of the secret key $K_x^{i,j}$, the controlled nonce $N_x^{i,j}$ and counter value $C_x^{i,j}$. We find the following power model.

Listing 1.2. Portion of the θ computation of XOODYAK. Instructions highlighted in blue are potential targets for CPA.

```

1  lui    a5,0x3000
2  addi   a5,a5,8
3  li     a4,1
4  sw     a4,0(a5)           // GPIO trigger up
5  lw     a4,-24(s0)
6  lw     a5,-20(s0)
7  xor    a5,a5,a4           // a5 <- K[0:3]^N[0:3]
8  lw     a4,-28(s0)
9  xor    a5,a5,a4           // a5 <- K[0:3]^N[0:3]^C[0:3]
10 sw    a5,-24(s0)         // store X
11 lui    a5,0x3000
12 addi   a5,a5,8
13 sw     zero,0(a5)        // GPIO trigger down

```

$$HW[P_x^{i,j}] = HW[K_x^{i,j} \oplus N_x^{i,j} \oplus C_x^{i,j}] \quad (3)$$

Correlating the power model and the power traces yields the subkey of K_0 . Figure 6 shows a correlation plot of the XOODYAK CPA. Two peaks are found, one on the true key byte (253) and one on the complementary key byte (2). Both the simulated and measured correlation plot are similar, even though the measured plot requires 700K traces due to the noisy SNR.

Xoodyak Template Attack. The template attack on XOODYAK proceeds as on ASCON, and builds the template on the Hamming Weight of the θ function output. Table 1 shows that the key is extracted on 84 simulated power traces or 520 measured power traces.

4 Deep Learning Assisted Side Channel Analysis

We now develop the transfer learning technique as an extension of deep learning based side-channel vulnerability analysis.

4.1 Deep Learning SLA on ASCON

Ascon CNN Development. The network architecture and hyperparameter selection play an important role in successful adversarial threat modeling [16]. The CNN for a single ASCON keybyte consists of a feature extractor and a 256-class classifier. The input to the CNN is a window of 64 power samples, selected through the SNR analysis of Fig. 3, bottom. A convolutional layer extracts specific features, similar to POIs, from the power samples. Next, the dense layers map the variation within and across different traces into a set of 256 probabilities. Batch normalization transforms the output of a previous layer by subtracting the batch mean and dividing by the batch standard deviation. Dropouts are used to randomly turn off a percentage of the network’s neurons in order to

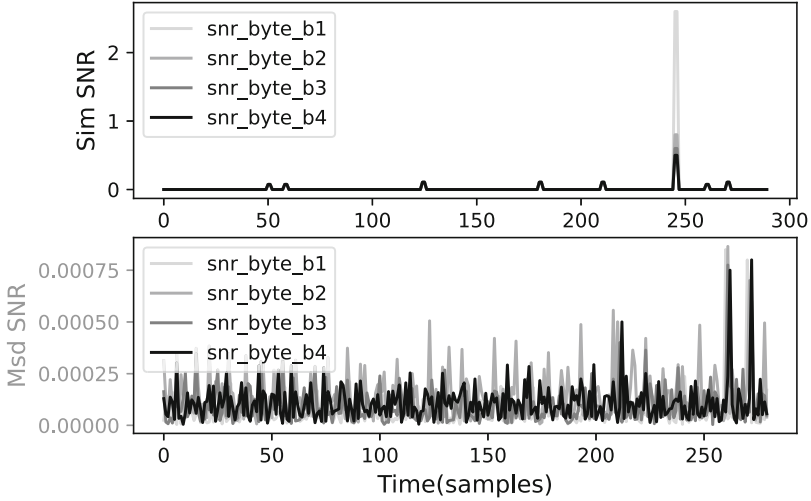


Fig. 5. XOODYAK: (a) SNR on 10K simulated traces (b) SNR on 1500K measured traces

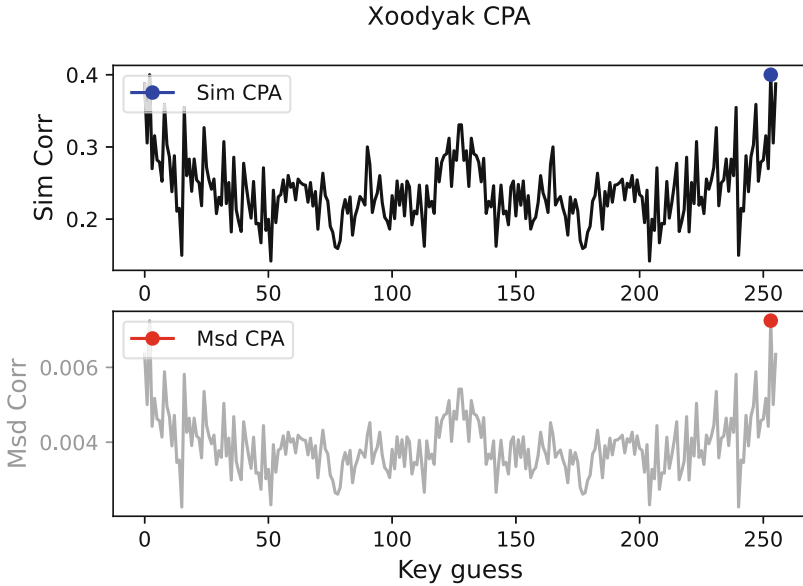
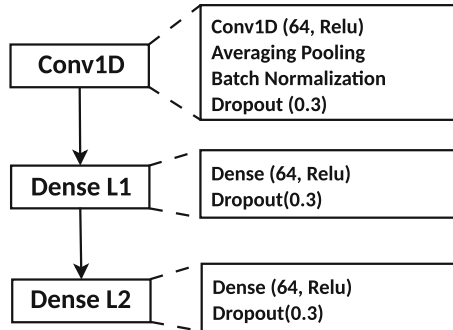


Fig. 6. XOODYAK: Correlation Power Analysis on simulated (black) and measured (grey) traces. (Color figure online)

improve the model’s learning. Figure 7 shows our network and its hyperparameters. We adopted the ASCAD network [3] and optimized it for ASCON using random search over the hyperparameters provided in Table 2. The resulting sim-

Table 2. Hyperparameter search space for ASCON CNN. We selected the best hyperparameter (fit) through exhaustive search of the search space.

Hyperparameter	Ranges		
	Min	Max	Fit
Batch size	50	200	50
Convolution layers	1	5	1
Kernel size	1	11	3
Stride	1	4	1
Dense layers	1	3	2
Neurons	10	256	64
Learning rate	0.00001	0.001	0.001
Epochs	50	500	200
Drop out	10%	30%	30%
	Options		Fit
Pooling type	(Average,Max)		Average
Optimizer	(Adam, RMSprop)		Adam
Activation function	(ReLU,SeLU)		ReLU

**Fig. 7.** ASCON: Convolutional Neural Network architecture for adversarial threat model of simulated and measured traces

ulated model has an accuracy of 94%, whereas the measured model and transfer learning model are close to each other (82% and 81% respectively).

Ascon Transfer Learning. We now apply transfer learning and demonstrate a reduction in learning time as well as in assessment effort. The idea is to transfer a part of the pre-silicon threat model to the post-silicon threat model. Post-silicon traces are noisy, which means that a large amount of traces are needed to learn the threat model at a high learning cost. Pre-silicon simulations are slow, but the pre-silicon traces are noiseless and a threat model can be learned from them quickly using much fewer traces.

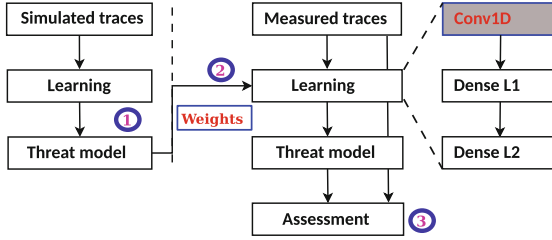


Fig. 8. Transfer Learning: (1) training from simulated traces, (2) transfer learning on measured traces keeping the convolutional layer frozen, and (3) assessment using the transfer-learned CNN.

Table 3. Three test cases are investigated over simulated and transfer learning models. For each test case, corresponding simulated model weights are used for transfer learning

Primitive	SLA of Sbox X_4	Simulated			Transfer		
		Profiling (x 1,000)	MTD	Accuracy	Profiling (x 1,000)	MTD	Accuracy
ASCON	Test case_1	5	11	94%	19	191	80%
	Test case_2	10	2	94%	40	176	81%
	Test case_3	20	2	94%	60	162	82%

Figure 8 illustrates the proposed transfer learning. First, we perform deep learning SCA on the simulated traces to identify the architecture, hyperparameters and weights. Next, we continue learning with these parameters on the measured traces. In the second phase, the convolutional layer remains frozen. This keeps the feature extraction layer unchanged, while the other layers maintain trainable parameters for the classification. Finally, we perform assessment on the measured traces using this new network created from transfer learning.

Table 3 represents the number of profiled traces against the number of test traces (MTD) for the CNN on simulated and transfer learning on measured traces. Here, three test cases are used to demonstrate different trade-offs between profile learning and testing. In the Table 4, we calculated the number of test traces against the number of profiled traces for measured traces. Using transfer

Table 4. On average, transfer learning model requires 1.97 times less profiling traces and 2.85 times less testing traces compare to measured learning model

Primitive	SLA of Sbox X_4	Measured		
		Profiling (x 1,000)	MTD	Accuracy
ASCON	Test case_1	45	521	80%
	Test case_2	90	491	82%
	Test case_3	100	490	82%

Table 5. Two test cases are investigated over simulated and transfer learning models. For each test case, corresponding simulated model weights are used for transfer learning

Primitive	SLA of Linear(θ)	Simulated			Transfer		
		Profiling (x 1,000)	MTD	Accuracy	Profiling (x 1,000)	MTD	Accuracy
Xoodyak	Test case_1	18	56	93%	45	160	80%
	Test case_2	19	60	94%	60	170	81%

Table 6. On average, transfer learning model requires 2.87 times less testing traces compare to measured learning model

Primitive	SLA of Linear (θ)	Measured		
		Profiling (x 1,000)	MTD	Accuracy
Xoodyak	Test case_1	80	486	81%
	Test case_2	90	494	83%

learning, we obtain faster learning because we need to process fewer traces. Moreover, we need fewer test traces to assess the design. Overall, the accuracy for simulated, transfer and measured are 94%, 81% and 82% respectively.

Figure 9 displays 16 subplots corresponding to the 16 key bytes of ASCON. Each subplot represents convergence of the key rank of the measured and transfer learning model. A major rank comparison between the transfer and the measured learning model in the convergence region shows that the model on measured traces lags by 42 ranks on average. This indicates that transfer learning models provide a gain of 5 to 6 bits in guessing entropy.

4.2 Deep Learning SLA on XOODYAK

Xoodyak CNN development We adopted the same architecture as in Fig. 7 with the following changes. First, all layers use batch normalization and dropout (0.3). Second, the learning rate is fine-tuned to 0.0001.

Xoodyak Transfer Learning. Table 5 compares the CNN performance for simulated and transfer learning on measured traces. From Table 6, it is clear that, transfer learning model (TL) requires 1.61 and 2.88 times less profile and test traces compare to measured model (CNN). Once again, transfer learning achieves faster learning and shorter evaluation.

Similar to ASCON, transfer learning model of Xoodyak converge 68 rank faster compare to measured model as given in Fig. 10.

5 Analysis of Results

Finally, we compare the performance of the proposed transfer learning technique to classic SLA as well as deep learning SLA.

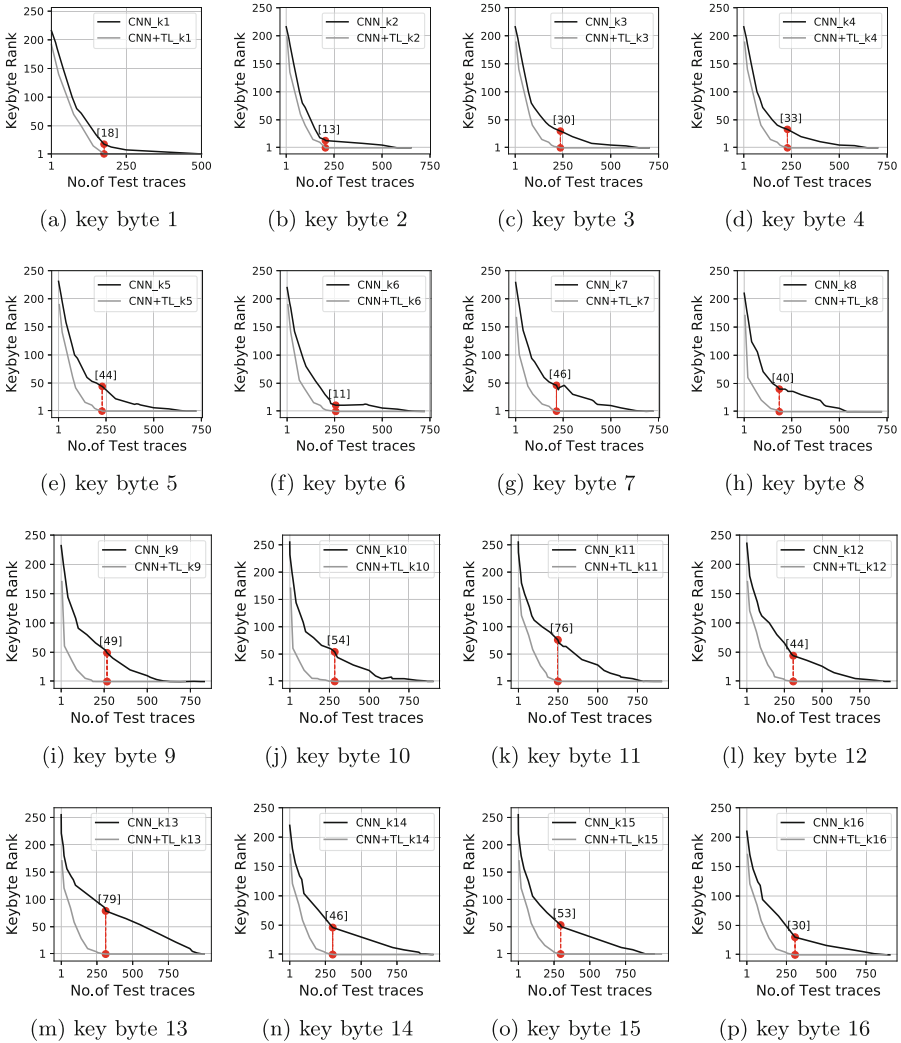


Fig. 9. The red color highlighted in the subplots indicates that there is a difference in key byte rank between measured(CNN) and transfer(CNN+TL), when CNN+TL converges to rank zero. (Color figure online)

Assessment Complexity. We summarize the experiments on transfer learning with simulated traces as follows. First, it is clear that the proposed transfer learning method outperforms all other assessment we tried. Table 7 expresses the relative assessment gain over CPA. This is the ratio of the number of traces required to reveal a key byte using a chosen assessment over the number of traces required using CPA. For the transfer learning method, the gain goes up to 4,100x for a *noisy* target. This is not unexpected since noisy traces are a

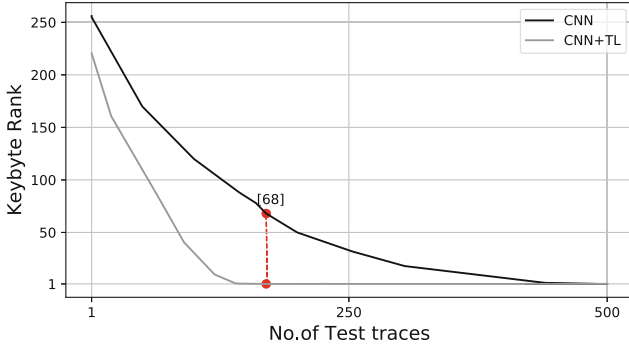


Fig. 10. XOODYAK : Key rank converge of transfer learning (CNN+TL) is 68 ranks faster than measured (CNN)

Table 7. Relative Assessment gain in number of traces, using Correlation Power Analysis as the reference. A Gain of N means that N times more traces are needed in CPA, so higher is better. The transfer learning method outperforms the best template attack as well as the measurement-only deep learning method.

Assessment	Relative Assessment Gain on CPA	
	ASCON	XOODYAK
CPA	1	1
Template Attack	3.4	1,300
CNN	4	1,400
CNN+TL	11.4	4,100

harder training target for the deep learning threat model. Second, it is clear that the proposed transfer learning method is much less sensitive to distortions from the measurement setup than any other attack. Table 8 expresses the relative assessment loss for each assessment, which is measured as the increase in number of traces for an attack when moving from simulated traces to measured traces. The transfer learning method shows the lowest relative assessment loss among all assessments.

Time Complexity. There are two dimensions in the analysis of time complexity of the proposed technique. One dimension quantifies the difference between simulating a power trace, versus capturing a power trace from a real chip. The second dimension quantifies the cost of SLA on the collected power traces. We perform all simulation and SLA experiments on an Intel Xeon Gold 6248 server. The power simulation for one power trace of ASCON took approximately 5 min, which can be shortened to 30 s per simulated trace by running 10 parallel simulation threads. In contrast, capturing a trace took from a real chip took 0.15 s, so that the measurement of traces is 200 times faster than their gate-level simulation. Hence, we confirm that power simulation time remains a dominant portion

Table 8. Relative assessment loss in number of traces, when comparing assessments on simulated traces to assessments on measured traces. A Loss of N means that N times more traces are needed on the measurements, and lower is better. The transfer learning method outperforms all other methods for both ASCON and XOODYAK.

Assessment	Relative Assessment Loss over Sim	
	ASCON	XOODYAK
CPA	250	7600
Template Attack	136	6.2
CNN	250	8.2
CNN+TL	88	2.8

Table 9. Time complexity of all side-channel vulnerability analysis for ASCON. # = number of traces needed, AT = Attack Time, LT = Learning Time.

Primitive	SLA flow	CPA		TA			CNN		
		#	AT	#	LT	AT	#	LT	AT
ASCON	Simulated	8	< 1m	9K	10m	5m	9K	50m	10m
	Measured	2k	< 10m	90K	30m	20m	90K	6hr	20m
	TL	-	-	-	-	-	19K	60m	15m

in data collection. Table 9 shows the time complexity of CPA, TA and CNN. Each experiment lists the number of traces required and the associated learning time and attack time. The assessment part of the transfer learning method is competitive with traditional (measurement-based) CNN, as it completes the task in 60+50 min as opposed to 6 h. XOODYAK has a similar pattern of time complexity. Our machine learning experiments are running on a traditional CPU configuration (without GPU), which makes them relatively slow compared to some published results [11].

6 Conclusion

This work shows that transfer learning based side channel analysis on post-silicon using a pre-silicon threat model. The proposed technique evaluates the design by 2.87 times fewer traces compared to the Naive CNN technique. We are considering further improvements to our method, such as using techniques to understand and eliminate noise and distortions on measured traces. This material is based upon work supported by the National Science Foundation under Grant No. 1931639.

References

1. Arsath K F, M., Ganesan, V., Bodduna, R., Rebeiro, C.: PARAM: a microprocessor hardened for power side-channel attack resistance. In: 2020 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), pp. 23–34 (2020). <https://doi.org/10.1109/HOST45689.2020.9300263>
2. Batina, L., et al.: Side-Channel evaluation report on implementations of several NIST LWC finalists (August 2022). <https://hdl.handle.net/2066/253567>
3. Benadjila, R., Prouff, E., Strullu, R., Cagli, E., Dumas, C.: Deep learning for side-channel analysis and introduction to ASCAD database. *J. Crypt. Eng.* **10**(2), 163–188 (2020)
4. Brier, E., Clavier, C., Olivier, F.: Correlation power analysis with a leakage model. In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 16–29. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-28632-5_2
5. Buhan, I., Batina, L., Yarom, Y., Schaumont, P.: SoK: design tools for side-channel-aware implementations. In: Suga, Y., Sakurai, K., Ding, X., Sako, K. (eds.) ASIA CCS 2022: ACM Asia Conference on Computer and Communications Security, Nagasaki, Japan, 30 May 2022–3 June 2022, pp. 756–770. ACM (2022). <https://doi.org/10.1145/3488932.3517415>
6. Chari, S., Rao, J.R., Rohatgi, P.: Template attacks. In: Kaliski, B.S., Koç, K., Paar, C. (eds.) CHES 2002. LNCS, vol. 2523, pp. 13–28. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36400-5_3
7. Daemen, J., Hoffert, S., Peeters, M., Van Assche, G., Van Keer, R.: Xoodyak, a Lightweight Cryptographic Scheme. *IACR Transactions on Symmetric Cryptology*, pp. 60–87 (2020)
8. Das, D., Golder, A., Danial, J., Ghosh, S., Raychowdhury, A., Sen, S.: X-DeepSCA: cross-device deep learning side channel attack. In: Proceedings of the 56th Annual Design Automation Conference 2019, DAC 2019, Las Vegas, NV, USA, June 02–06, 2019, p. 134. ACM (2019). <https://doi.org/10.1145/3316781.3317934>
9. Dobraunig, C., Eichlseder, M., Mendel, F., Schl  ffer, M.: Ascon v1.2. Submission to Round 1 of the NIST lightweight cryptography project (2019). <https://csrc.nist.gov/CSRC/media/Projects/Lightweight-Cryptography/documents/round-1/spec-doc/ascon-spec.pdf>
10. Gross, H., Wenger, E., Dobraunig, C., Ehrenh  fer, C.: Suit up!-made-to-measure hardware implementations of ASCON. In: 2015 Euromicro Conference on Digital System Design, pp. 645–652. IEEE (2015)
11. Ito, A., Saito, K., Ueno, R., Homma, N.: Imbalanced data problems in deep learning-based side-channel attacks: analysis and solution. *IEEE Trans. Inf. Forensics Secur.* **16**, 3790–3802 (2021)
12. Kiaei, P., Schaumont, P.: SoC Root Canal! Root cause analysis of power side-channel leakage in system-on-chip designs. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2022**(4), 751–773 (2022). <https://doi.org/10.46586/tches.v2022.i4.751-773>
13. Martin, D.P., Martinoli, M.: A note on key rank. *Cryptology ePrint Archive*, Paper 2018/614 (2018). <https://eprint.iacr.org/2018/614>
14. Paguada, S., Batina, L., Buhan, I., Armendariz, I.: Playing with blocks: toward re-usable deep learning models for side-channel profiled attacks. *IEEE Trans. Inf. Forensics Secur.* **17**, 2835–2847 (2022). <https://doi.org/10.1109/TIFS.2022.3196273>
15. Papagiannopoulos, K., Glamocanin, O., Azouaoui, M., Ros, D., Regazzoni, F., Stojilovic, M.: The side-channel metric cheat sheet. *IACR Cryptol. ePrint Arch.* p. 253 (2022). <https://eprint.iacr.org/2022/253>

16. Picek, S., Perin, G., Mariot, L., Wu, L., Batina, L.: SoK: deep learning-based physical side-channel analysis. *IACR Cryptol. ePrint Arch*, p. 1092 (2021). <https://eprint.iacr.org/2021/1092>
17. Ramezanpour, K., Abdulgadir, A., Diehl, W., Kaps, J.P., Ampadu, P.: Active and passive side-channel key recovery attacks on ASCON. In: *Proceedings of the NIST Lightweight Cryptogr. Workshop*, pp. 1–27 (2020)
18. Samwel, N., Daemen, J.: DPA on hardware implementations of Ascon and Keyak. In: *Proceedings of the Computing Frontiers Conference*, pp. 415–424 (2017)
19. Shelton, M.A., Chmielewski, L., Samwel, N., Wagner, M., Batina, L., Yarom, Y.: Rosita++: automatic higher-order leakage elimination from cryptographic code. In: *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pp. 685–699. CCS 2021, Association for Computing Machinery, New York, NY, USA (2021). <https://doi.org/10.1145/3460120.3485380>
20. Šijačić, D., Balasch, J., Yang, B., Ghosh, S., Verbauwhede, I.: Towards efficient and automated side-channel evaluations at design time. *J. Crypt. Eng.* **10**(4), 305–319 (2020). <https://doi.org/10.1007/s13389-020-00233-8>
21. Thapar, D., Alam, M., Mukhopadhyay, D.: Deep learning assisted cross-family profiled side-channel attacks using transfer learning. In: *22nd International Symposium on Quality Electronic Design, ISQED 2021, Santa Clara, CA, USA, April 7–9, 2021*, pp. 178–185. IEEE (2021). <https://doi.org/10.1109/ISQED51717.2021.9424254>
22. Wang, H., Brisfors, M., Forsmark, S., Dubrova, E.: How diversity affects deep-learning side-channel attacks. In: Nurmi, J., Ellervee, P., Halonen, K., Rönning, J. (eds.) *2019 IEEE Nordic Circuits and Systems Conference, NORCAS 2019: NORCHIP and International Symposium of System-on-Chip (SoC)*, Helsinki, Finland, October 29–30, 2019, pp. 1–7. IEEE (2019). <https://doi.org/10.1109/NORCHIP.2019.8906945>