Power Swapper: Approximate Functional Block Assisted Cryptosystem Security

Abhijitt Dhavlle*, Setareh Rafatirad†, Houman Homayoun† and Sai Manoj Pudukotai Dinakarrao*
*George Mason University, Fairfax, VA, USA; †University of California, Davis, CA, USA
Email: {adhavlle, spudukot}@gmu.edu, {srafatirad, hhomayoun}@ucdavis.edu

2021 IEEE 34th International System-on-Chip Conference (SOCC) | 978-1-6654-2931-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/SOCC52499.2021.9739304

***Abstract*—The hardware security domain in the recent years has experienced a plethora of threats of which side-channel attacks (SCAs) has been one of the emerging threats. These SCAs function by exploiting the side-channels which invariably leak important data during an application's execution. The information leaked through side-channels are inherent characteristics of the system and is often unintentional. This information can be microarchitectural or physical information such as power consumption, thermal maps, timing of the operation, acoustics, and cache-trace. Intercepting secret information based on the study of power signature is a subdivision of SCAs where power consumption information serves as a covert channel leaking crucial information about the executed operations. Such physical SCAs are known to be a significant threat to cryptosystems such as AES (Advances Encryption Standard) and can reveal the encryption key efficiently. To overcome such concerns and protect the data integrity, we introduce *Power Swapper* in this work. The proposed Power Swapper thwarts the attack by randomly choosing one of the multiple modules that perform the intended activity, but have power consumption different than a standard implementation and can lead to similar power consumption as one of the other modules that perform a different operation. To achieve this, we introduce carefully crafted swapping of the standby modules that are responsible for the AES operation thus deluding the attacker without hurting the crypto operations. To minimize the overheads, we design reconfigurable computing elements that can perform a given operation, but with different power consumption depending on the configuration. This methodology has been validated for the AES power analysis attack and the key information observed by the attacker is seen to be completely futile, indicating the success of the proposed method.**

***Index Terms*—Physical side-channels; Hardware security; Cryptography; Power analysis.**

I. Introduction

Data integrity and security became an essential part in the era of digital systems where privacy and confidentiality needs to be ensured. There have been a plethora of works addressing the attacks on systems, like those posed by malware [1]–[4], reverse engineering of hardware [5], [6]; attacks on machine-learning assisted hardware-based malware detectors (HMDs) [7], [8], adversarial attacks on machine learning [9], cache based side-channel attacks [10]–[12], etc. Of these, side-channel attack and cryptosystem has been discussed in this work. To prevent such attacks, cyrptographic mechanisms are employed to offer security to the data by encrypting the data streams with a secret key and transform the data into a human non-readable format. The attempt to exercise a brute force to decrypt the information is exhaustive and can even be unfeasible. To efficiently decode the secret key and decrypt the information, adversaries target utilizing the information obtained through side-channels, termed as side-channel attacks. Side-channels are inherent in any given design and side-channel attacks exploit the information from these rather than exploiting vulnerabilities in the software. There exist both physical and microarchitectural side-channels that can leak secure critical information through acoustics, electromagnetic (EM) radiations, power trace, thermal maps and cache-access information. Power signature based side-channel threats are a pivotal threat as power consumption is an inherent and preliminary characteristic of any digital system.

In this work we consider a power signature based side-channel attack on encryption algorithm executing on FPGAs as they are proliferating into data centers for compute-intensive operations such as encryption. For the power analysis based SCA to be successful, the attacker measures the power traces from the system while triggering crypto operations on the system. This trace is then studied statistically to deduce the secret key. The fundamental principle underlying this attack is that different modules (operations) of AES consume different power, and thereby studying the power trace reveals the operation, based on which the secret key can be deduced.

Pengyuan Yu et al. in [13] propose an intelligent place-and-route technique to facilitate symmetrical routing as a defense against power analysis SCA on FPGA. Work in [14] describes how a circuit can be transformed to a larger circuit to defend against probe-based physical SCAs, but, the technique proposed is very complex. Work in [15] and [16] describes algorithmic countermeasures to thwart SCAs which attempts to minimize the correlation between the intermediate values and the secret key ;and by algorithmically adding noise respectively. Also, circuit-level countermeasures are presented in papers [17]–[21]. It is observed that the existing defenses require modifications in physical designs, leading to larger overheads and design complexity.

To overcome these challenges and defend against power analysis SCA, we propose Power Swapper. More details of the proposed Power Swapper is presented below.

II. Power Measurement and Correlation Power Analysis

The setup we harnessed for measuring the power has been described in this section followed by a brief introduction to the process of CPA (Correlation Power Analysis) analysis for key extraction.

a) FOBOS [22], [23] Setup: : The setup has been built specially for purposes that require measuring FPGA core power for physical side-channel attack analysis. The setup has been termed as FOBOS (Flexible Open-source workBench fOr Side-channel analysis) [22], [23]. Figure 1 shows the block diagram of the FOBOS setup. The Controller is an Artix-7 based FPGA that receives the test vectors from the PC and communicates with the DUT FPGA (Device Under Test) which is the target FPGA platform running the AES implemenatation. The target initiates instances of the AES cryptosystem and delivers the results to the controller. Meanwhile, the controller also triggers picoscope measurement cycle at the same time as the AES and delivers the measured power and the cipher text to the PC. The picoscope captures the entire trace of the AES cycle and keeps iterating for every new AES cycle. The PC runs the Python scripts that are responsible for sending the test vectors and key (secret key) to the controller and accumulating all the results in a numpy array. The FOBOS is completely reconfigurable to suit the specific needs of the measurements and application. The setup has been described in more detail in [22], [23].

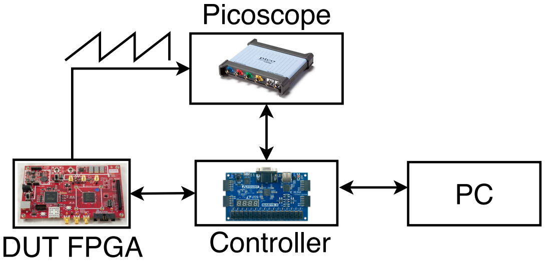

Fig. 1. In-house setup for measuring the power of the DUT running AES implementation [22], [23]

b) Attack Model: : The attack model and the assumptions for a successful attack have been described below:

1) Physical access: The adversary needs to have physical access to the cryptosystem for obvious reasons - the CPA analysis needs power traces as one of its inputs to calculate key bytes. The physical access taps the power input to the FPGA so the traces can be captured.
2) Access to PT: The attacker also has access to the plaintext(PT) which is used by the system.
3) Access to the AES implementation: The adversary needs to have an idea of how the AES has been implemented internally. This is needed to choose the appropriate attack point and decide whether only PT and power traces are sufficient to succeed the attack phase.
4) AES timing: It is helpful for the attacker to have access to the time it takes for the intermediate values to be processed and be available at the output of the point of attack. This has been discussed in the next section.

c) CPA Analysis: : Figure 2 illustrates the process of how CPA analysis is performed on the system to derive the correct combination of key input. In our design, the length of the plain text, cipher text and the key is 128 bits wide. The adversary must try to derive as much correct key bits as possible to break the cryptosystem security. Once the correct key has been derived, the adversary gets access to the systems where the same key was used for delivering security. There are some assumptions that are precursory to the tampering of the system which have been discussed previously. The CPA attack is one of the ways in which an adversary can gain access to the AES key. Referring to Figure 2, the attacker begins by deciding the point of attack. The point of attack is selected such that the value (output) available relates to the combination of the plaintext and the key (the attacker does not have access to the key, partial or whole). The block in conventional AES implementation that is chosen is the 'sbox' aka substitution box. The contents of this lookup table is open sourced and hence even the attacker has access to it. As discussed previously, the attacker has access to the time it takes for the data to reach the point of attack or the sbox in this case. Proceeding further, the power to the DUT FPGA is measured and stored. Power values corresponding to one full AES cycle are known as samples, whereas, the individual runs of the AES (with different test vector, with the same key) are known as traces. The CPA then involves calculating the hypothetical intermediate and the hypothetical power. The output of a sbox block is tried to mimic here to calculate the hypothetical power. Thing to note here is the attacker has no access to the actual key used and hence, it tries to generate all possible values (typically it is done byte wise, so a total of 256 possible values). After the output value of the sbox is known, by a combination of guessed key and plaintext, hamming weight or distance is calculated to represent power. This hypothetical power and the actual measured power are then correlated to see which power output value (hypothetical) corresponds strongly with the actual measured value and the correct key is the one that corresponds to that hypothetical value calculated previously. By iterating through this process a number of times the full key is derived.

III. Power Swapper :Implementation and Results

Our proposed Power Swapper has been outlined in Figure 3 where part (a) shows the internal structure of a conventional FPGA cryptosystem where each module has its own power consumption rating. The attacker then performs the power analysis on the system and then through statistical methods the adversary tries to deduce the secret key information. This is possible due to the fact that the instantaneous power consumption value would correspond to the operation of module. Based on these sequence of operations, the attacker can deduce the secret information. As shown in Figure 3(c), the power consumption waveform has seven peaks in total each corresponding to some operation. For instance, peak 1 and 5 have the same magnitude and inferred to belong to the same operation 'OP-1'; peaks 2, 3 and 4 belong to 'OP-2' and so on. The information leakage in this case is maximum and it is highly correlating the power traces which can lead to leakage of secret information.

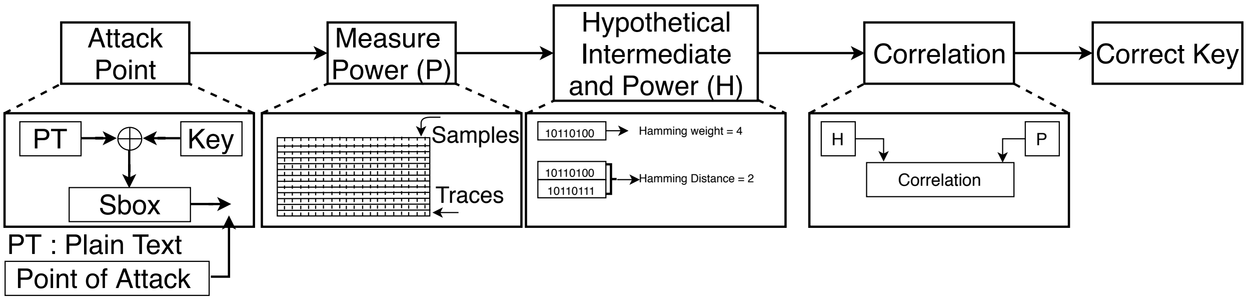

Fig. 2. The process of Correlation Power Analysis (CPA) to extract the correct sequence of key used by the AES

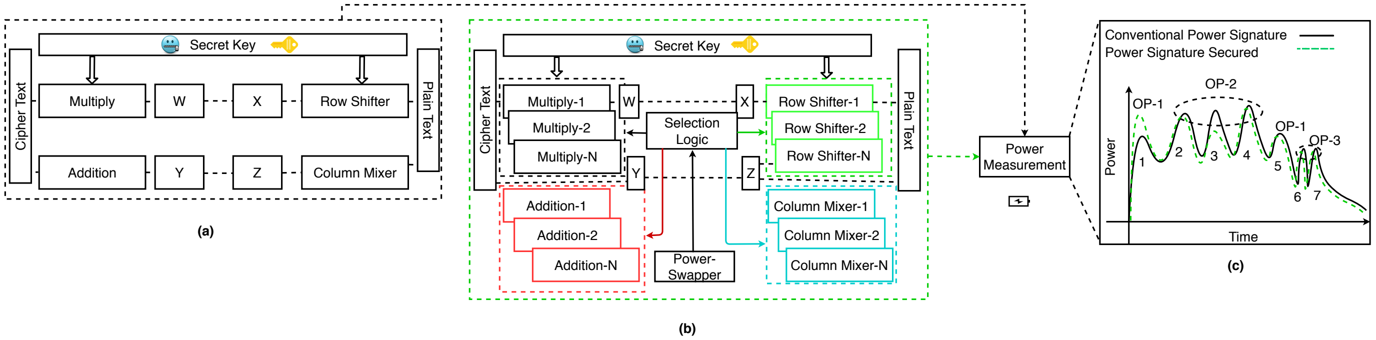

Fig. 3. (a)Conventional cryptosystem where the attacker can deduce secret information from the observed power traces that correspond to the operations;(b) Cryptosystem protected by Power Swapper where the power trace does leak information but it leads to wrong information deduction by the attacker as standby modules aka approximate-modules add misleading information by leaking power traces, similar to other modules in magnitude, that are not known previously to the attacker; (c) Power signatures of the outputs of a conventional unprotected cryptosystem and a system protected by the proposed Power Swapper. *The figure is shown for visualization purposes and should not be used as an accurate depiction of a AES cryptosystem's internal structure*

On the contrary, Figure 3(b) shows the cyrptosystem being secured by Power Swapper where the internally implemented functional modules still perform the same tasks as in a conventional FPGA shown previously except for the fact that there are other approximate-blocks that perform the same function but are designed in such a way that they have different power consumption ratings. These approximate blocks are chosen randomly during runtime by the selection logic which is controlled by the Power Swapper. Since each block still does the same task, there will not be any deviations in the functionality of the application. We use physically unclonable function (PUF) block within the Power Swapper to make the selection process random and unpredictable to the attacker.

Figure 3(c) shows the waveform of the power traces corresponding to the proposed Power Swapper where some of the peaks show different values compared to the conventional FPGA power trace. Peak 1 which previously would give information of operation OP-1 now corresponds to OP-2 as per the attacker based on the power analysis. Peak 3 which belonged to OP-2 now corresponds to the power consumption similar to OP-1. As can be seen, the victim is not altered yet the power traces are completely different and they are not known to and which mislead the attacker. Even if the attacker tries to study a large number of patterns to find the power trace modifications injected by the approximate modules, the efforts would become futile as the trace will keep sweeping between different power magnitudes due to the randomness derived from the PUF block. The power consumption magnitudes of the alternate implementations of basic blocks range from *p1* to *p5* where *p1* < *p2*< *p3*< *p4*< *p5* and the range of the alternate block-1 performing the operations is in the range *p1 to p3*, while that of the block-2 would be in range of *p2 to p4* while yet another block would have it in *p3 to p5* range. As there is overlap in the power consumption range, one approximate block's power corresponds to some other block's range when they both are swapped during runtime. Hence, the attacker would be forced to deduce the sequence of operations as two different ones whereas internally the same row shifting operation was performed with two different power consumption values.

Refer to Figure 4 which shows the power trace of AES implementation without the proposed method. The trace was observed with the following parameters: DUT clock as 1 MHz, sampling frequency of 50 MHz and ADC sampling rate of 50MSps. If we observe the magnitude of the waveform and compare that with the magnitude shown in Figure 5, the change can be vividly seen owing to the approximate modules that perform essentially the same task but with a different power consumption. The way this disrupts the CPA power analysis is: the hypothetical intermediate that will be calculated by the adversary will remain the same (as discussed previously); key and the plaintext remain the same for a particular AES cycle. On the contrary, the actual power consumption would come out to be different and hence, if not for all the parts of the key but for some, parts of the key the adversary derives is incorrect disrupting the CPA analysis. Figure 5 illustrates an increase in the magnitude but it can also be the opposite depending how the approximate modules are designed. Hence, theoretically, if one harnesses the approximate modules for enhancing security in FPGA based crypto implementations, power analysis based side channel attacks could be thwarted. **Ovehead Analysis**: As with every system, our proposed methodology will also have overheads. The Power Swapper requires that approximate modules be added to the original implementation of AES and these modules will be selected during the application execution. Needless to say, the modules will require additional space on the FPGA fabric along with some increase in power consumption. Switching between these modules will also lead to small overheads. The small, if not insignificant, overhead would be the trade off between security and power/area. We plan to present a detailed analysis of overhead comparison with other related works in future.

IV. Experimental Results

The experimental setup used for capturing AES traces (without Power Swapper) was: 1. Artix-7 based FPGA controller, CW305 Artix FPGA Target DUT board, PicoScope 5000 series for capturing the power traces, system clock frequency used was 1 MHz. The cyrpto application we implemented on the DUT board was AES [22] with 128 bits of plaintext, key and output cipher text. Pearson correlation was used to calculate hypothetical power. Automation scripts were used to provide test vectors to the DUT and 1 million traces were collected for analysis.

Refer to Table I for the power traces observed by the attacker with Power Swapper. As can be seen from table, the information deduced by the attacker based on power consumption values are completely different compared to the actual operation executed on the core. The modified power signatures observed by the attacker are highlighted in red. The modified signatures are a result of the Power Swapper choosing one of the approximate blocks. Similarly, referring to Table II, the key derived using CPA without and with Power Swapper has been shown. The bytes/nibble that were wrongly correlated to the key guesses - as described previously - have been highlighted. These wrong portions of the keys are observed as the effect of the approximate modules introduced by Power Swapper . It is to be noted that the length of the plaintext and key does not in anyway affect the efficacy of the proposed method. The results in the Tables I, II provide sufficient proof that by employing approximate modules, cryptosystems can be rendered resilient against power analysis based side-channel attacks.

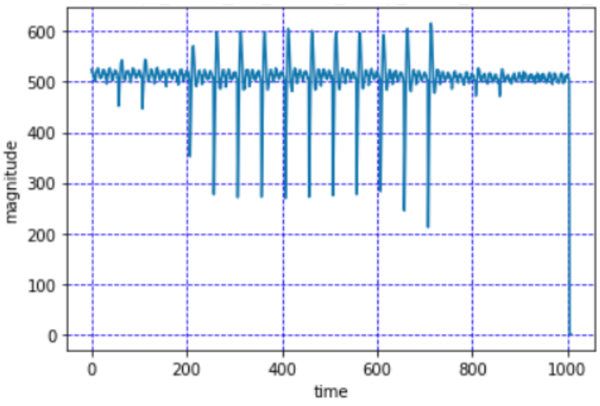

Fig. 4. Power trace of AES implementation on FPGA

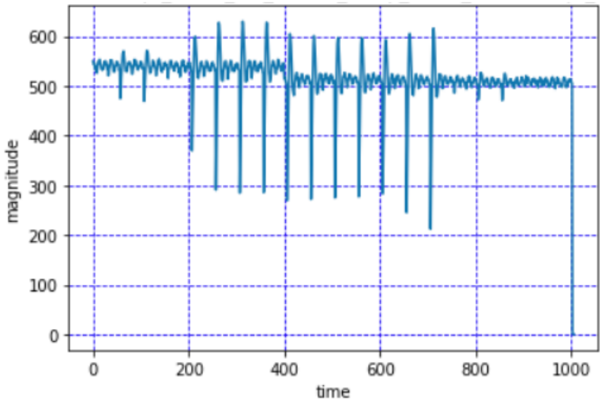

Fig. 5. Power trace of AES controlled by Power Swapper . The magnitude is seen to increase(in this case) given the presence of approximate functional blocks

TABLE I
Impact of Power Swapper on power trace extraction

Scenario	Victim Power Trace	Power Trace with Power Swapper	
		Instance-1	Instance-2
Scenario-1	OP-1/OP-2/OP-3/OP-4	OP-3/OP-2/OP-1/OP-4	OP-2/OP-3/OP-4/OP-1
Scenario-2	OP-1/OP-1/OP-4/OP-3	OP-2/OP-3/OP-2/OP-1	OP-3/OP-1/OP-1/OP-3

TABLE II
Impact of Power Swapper on keys

Trace #	Correct key	Incorrect key with Power Swapper
Trace-1	51720187c36e0c8523acb8535a870703	51522187ca6ea28523acb8e35a870793
Trace-2	d14a900c7391d64101fe33a85b0793cb	a14a90dc7391d63201fe33a85b1693cb

V. Conclusion and Future Work

In this work, we discussed the physical power SCAs, discussed the severity of the threats posed and delineated the works in the past. In contrast to the existing works, proposed Power Swapper will preserve the victim's secret information without any modifications to the victim algorithm in itself. Our proposed mechanism is capable to protect any cryptosytem for that matter but we have kept the details concise for this paper. We hope the community will be intrigued by the preliminary results discussed in this work and we plan to develop this work in future to deliver more details of our mechanism that would benefit the security critical processes.

ACKNOWLEDGEMENT

We thank Dr. Jens-Peter Kaps and Mr. Abubakr Abdulgadir from Cryptographic Engineering Research Group - George Mason University for providing access to the FOBOS [22] setup for power-side channel analysis, and offering all the kind help in configuring FOBOS for the experiements.

REFERENCES

[1] S. Shukla, G. Kolhe, S. M. P D, and S. Rafatirad, "Stealthy malware detection using rnn-based automated localized feature extraction and classifier," in *International Conference on Tools with Artificial Intelligence (ICTAI)*, 2019.

[2] S. Shukla, G. Kolhe, S. M. PD, and S. Rafatirad, "Rnn-based classifier to detect stealthy malware using localized features and complex symbolic sequence," in *International Conference On Machine Learning And Applications (ICMLA)*, 2019.

[3] S. Shukla, G. Kolhe, S. M. P. Dinakarrao, and S. Rafatirad, "On-device Malware Detection using Performance-aware and Robust Collaborative Learning," *Design Automation Conference (DAC)*, 2021.

[4] A. Dhavlle, S. Shukla, S. Rafatirad, H. Homayoun, and S. M. Pudukotai Dinakarrao, "Hmd-hardener: Adversarially robust and efficient hardware-assisted runtime malware detection," in *Design, Automation Test in Europe Conference Exhibition (DATE)*, 2021, pp. 1769–1774.

[5] G. Kolhe and et.al., "Security and complexity analysis of lut-based obfuscation: From blueprint to reality," in *Int. Conference On Computer Aided Design*, 2019.

[6] Z. Chen, G. Kolhe, and et.al, "Estimating the circuit deobfuscating runtime based on graph deep learning," in *Design, Automation and Test in Europe Conference (DATE)*, 2020.

[7] S. M. P. Dinakarrao, S. Amberkar, S. Bhat, A. Dhavlle, H. Sayadi, A. Sasan, H. Homayoun, and S. Rafatirad, "Adversarial attack on microarchitectural events based malware detectors," in *Design Automation Conference*, 2019.

[8] S. Shukla, G. Kolhe, S. M. P. D, and S. Rafatirad, "Microarchitectural events and image processing-based hybrid approach for robust malware detection: Work-in-progress," in *Proceedings of the International Conference on Compliers, Architectures and Synthesis for Embedded Systems Companion*, 2019.

[9] S. Barve, S. Shukla, S. M. P. Dinakarrao, and R. Jha, "Adversarial Attack Mitigation Approaches using RRAM Neuromorphic Architectures," *GLSVLSI*, 2021.

[10] F. Brasser, L. Davi, A. Dhavlle, and et al., "Advances and throwbacks in hardware-assisted security: Special session," in *Conference on Compilers, Architecture and Synthesis for Embedded Systems*, 2018.

[11] A. Dhavlle, S. Bhat, S. Rafatirad, H. Homayoun, and S. M. P. D, "Work-in-progress: Sequence-crafter: Side-channel entropy minimization to thwart timing-based side-channel attacks," in *Conference on Compliers, Architectures and Synthesis for Embedded Systems (CASES)*, 2019.

[12] A. Dhavlle, R. Mehta, S. Rafatirad, H. Homayoun, and S. M. P. D, "Entropy-shield:side-channel entropy maximization for timing-based side-channel attacks," in *21 st International Symposium on Quality Electronic Design (ISQED)*, 2020.

[13] P. Yu and P. Schaumont, "Secure fpga circuits using controlled placement and routing," in *IEEE/ACM International Conference on Hardware/Software Codesign and System Synthesis*, 2007.

[14] Y. Ishai, A. Sahai, and D. Wagner, "Private circuits: Securing hardware against probing attacks," in *Advances in Cryptology*, 2003.

[15] P. C. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology*, ser. CRYPTO '99, 1999.

[16] S. Chari, C. S. Jutla, J. R. Rao, and P. Rohatgi, "Towards sound approaches to counteract power-analysis attacks," in *Advances in Cryptology — CRYPTO' 99*, M. Wiener, Ed., 1999.

[17] D. Suzuki, M. Saeki, and T. Ichikawa, "Random switching logic: A new countermeasure against dpa and second-order dpa at the logic level," *IEICE Transactions*, vol. 90-A, pp. 160–168, 01 2007.

[18] E. Trichina, "Combinational logic design for aes subbyte transformation on masked data," IACR report, Tech. Rep., 2003.

[19] Shengqi Yang, W. Wolf, N. Vijaykrishnan, D. N. Serpanos, and Yuan Xie, "Power attack resistant cryptosystem design: a dynamic voltage and frequency switching approach," in *Design, Automation and Test in Europe*, 2005.

[20] A. Dubey, R. Cammarota, and A. Aysu, "Maskednet: The first hardware inference engine aiming power side-channel protection," in *International Symposium on Hardware Oriented Security and Trust (HOST)*, 2020, pp. 197–208.

[21] R. Matovu, A. Serwadda, A. V. Bilbao, and I. Griswold-Steiner, "Defensive charging: Mitigating power side-channel attacks on charging smartphones," in *Conference on Data and Application Security and Privacy*. Association for Computing Machinery, 2020, p. 179–190.

[22] A. Abdulgadir, W. Diehl, and J.-P. Kaps, "An open-source platform for evaluating side-channel countermeasures in hardware implementations of lightweight authenticated ciphers," in *International Conference on Reconfigurable Computing and FPGAs (ReConFig)*, Cancun, Mexico, Dec 2019.

[23] R. Velegalati and J.-P. Kaps, "Towards a Flexible, Opensource BOard for Side-channel analysis (FOBOS)," Cryptographic architectures embedded in reconfigurable devices, CRYPTARCHI 2013, June 2013.