Embedded Randomness and Data Dependencies Design Paradigm: Advantages and Challenges

1st Itamar Levi
Faculty of Engineering
Bar-Ilan University
Ramat-Gan, Israel
itamar.levi@biu.ac.il

2st Yehuda Rudin
Faculty of Engineering
Bar-Ilan University
Ramat-Gan, Israel
yehuda.rudin@biu.ac.il

3nd Alexander Fish
Faculty of Engineering
Bar-Ilan University
Ramat-Gan, Israel
alexander.fish@biu.ac.il

4rd Osnat Keren
Faculty of Engineering
Bar-Ilan University
Ramat-Gan, Israel
osnat.keren@biu.ac.il

***Abstract*—Information leakage through physical channels is a major hurdle in embedded hardware security. This paper overviews the three key factors in the embedded hardware security space, focusing on *gray-box* (*bounded resources*) power analysis attacks: the adversary's knowledge and abilities, the security metrics used by adversaries' and security evaluators and gate-level countermeasures. A new design paradigm, dubbed *pAsynch*, that utilizes *internal signals* and random signals to uniformly spread the information-carrying energy within the clock period in a specific way with a resolution below the band-width and noise-filtering abilities of advanced measurement equipment is introduced. The advantages and design challenges introduced by the *pAsynch* paradigm are discussed.**

***Index Terms*—hardware security, information leakage, pAsynch, pseudo asynchronous, side channel**

I. Background - hardware security

Embedded systems aim to provide security from malicious eavesdropper attackers by implementing cryptographic algorithms that run on dedicated crypto-cores. However, these algorithms fail to prevent secrets leak when they are subjected to side channel analysis attacks (SCA) [1], [2]. SCAs on cryptographic devices exploit unintentional information leak from physical measurable channels (*e.g.*, power supply current, electromagnetic emissions *etc.*). Power Analysis (PA) attacks are the most highly investigated forms of SCAs, and numerous PA attack resilient methodologies have been proposed over the years. PA attacks are attractive because of their low computational effort, low-cost and equipment requirements. The advantage of this type of attack stems from the fact that through their power consumption, circuits leak information related to internal signals within the design.

The amount of sensitive information that can be extracted from a device depends on the methods used to analyze the side-channel information (*evaluation metrics* and statistical tools), the adversary's knowledge about the design (design transparency), its resources, and capabilities. Here, we elaborate on two key factors in hardware security (see Fig. 1): the adversary and the metrics used to quantify security. The third factor, i.e., countermeasures, is discussed in Section II.

This work was supported by the Israel Science Foundation under Grant 1868/16.

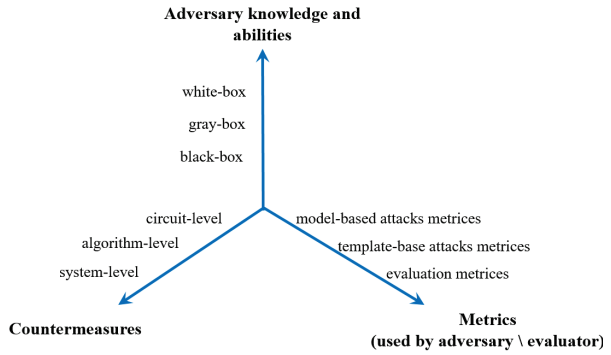

Fig. 1. Adversary's knowledge, evaluation metrics and countermeasures

A. The adversary

Adversaries are typically classified according to their knowledge and abilities. The so-called *blackbox*, *graybox* and *whitebox* are abstractions of an adversary's abilities and knowledge, in ascending order (as illustrated in Fig. 1):

Blackbox: the *blackbox* adversary is assumed to know the functionality of the design and values in its global IOs. This adversary is clearly too weak to constitute a side-channel threat and therefore is not discussed here.

Whitebox: As opposed to the *blackbox* adversary, the *whitebox* adversary is assumed to have full observability of as many real-time intermediate leakages as it wants and to know all the possible design details. Whitebox attacks are considered non-realistic since internal signals cannot be sampled without a substantial information loss due to band-width limitations, accuracy and storage-complexity.

Graybox: The *graybox* adversary has physical access to the device. Such adversaries are traditionally classified as *invasive* or *non-invasive* and *passive* or *active*. A *passive* attacker extracts information from the system without affecting the system or its resources (*e.g.*, energy, radiation, temperature) whereas an *active* attacker affects system operation or alters its resources. *Invasive* or *non-invasive* attacks refer to the interface through which information is extracted. An *invasive* attacker can alter the device by a range of means such as depackaging, de-layering and probing whereas the noninvasive

attacker does not alter the cryptographic device. Below we focus on power supply monitoring (*passive* and *non-invasive*).

A *graybox* adversary typically needs to construct a solid *model* of some internal functionality within the design. To construct an abstract mathematical model, an adversary must have architectural and RTL-netlist (register-transfer-logic) knowledge. There are some limiting factors to these attacks: the attacker can access filtered (low-pass-filter) measurements as a result of the parasitic capacitive and resistive elements on the power-grid/package/board etc., and he also needs to preprocess the measurements to filter system/algorithmic/thermal noise elements. In practice, modeling can be quite challenging (especially in the presence of sophisticated countermeasures designed to make it hard to model the physical side-channel or obscure the implementation details).

B. Metrics used by adversaries and evaluators

Security metrics are often classified into metrics used by adversaries (or defenders) and security metrics used by evaluators (evaluation labs and product providers). In what follows we describe metrics for model- and profile-based attacks aimed to measure the information that can be obtained from the power consumption, and metrics related to security evaluations of the content of information within the current traces regardless whether it can be practically exploited.

1) ***Metrics used by model-based adversaries****:* Model-based attacks uncover the secret key by modeling the logic activity of internal variables of the cryptographic algorithm. Since the secret key is not known, the attacker needs to formulate a hypothesis as to its value and verify this by statistical analysis of the dissipated current. Attacks such as the Correlation Power Analysis (CPA) and Differential Power Analysis (DPA) [3], [4] that use statistical analysis, and Simple Power Analysis (SPA) have long been shown to successfully break computationally-secure cryptographic architectures. The total current measurement contains the current of the signals under attack ($i.e.$, the signal which is modeled) and additional system elements currents (algorithmic noise), thermal noise and noise due to variations in the voltage source, coupling, $etc.$; perfect synchronization can remove the noise up to a certain extent whereas inaccurate synchronization can amplify it. Note that for efficient processing high sampling rates are required for highly protected designs in advanced technologies ($5 - 15$GHz). This implies that the adversary's storage and computation complexity must be proportionate.

In the last ten years, more complex statistical methods have been proposed to efficiently extract information from the current samples from multiple time samples (high dimensionality samples) and multivariate currents distributions, such as High-Order (HO) power analysis attacks and Multivariate (MV) power analysis attacks [5], [6]. In particular, for an attack on Masking or Threshold Implementations (TI) designs (where the secret information resides in higher-statistical moments) to succeed, a *Gray-box* attacker need access to the RTL level description (and in some cases the gate-level description), and an accurate *inter clock-cycle* current model (and in some cases even an *intra clock-cycle* current model). That is, an attacker of order n must hypothesize n points in time when information that relates to a specific computation will leak per computation [7]–[9].

2) ***Metrics used by profile/template-based adversaries****:* Profile-based SCA attacks are highly researched. Profiling (template) based attacks [10]–[12] do not require a model of the physical behavior of the internal signals within the device (current model). It is assumed that an identical device can be initialized with a secret key for leakage characterization. In turn, it implies a very strong adversarial assumptions (in possession of a *"cracked"* device). Most attacks take place in two phases: in the first phase, the side channel information is profiled for many *plaintexts* and *keys* using a sample *"cracked"* device. In the second phase, side channel information is measured from a *sealed* device and curve fitting techniques are utilized combined with various statistical methods to find the best match and extract the secret key. They require very high measurement band-width, storage, processing abilities and commensurate expertise.

In order to reduce the complexity of profiling the (noise) is assumed to be a *white* Gaussian [10]–[12]. It is also assumed that the (high) algorithmic noise can be removed from the *templates* by advanced statistical methods. In fact, *Profiling*-based attacks are very sensitive to "DIE-to-DIE" (global) physical variations and practically, system/algorithmic/thermal noise and noise due to randomization-countermeasures.The latter noise increases the computational and memory complexity). Several *profile-based* attacks that do not require a *"cracked"* device have been discussed in the literature. The most prominent is Correlation Enhanced Collision Attacks [13]. This method identifies and exploits collisions between masked substitution-boxes (SBOXes). It entails correlating the activity of one section of the device to another or alternatively using the activity of one section of the device as a profile/ template to attack another section. The effectiveness of this attack procedure degrades if: (1) the physical signature (current measurement) of one section is different from the other, and (2) when randomization based countermeasures techniques are present.

3) ***Metrics used by evaluators****:* Common *evaluation methods*, e.g., [14]–[16] deal with the issue of how to quantify the information that can be exploited from a side channel and not how to exploit it. Thus, the indications they provide are only as sound as their underlying statistical assumptions. The most highly explored approach is the Test Vector Leakage Assessment (TVLA) method [15]. In general, it uses Welch's t-test statistics to quantify the extent of the differences between the means of two sets of measured data. TVLA tests require far less computational power, measurement time (number of traces) and expertise than powerful template-based attacks. However, the validity of the test results relies on highly simplifying assumptions and on the probability distribution of the measurements; it typically assumes a normal distribution of the side channel measurements and only uses the expected value as a distinguisher. In several reports, the information from

higher statistical moments (and not the mean) were shown to be efficiently used by the test (in cases such as Masking where the secret information lies in the higher moments); however, such tests also rely on statistical assumptions regarding the side-channel distribution, the points of interest (POI) in time where the information lies and their in/dependence. Additional t-test issues such as false-negatives have also been reported [16].

In addition, information-theoretic based metrics also bound the level of information leakage $e.g.$, [17], [18]; however, they require some knowledge or careful characterization of the side-channel probability distribution which might not be correctly obtained and modeled even by extremely powerful adversaries. More concretely, though extremely accurate, these evaluation methods typically assume very powerful adversaries (knowledge, processing, storage, physical and noise-removing capabilities), and also rely on correct modeling of multi-dimensional and multi-variate leakage distributions. In some cases this might not be practical.

II. Countermeasures

SCA attacks have prompted both academia and industry to develop numerous countermeasures. PA attack countermeasures can be classified by their abstraction level into the system level, algorithm/logical level, the circuit/gate level and the device level (as illustrated in Fig. 2). In most cases, some form of multi-level security is embedded into these devices. All countermeasures aim to decrease the correlation between the instantaneous power dissipation and the intermediate processed data within the cryptographic device.

There are two main approaches to coping with information leakage (at all abstraction levels): *hiding* and *masking* [19]. *Masking* refers to manipulations of the algorithm's internal variables by random values (called *masks*) [5], [7], [20]. An advanced form of masking dubbed Threshold-Implementation (TI) [8], [21] also splits the computation variables into sections (shares) and never performs (internal) computations on all the shares jointly (optionally making the leakage distribution multi-dimensional).

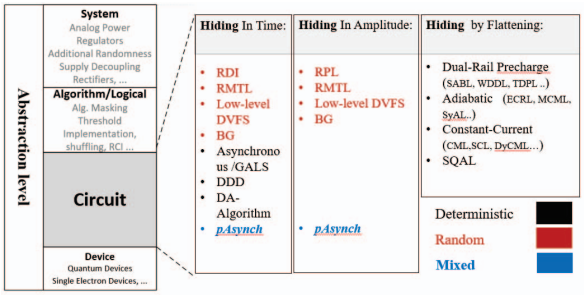

Fig. 2. circuit-level countermeasures classification

Masking and *TI* are typically considered architectural/ algorithmic countermeasures (see however proposals for *gate-level masking* [22], [23]). When multiple (d) *masks* are used to manipulate a value, the masking is dubbed d^{th} order *masking*. Clearly as d increases so does the security; however, it goes hand-in-hand with the area and energy cost of the system. For example, the area overhead of a 2^{nd} order *TI* is x7 compared to an unprotected design as a minimum of 5 input shares and 10 output shares are required, for higher orders it increases rapidly.

By contrast to the algorithm level, circuit level countermeasures require smaller area-overhead. There are many examples of bypassing and neutralizing architectural countermeasures by sophisticated attackers. The main advantage of lower abstraction levels is that they are harder to neutralize. Therefore, circuit-level countermeasures provide inherent security.

Circuit-level *hiding* countermeasures are traditionally categorized into ones that (a) randomize (in time or amplitude domains) the dissipated current, (b) flatten the energy consumption per cycle and (c) embed gate-level *masking* (as illustrated in Fig. 2):

Note that gate-level masking countermeasures [22], [23] are less efficient in terms of area and design-effort than their higher level algorithmic counterparts.

Countermeasures that randomize the power profile include: Random pre-charge Logic (RPL) [24], random clock gating [25], gate level randomization such as RMTL [26] and many more as shown Fig. 2. Some of these randomize the computation in the time domain and others randomize the amplitude of the dissipated current.

The constant energy consumption-based countermeasures (dubbed flattening) can be implemented at the *architectural level* (by utilizing voltage regulators or rectifiers that maintain a constant current regardless of their inputs) or at the circuit level. [27]–[30]. The latter ones are harder to neutralize ($e.g.$, by bypassing the regulators or by sensing the current after the regulator). Numerous countermeasures at the circuit level have been designed to consume constant energy per cycle and thus make it data-independent including SABL [31], WDDL [32], and many more (See Fig. 2). Flattening based countermeasures rely on circuit level symmetry and have been shown to be sensitive to process mismatch, hazards, coupling capacitances, process variations, noise [33]–[35], delay imbalance [36] , $etc.$; all of these non-idealities cause them to leak information [33], [34], [36]–[38].

A sub-class of constant energy consumption countermeasures includes the Adiabatic logic families, e.g., SyAL logic [39], CSSAL logic [40], [41] and others, which work to achieve minimal energy requirements by transferring the charge stored on the logical nodes (capacitors) back to the voltage source. Nevertheless, these circuits are hard to design.

III. The pseudo-asynchronous design style and its underlying advantages and challenges

The solutions above aim to reduce the signal-to-noise ratio and make it harder to sense small differences in energy consumption. However, these countermeasures target constant *inter-cycle* energy consumption but do not consider or aim to achieve constant *intra-cycle* instantaneous power dissipation. This leaves them vulnerable to advanced attacks.

Recall that in both *model-based* and *profile-based* power analysis attacks an attacker tries to locate the set of *best* Points-Of-Interest (POIs). Clearly, as their number, d, increases, the possibility of extracting meaningful information decreases and the computational complexity and memory requirements of the attack increases. These POIs can be located within a single clock cycle (*intra-cycle*) [42]–[44] or across several cycles depending on the circuit/ algorithm implementation [37], [45], [46]. The complexity of finding fixed POIs for masking implementations increases with d. Nevertheless, in cases where the POIs also vary in time, their identification becomes even harder. The *pAsynch* design techniques presented next vary the POIs in time to make such high dimensionality order PAs significantly harder to perform.

A. The pAsynch design style

All statistical power analysis attacks and evaluation metrics require or rely on some synchronization of power measurements and assume that intermediate variable bits are computed simultaneously or instantaneously. The Pseudo-Asynchronous (*pAsynch*) design invalidates these assumptions. This systematic design approach which combines the security advantages of asynchronous circuits with the ease of synchronous design was recently proposed [44]. In [42], [43], we showed that randomization and data-dependencies in *internal signals* were efficient in hiding (in time) the leakage of information during the *Active-region* (dynamic switching currents) and hiding (in amplitude) the information leakage during the *Static-region* (transistors leakage currents). The *pAsynch* design style modulates (locally) the phases of the clock signals per-bit in a module and the local power-supply resistance. The outcome is a (relatively) low-area-overhead design with relatively simple to employ (local) security elements.

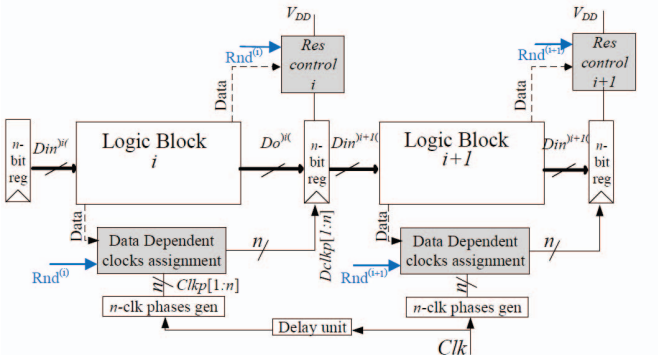

Fig. 3. The (local) Pseudo-Asynchronous, *pAsynch* block architecture

The general *pAsynch* architecture is shown in Fig. 3 where a sensitive n-bit output variable ($D_o(i)$) is sampled by n different clock signals. First, n different phases (*Clkp[1:n]*) of the main clock are generated by the phase-gen module; each is shifted by a constant (predefined) delay Δ. These clock signals (phases) are assigned randomly and/or data-dependently to the n sampling elements (*Dclkp[1:n]*). The assignment is performed at the clock frequency (time dependent). Due to the n different sampling times of the inputs/outputs, bits of the new input vectors enter the block one by one in a mixture of a random and data-dependent order. This sequence of transitions triggers a sequence of transitions at the output; the sequence of output transitions as well as its length depend on the circuit implementation and may change as a function of noise, variations and logic delays. An exemplary current waveforms of the *pAsynch* design (reproduced from [43]) *vs.* time is shown in Fig. 4 for $\Delta = 0, 4, 16$. This mechanism of gradual change of inputs has several advantages [44]:

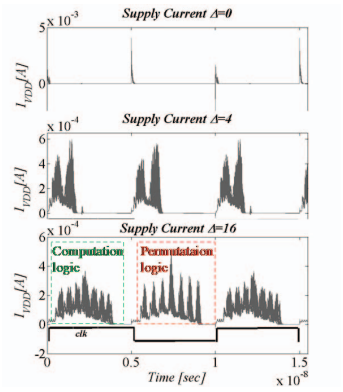

Fig. 4. Power supply current, for $\Delta = 0, 4, 16$, reproduced from [44]

- The number of intra-cycle Points-Of-Interest and intra cycle state hypotheses increases significantly.
- A profiling attacker needs to sufficiently filter the randomization and collect/process high dimensional data-set.
- The instantaneous signal to noise ratio is reduced by a factor of n.
- The delay generator is local and therefore less prone to external tampering and can regulate at a resolution of tens of piko-S.

To efficiently hide the information that leaks from devices in the *Steady state*, internal data dependency and randomness are utilized to manipulate the leakage current amplitudes independently for each sequential elements (See Fig. 3). A control dependent always-on power gate is embedded in the design for each FF to provide a Virtual-VDD (VVDD). It selectively opens a Low VT (LVT) nMOS device or a standard VT (SVT) pMOS device in a key- and data-dependent and random manner.

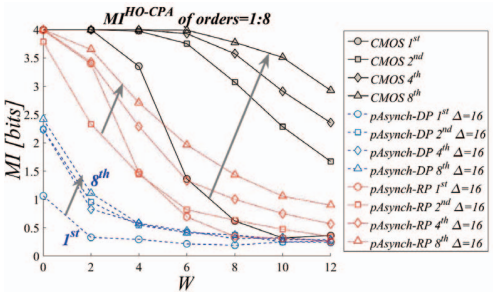

Fig. 5. Mutual Information analysis vs. algorithmic noise , *Active region*

The Mutual Information MI[1] reflects the guessing entropy - the (average) number of key candidates to test after the side channel attack is 2^{n-MI}. The MI is computed between the most correlated key (by a High-Order CPA attack with the appropriate *combining function*, and the correct key). Fig. 5 shows the MI as a function of the number of points of interest (POI) that were used for a one-to-eight High-Order (HO) CPA attack in the presence of architectural noise from W SBOX's. As expected, as W increases, the amount of information that can be learned decreases. Clearly, the unprotected CMOS design is (inherently) vulnerable and so is the truly random based (*R*) flavors of the pAsynch. The Data-Dependent Permutation *(DP)* pAsynch clearly reveals substantially less information on the secret key (even in a noiseless evaluation environment). The most interesting observation is that with very few algorithmic elements the information decreases rapidly.

B. pAsynch design: advantages and challenges

Hardware countermeasures are cardinal to secure designs, the secret ingredient which cannot be jeopardized by *whitebox* adversary is physical noise (and its amplification).

Security evaluation methodologies rely on a good formulation of statistical and the mathematical properties of leakage functions hand-in-hand with a considerable set of physical assumptions regarding the hardware implementation and the adversarys abilities. The integrity of security evaluation during the design stage, through verification to field deployment, depends to a great extent on the physical models accuracy (devices, physical layers, statistical variations and noise modeling), the possibility to simulate realistic settings with simulation tools and emulate peripherals properties (IOs, bonding, package, board, sockets, probes etc.). EDA companies and technology vendors play a major role in providing tools to support the embedding of such evolved hardware design and to enable verifiable-security.

The *pAsynch* design advantages are many-fold: (1) the locality of security elements and their proximity to logic-elements make them hard to neutralize (2) area overhead efficiency (3) it combines the secret-sauce of mixing data dependencies and randomization to make it hard for a modeling-adversary to model the activity and for a template-base adversary (evaluator)to collect, template and process the information and (4) as time manipulation is done in resolution of tens to hundreds of piko-S any acquisition equipment will suffer from very high noise.

With this set of advantages there are challenges we face as security evaluators: mathematical formulation of security properties and concrete bounds of the information leakage for (a scale of) adversaries are still a fairly open problem. Physical assumptions need to be verified as part of the design process and flow.

There are also several challenges that require special design considerations:

[1]to not be confused with the MI discussed in I-B

(1) **Data dependency mapping scheme** - approaches which embed security mechanisms *locally* rely on HDL design partitioning according to size and independence of internal secret variables. Such partitioning should (clearly) be performed by EDA tools as well as the insertion of the security mechanisms. The choosing and allocation of data-dependent internal signals (and random bits) to control the phase permutations or the power-network is also a task which should be automated (according to, *e.g* uniformity, proximity).

(2) **Timing closure considerations** - The modulation of the local clock introduces complexity to the static timing analysis (STA) flow which normally assumes a regular and consistent clock period to calculate setup and hold timing margins. Clock modulation is causing a dynamic (time dependent) change the sampling timing of sequential logic. Several possible approaches can be considered. One is to rely on complex STA flow to restrict some of the clock timing selection options that will result in setup or hold timing violations. A different approach would be to design the clock modulation scheme in a way that will allow a more standard STA flow by guarantying some clock timing attributes that simplify the STA flow.

(3) **Custom cells** - Implementation of *pAsynch* design may require limited usage of special cells which are not included in the standard cell library. Timing characterization, for example, of such cells might not be easily supported by current characterization tools due to complex timing arcs. The local always-on power gating used to modulate the current needs attention during power grid design and physical placement.

In general, security-oriented design will benefit from physical design tools with capabilities to support interactive spatial distance specifications between modules (which are driven by cross talk and flexible design specifications), design-symmetry and balancing requirements. The automation of the design, verification and performance evaluation of embedded hardware security is yet a challenge to be coped by EDA tools.

IV. Conclusions

Hardware countermeasures are cardinal to secure designs, the secret ingredient which cannot be jeopardized by *whitebox* adversary is physical noise (and its amplification). This paper overviews the three key factors in the embedded hardware security space: the adversary knowledge and abilities, the security metrics which are used by adversaries and security evaluators and gate-level countermeasures. A new design paradigm, dubbed *pAsynch* is discussed alongside its' advantages which relate to locality of security elements, combination of data dependencies and randomization and time manipulation which is done in resolution below that of acquisition equipment. The challenges are also discussed in contexts of security evaluation, design and EDA tools.

Acknowledgment

References

[1] P. Kocher, J. Jaffe, B. Jun, and P. Rohatgi, "Introduction to differential power analysis," *Journal of Cryptographic Engineering*, vol. 1, no. 1, pp. 5–27, 2011.

[2] F.-X. Standaert, T. Malkin, and M. Yung, "A unified framework for the analysis of side-channel key recovery attacks." in *Eurocrypt*, vol. 5479. Springer, 2009, pp. 443–461.

[3] P. Kocher, J. Jaffe, B. Jun, and P. Rohatgi, "Introduction to Differential Power Analysis," *Journal of Cryptographic Engineering*, vol. 1, no. 1, pp. 5–27, Mar. 2011.

[4] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2004, pp. 16–29.

[5] F.-X. Standaert, E. Peeters, and J.-J. Quisquater, "On the masking countermeasure and higher-order power analysis attacks," in *Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on*, vol. 1. IEEE, 2005, pp. 562–567.

[6] V. Grosso, F.-X. Standaert, and E. Prouff, "Leakage squeezing, revisited," *CARDIS, Lecture Notes in Computer Science. Springer, Berlin*, 2013.

[7] O. Reparaz, B. Gierlichs, and I. Verbauwhede, "Selecting time samples for multivariate dpa attacks," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2012, pp. 155–174.

[8] A. Moradi, A. Poschmann, S. Ling, C. Paar, and H. Wang, "Pushing the limits: a very compact and a threshold implementation of AES." in *Eurocrypt*, vol. 6632. Springer, 2011, pp. 69–88.

[9] B. Bilgin, B. Gierlichs, S. Nikova, V. Nikov, and V. Rijmen, "Higher-order threshold implementations," in *Lecture Notes in Computer Science*, vol. 8874. Springer-Verlag, 2014, pp. 326–343.

[10] F.-X. Standaert, T. G. Malkin, and M. Yung, *A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 443–461.

[11] S. Chari, J. R. Rao, and P. Rohatgi, "Template Attacks," in *Cryptographic Hardware and Embedded Systems - CHES 2002*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2003, no. 2523, pp. 13–28.

[12] B. Gierlichs, K. Lemke-Rust, and C. Paar, "Templates vs. stochastic methods," in *CHES*, vol. 4249. Springer, 2006, pp. 15–29.

[13] A. Moradi, O. Mischke, and T. Eisenbarth, "Correlation-enhanced power analysis collision attack." in *Ches*, vol. 6225. Springer, 2010, pp. 125–139.

[14] *National Institute of Standards and Technology (NIST). FIPS PUB 140-2: Security Requirements for Cryptographic Modules*, 2001.

[15] J. Cooper, E. Demulder, G. Goodwill, J. Jaffe, G. Kenworthy, and P. Rohatgi, "Test vector leakage assessment (tvla) methodology in practice," *Cryptography Research Inc.*, 2013.

[16] F. Durvaux and F.-X. Standaert, "From improved leakage detection to the detection of points of interests in leakage traces," Cryptology ePrint Archive, Report 2015/536, 2015.

[17] B. Gierlichs, L. Batina, P. Tuyls, and B. Preneel, "Mutual information analysis," *Cryptographic Hardware and Embedded Systems–CHES 2008*, pp. 426–442, 2008.

[18] S. Mangard, E. Oswald, and F.-X. Standaert, "One for all–all for one: unifying standard differential power analysis attacks," *IET Information Security*, vol. 5, no. 2, pp. 100–110, 2011.

[19] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. Springer Science & Business Media, Jan. 2008.

[20] T. S. Messerges, "Using second-order power analysis to attack dpa resistant software," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2000, pp. 238–251.

[21] B. Bilgin, B. Gierlichs, S. Nikova, V. Nikov, and V. Rijmen, "A more efficient AES threshold implementation," in *International Conference on Cryptology in Africa*. Springer International Publishing, 2014, pp. 267–284.

[22] A. J. Leiserson, M. E. Marson, and M. A. Wachs, "Gate-level masking," Feb. 14 2017, uS Patent 9,569,616.

[23] W. Fischer and B. M. Gammel, "Masking at gate level in the presence of glitches," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2005, pp. 187–200.

[24] M. Bucci, R. Luzzi, M. Guglielmo, and A. Trifiletti, "A Countermeasure against Differential Power Analysis Based on Random Delay Insertion," in *IEEE International Symposium on Circuits and Systems, 2005. ISCAS 2005*, May 2005, pp. 3547–3550 Vol. 4.

[25] H. Qu, J. Xu, and Y. Yan, "A Random Delay Design of Processor against Power Analysis Attacks," in *2010 10th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)*, Nov. 2010, pp. 254–256.

[26] M. Avital, H. Dagan, O. Keren, and A. Fish, "Randomized multitopology logic against differential power analysis," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 4, pp. 702–711, 2015.

[27] H. Hernandez, J. Scott, and W. V. Noije, "Dpa insensitive voltage regulator for contact smart cards," in *2012 25th Symposium on Integrated Circuits and Systems Design (SBCCI)*, Aug 2012, pp. 1–4.

[28] V. Telandro, E. Kussener, H. Barthélemy, and A. Malherbe, "A bi-channel voltage regulator protecting smart cards against power analysis attacks," *Analog Integrated Circuits and Signal Processing*, vol. 59, no. 3, pp. 275–285, 2009.

[29] M. Kar, A. Singh, S. Mathew, A. Rajan, V. De, and S. Mukhopadhyay, "8.1 Improved power-side-channel-attack resistance of an AES-128 core via a security-aware integrated buck voltage regulator," in *Solid-State Circuits Conference (ISSCC), 2017 IEEE International*. IEEE, 2017, pp. 142–143.

[30] J. Liu, Y. Yu, F.-X. Standaert, Z. Guo, D. Gu, W. Sun, Y. Ge, and X. Xie, "Small tweaks do not help: differential power analysis of milenage implementations in 3g/4g usim cards," in *European Symposium on Research in Computer Security*. Springer, 2015, pp. 468–480.

[31] K. Tiri and I. Verbauwhede, "A logic level design methodology for a secure dpa resistant asic or fpga implementation," in *Proceedings of the conference on Design, automation and test in Europe-Volume 1*. IEEE Computer Society, 2004, p. 10246.

[32] ——, "Secure logic synthesis," *Field Programmable Logic and Application*, pp. 1052–1056, 2004.

[33] L. Sauvage, S. Guilley, J.-L. Danger, Y. Mathieu, and M. Nassar, "Successful attack on an fpga-based wddl des cryptoprocessor without place and route constraints," in *Proceedings of the Conference on Design, Automation and Test in Europe*. European Design and Automation Association, 2009, pp. 640–645.

[34] S. Guilley, S. Chaudhuri, L. Sauvage, T. Graba, J.-L. Danger, P. Hoogvorst, V.-N. Vong, M. Nassar, and F. Flament, "Shall we trust wddl?" in *Future of Trust in Computing*. Springer, 2009, pp. 208–215.

[35] S. Mangard, N. Pramstaller, and E. Oswald, "Successfully attacking masked aes hardware implementations," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2005, pp. 157–171.

[36] D. Suzuki and M. Saeki, "Security evaluation of dpa countermeasures using dual-rail pre-charge logic style," in *CHES*, vol. 4249. Springer, 2006, pp. 255–269.

[37] C. Clavier, J.-S. Coron, and N. Dabbous, "Differential power analysis in the presence of hardware countermeasures," in *Cryptographic Hardware and Embedded SystemsCHES 2000*. Springer, 2000, pp. 13–48.

[38] S. Mangard, "Hardware countermeasures against dpa-a statistical analysis of their effectiveness," in *ct-rsa*, vol. 2964. Springer, 2004, pp. 222–235.

[39] B.-D. Choi, K. E. Kim, K.-S. Chung, and D. K. Kim, "Symmetric Adiabatic Logic Circuits against Differential Power Analysis," *ETRI Journal*, vol. 32, no. 1, pp. 166–168, Feb. 2010.

[40] C. Monteiro, Y. Takahashi, and T. Sekine, "Low Power Secure AES S-box using Adiabatic Logic Circuit," in *2013 IEEE Faible Tension Faible Consommation (FTFC)*, Jun. 2013, pp. 1–4.

[41] M. Avital, H. Dagan, I. Levi, O. Keren, and A. Fish, "DPA-Secured Quasi-Adiabatic Logic (SQAL) for Low-Power Passive RFID Tags Employing S-Boxes," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no. 1, pp. 149–156, Jan. 2015.

[42] I. Levi, O. Keren, and A. Fish, "Data-dependent delays as a barrier against power attacks," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no. 8, pp. 2069–2078, 2015.

[43] I. Levi, A. Fish, and O. Keren, "Cpa secured data-dependent delay-assignment methodology," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2016.

[44] ——, "Low-cost pseudoasynchronous circuit design style with reduced exploitable side information," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2017.

[45] M. Tunstall and O. Benoit, "Efficient use of random delays in embedded software," *wistp*, vol. 4462, pp. 27–38, 2007.

[46] N. Veyrat-Charvillon, M. Medwed, S. Kerckhof, and F.-X. Standaert, "Shuffling against side-channel attacks: A comprehensive study with cautionary note," *Advances in Cryptology–ASIACRYPT 2012*, pp. 740–757, 2012.