RAD-FS: Remote Timing and Power SCA Security in DVFS-Augmented *Ultra-Low-Power* Embedded Systems

DANIEL DOBKIN[**], Bar-Ilan University, Faculty of Engineering, Israel
NIMROD CEVER[*], Bar-Ilan University, Faculty of Engineering, Israel
ITAMAR LEVI[*], Bar-Ilan University, Faculty of Engineering, Israel

High-performance crypto-engines have become crucial components in modern System-On-Chip (SoC) architectures across platforms, from servers to edge-IoTs'. Alas, their secure operation faces a significant obstacle caused by information-leakage accessed through Side-Channel Analysis (SCA). Adversaries exploit statistical-analysis techniques on measured (e.g.,) power and timing signatures generated during (e.g.,) encryption, extracting secrets. Mathematical countermeasures against such attacks often impose substantial power-performance-area overheads. Dynamic Voltage and Frequency Scaling (DVFS) techniques provide power-efficiency by varying power consumption according to workload; these modulations are called power-states. Unintentionally, DVFS introduces new inherent weaknesses exploitable by malicious actors: power-states *leak* information in both power and timing side-channels, measurable in software and hardware. We introduce a method to increase side-channel resistance using integrated voltage regulators and DVFS: (1) Pushing known prior-art in the topic to Ultra Low Power (ULP) regime (2) For the first time introducing a mechanism to aid in counteracting the inherent weakness of DVFS in SCA (3) Providing measurements performed on 40nm process ULP PLS15 test-chip down at 580 mV power-supply (4) Offering improved and parameterized resistance to *remote*-timing vulnerabilities inherent to DVFS. We present various results and perform a detailed analysis while comparing performance and security to prior-art. Importantly, our solution is configurable in terms of security, maintaining degrees-of-freedom for power-optimization of DVFS.

CCS Concepts: • **Security and privacy** → **Hardware security implementation**; **Cryptanalysis and other attacks**; Embedded systems security; **Side-channel analysis and countermeasures**; **Cryptography**; **Web application security**; *Hardware-based security protocols*; • **Hardware** → *Hardware test*; *System on a chip*; Signal integrity and noise analysis.

ACM Reference Format:
Daniel Dobkin, Nimrod Cever, and Itamar Levi. 2024. RAD-FS: Remote Timing and Power SCA Security in DVFS-Augmented *Ultra-Low-Power* Embedded Systems. *ACM Trans. Embedd. Comput. Syst.* 1, 1, Article 1 (June 2024), 27 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Side-Channel Analysis (SCA) attacks [9, 26, 45] are a category of hardware security attacks, which extract or retrieve information from a system by utilizing nonstandard channels as compared to conventional communication interfaces. These nonstandard channels are *leaking* sensitive information manipulated by the digital electronic system, namely the physical implementation of theoretical cryptographic systems.

[*] All authors contributed equally to this research.

Authors' Contact Information: Daniel Dobkin, Bar-Ilan University, Faculty of Engineering, Ramat-Gan, Israel, daniel.dobkin@live.biu.ac.il; Nimrod Cever, Bar-Ilan University, Faculty of Engineering, Ramat-Gan, Israel, Nimrod.cever@biu.ac.il; Itamar Levi, Bar-Ilan University, Faculty of Engineering, Ramat-Gan, Israel, itamar.levi@biu.ac.il.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM 1558-3465/2024/6-ART1
https://doi.org/XXXXXXX.XXXXXXX

Dynamic Voltage and Frequency Scaling (DVFS) [5, 30, 41, 54] is a widespread technique utilized to save power on a wide range of computing systems, from tiny Internet of Things (IoT) devices to embedded systems, laptops/desktops systems, high-performance server systems, etc., by reducing the frequency at which they operate.

Vast literature and research exists on the (very successful) use of DVFS to improve the energy efficiency of computation systems. This is done by adapting the voltage/frequency to the workloads of the system jointly constrained with the general timing limitations. Various modern microprocessors (if not all) and embedded systems are equipped with the DVFS functionality such as chips from NVIDIA, AMD, Intel [1, 36, 46]. Some devices offer hundreds of voltage and frequency levels with high resolution, which can be controlled through software like MSI Afterburner [51]. Different parts of these systems adapt independent subsystems, each with its own *optimized* DVFS strategy, as seen in architectures like ARM big.LITTLE [25, 40]. SCA attacks security is a critical requirement. Typical mechanisms to counteract SCA's in a mathematically rigorous way are very expensive. For example, by coding an internal value or variables in the computation to a redundant representation which is randomized and invertible, denoted by masking [15, 43].

On the less rigorous approach, more heuristic solutions exist utilizing other randomization mechanisms such as randomizing the time [48] or amplitude [31] signal domains. For time variations, various shuffling mechanisms were recently supported with theoretical models indicating quasi-linear security levels [32]. For amplitude randomization mechanisms, a fine-grain leakage model supported by silicon measurement was recently provided showing also quasi-linear security levels [8]. The electronic cost of both techniques is far lower than that of masking, but each approach has its own limitations such as algorithmic dependence or the need to embed specialized components, and more. Several other countermeasures that require randomizing voltage and frequency are described in [21, 29, 44], but they necessitate significant redesign, including, for example, all-digital clock modulation using a global modulator. In contrast, the proposed approach is inherently embedded in such platforms and is software (SW) controlled, pushing the limits toward Ultra Low Voltage (ULV) and Ultra Low Power (ULP) embedded systems.

DVFS introduces inherent weaknesses exploitable by malicious actors. The power-states (P-state) are easy to classify and categorize, as shown by a new class of software enabled attacks [49]. The hamming weight (HW) of calculated data causes variations in P-state that leak significantly. We offer a solution that coexists with power optimization (Section 4) while increasing side channel resistance in both power and timing (illustrated in Fig. 15c and 22) by orders of magnitude in software and hardware. Our approach is optimized towards ULP embedded devices, sacrificing performance for null cost in terms of power, area and implementation effort (Tab. 4). Randomly assigning P-states from a group of frequencies instead of the single frequency approach as described in Fig. 3, **will drastically reduce requirements from algorithmic hiding schemes [31] and enable additional degrees of freedom to system designers**. It is important to note that the utilization of the proposed mechanism is orthogonal to any other multi-layer approach which in addition embeds more security measures, such as by low-order masking in software or any other software / hardware or architectural countermeasure. Additionally, the cost of the proposed mechanism is tiny on all electronic cost terms; meaning, it already provides several nice features for such ULP devices which cannot bear higher countermeasures cost and need remote timing and power SCA security.

1.1 Contribution

In this research we push forward three observations: (1) nowadays embedded systems embed efficient DVFS mechanisms inherently which can be utilized to integrate security features without special design efforts and hardware intervention (2) such mechanisms can be very efficient, as supported by rigorous evaluation to counteract SCA, (3) native DVFS mechanisms inherently induce other SCA channels (timing and P-state monitoring) which become data/workload-dependent, our proposed mechanisms can potentially aid in mitigation of these channels. We provide for the first time significant advance on the analysis of all these aspects and we showcase several State of The Art (SOTA) use cases on a very advanced platform, significantly extending the body of knowledge in such ULP regime. **(1):** Working with a SOTA ULP embedded device, where scarce platforms exist in the market going down to 500 mV in operational chips **(2):** Providing security analysis over such devices for the first time to the best of our knowledge. **(3):** Proposing unique methodology and embedded mechanisms to provide SCA security utilizing the inherent DVFS features with ultra low-cost as compared to other solutions regarding area, power, implementation effort overheads; denoted Randomized Aliasing Dynamic Frequency Scaling (RAD-FS). **(4):** Reporting analysis both for a RISC-V processor core and an NXP encryption accelerator embedded on the same device in 40 nm technology, in a comparative view. **(5):** For the first time we demonstrate a countermeasure against a new class of timing attacks such as [49], which coexists with DVFS mechanisms (opposed to the naive solution of turning DVFS off), in addition to the inherent power SCA attacks immunity. We show RAD-FS is very relevant for such network remote timing attacks mitigation. Results are demonstrated via. an ideal (optimal) oracle modeling the RAD-FS parameters. Our constructed oracle is very *generous* with how much control is given to the adversary.

The outline of this paper is as follows. In Section 2, we present a background and literature review. In Section 3 we introduce the remote timing vulnerability inherent in DVFS. In Section 4, we present our novel solution with a detailed explanation and comparison to the current industry approach. Section 5 details the evaluated device and the measurement setup. Section 6 details the security metrics, an in-depth analysis of the results and an analytical approach to the performance cost. Section 7 puts RAD-FS in comparison to known methods. Section 8 discusses comparisons and challenges that adversaries face in real-life scenarios. Section 9 concludes the results of this work and proposes future prospects.

2 BACKGROUND

2.1 Side Channel Analysis

SCA attacks are powerful, repeatedly showing their efficiency in extracting sensitive information from a cryptographic system by analyzing unintentional side channels.

For example, the timing of signals [49] or power consumption [35, 38], and electromagnetic (EM) radiation [2, 42] of a device can all be used in an attack campaign. Generally, Differential Power Analysis (DPA) attacks are classified as model-based or profiling-based: model-based attack such as correlation power analysis (CPA) [10] utilizes Pearson's correlation coefficient ρ and a leakage model to compare side channel leakages to the model. Template attacks [17] utilize a template or a statistical model derived from the leakage of a specific internal value used to enhance the attack's effectiveness in an attack campaign by comparing it to the actual leakage. The costs of SCA countermeasures can be sublinear, linear [31, 32] or exponential [15, 43] and are very hard to embed with a strict energy budget. This is witnessed by the requirements of the NIST lightweight authenticated encryption contest [47]; especially for low-power constrained or battery operated IoT devices that can not sacrifice energy budget as needed for rigorous security level utilizing e.g., masking.

2.2 Dynamic Voltage Frequency Scaling

The concept of DVFS is based on the understanding that the energy consumption of a component is directly proportional to its supply voltage, while the computation delay is inversely proportional to the square of its operating frequency (depicted in Fig. 3(a)), stemming from strong-inversion/near-threshold current equations of transistors. By controlling the supply voltage or clock frequency, it is possible to achieve significant reductions in energy consumption. Frequency randomization is a technique used to introduce randomness or unpredictability in the occurrence or timing of events. It is often employed in various domains, including communication systems, network security, and data privacy, to mitigate potential attacks or reduce vulnerabilities. By randomizing the frequency of events, it becomes harder for adversaries to analyze patterns or launch coordinated attacks.

Double-edged sword: While DVFS has obvious merits, as hinted above, it introduces security flaws by its design. P-states, manifested by different data manipulations, affect the electrical characteristics of an operation (voltage, latency etc.), leading to various sensitivities. As opposed to conventional SCA attacks that measure power or EM emissions and require a *close-contact* adversary, DVFS also allows for SW based attacks by various mechanisms: For instance, Hertzbleed [49] represents a novel category of side-channel attacks that exploit network timing and latency profiles in the spectrum. These vulnerabilities can be exploited regardless of the physical distance. Hertzbleed has demonstrated extraction of cryptographic keys from remote servers which embed modern x86 CPU's in a scenario that was previously believed to be secure; *solely exploiting the inherent sensitivity DVFS offers.* Another example is [39] in which the voltage dependence offered by DVFS on an iPhone 13 affects the overall power of the device, and a video footage of a device's power LED, radiating to a long distance can be used for SCA. Alternatively, the authors have shown that if the device is powered by a USB hub, the hub's current draw can provide an attack entry. Noteworthy, in these reports the attacker makes use of an ultra low-resolution side-channel (as compared to high resolution e.g., power SCA):the device's embedded power sensor, or a low resolution web camera. As an example, its resolution is clearly far from being as high as of a conventional 10 to 14 bit quantizer of an oscilloscope. In addition, it is typically a very slow sensor hence significant averaging and noise is incorporated in this physically measurable quantity. *Therefore, as will be discussed below it will be quite easy for our randomization mechanism to make these attacks hard.*

2.3 Ultra Low Power Regime and Design

In processors implemented with standard process technologies, the operating voltage span ranges from V_{dd} overdrive of several hundred mV above the nominal voltage and down to strong inversion/high near-threshold region of transistors, only several hundreds of mV under the nominal voltage. Implementing electronic systems that operate over a much larger voltage span, from over V_{dd} to sub-threshold voltages requires special (and not always standard) design techniques such as embedding isolation cells, level shifters, and special cells libraries (high-V_{Th}) etc. The main challenge is that such devices are typically not fully characterized by the foundry across the entire voltage span, making it difficult to design robustly and to perform a fully digital design flow with guarantees. Nevertheless, this possibility raises the question of DVFS efficiency across such extended ranges. Existing theoretical research has proven that further energy efficiency is possible with extended voltage range, even ranging below $0.5 \cdot V_{dd}$.However, extending it to the full sub-threshold region is beneficial only for specific application classes. Therefore, in practice, with the emergence of ULP devices which sacrifice performance for longer battery life, we see unique ULP embedded platforms that are designed to work in 500-600 mV (generally mid to high near-threshold operation). Such systems heavily utilize DVFS techniques. *Designing such systems is not an easy task and require high expertise, but available devices exist on the market which we believe will provide game-breaking abilities for (e.g.,) IoTs.* One such unique platform is PLSense's PLS15 platform, used in this research.

Overall, the development of fully functional ULV standard cells has become crucial to meet the growing demand for energy-efficient, low-power electronic systems.

2.4 Adversary and Threat Model

The adversary is modeled by a Probabilistic Polynomial Time (PPT) algorithm. Utilizing different assumptions and capabilities per scenario.

2.4.1 *Power Analysis.*

For subsections 6.2 to 6.5 we assume the adversary has eavesdropper capabilities as well as SCA trace access (such as power leakages) as illustrated in Fig. 1. In our scenario the adversary has physical access to a device, with knowledge of the whereabouts of some power supply pin. The adversary requires no

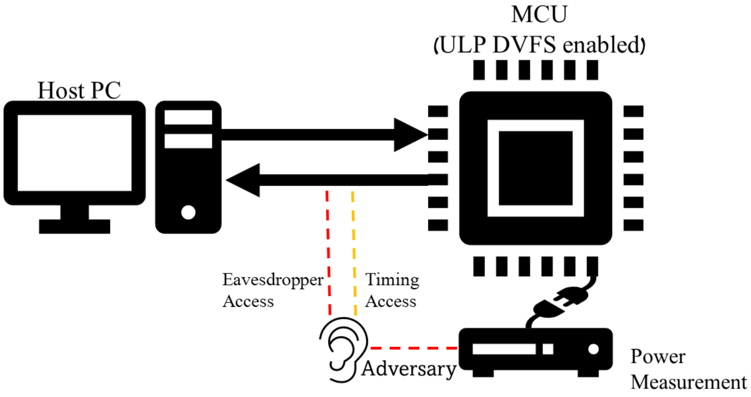


Fig. 1. Attack and adversarial model where different eavesdropper and measurement capabilities are assumed depending on the scenario.

assumptions about the internals of the device or architectural knowledge, nor do we conceal any information regarding the DVFS mechanism/algorithm.

2.4.2 Time-Domain Attacks. For subsection 6.6 we assume the adversary has access only to the communication channel and measures response time. The adversary assumes a DVFS module that introduces some dependency between average HW values in the plaintext to P-state (e.g., for concrete attacks utilizing such a scenario [49]), which becomes apparent in time domain measurements, evaluated as time between a request from the host and the reply from the DVFS enabled Micro-Controller Unit (MCU). The goal set for the adversary is extracting secret information (values) from the time domain distribution. Notably, timing information is trivial to extract when countermeasures are not embedded [11, 12].

3 REMOTE TIMING ATTACKS

Remote timing attacks are devastatingly effective. Such attacks can extract secret information from a target device *remotely* [49], on both symmetric and asymmetric [50] encryption schemes. It is known that the execution time for various algorithms utilizing (e.g.,) modular exponentiation or multiplication [7, 14], depends linearly on the number of '1' bits in the key or state [37], demanding for *constant-time* codes. However, even when some countermeasures are taken (e.g., non-conditional branching, pipelining) they do not ensure that logical '1' and '0' are the same power wise, which may still be an issue with DVFS enabled systems and the associated remote timing attack as discussed here.

Fig. 2 shows how data HW translates into **p-states**, that affect the total time required for 100 encryption requests from a simulated host. The variation comes in either the time required for asynchronous communication, or the algorithmic run time, and is dependent on the encryption scheme. This translates into leakages with Gaussian proximate distributions, that can be measured both on-device with cycle counting, and off-device by measuring response time, a clear vulnerability that could be utilized by adversaries.

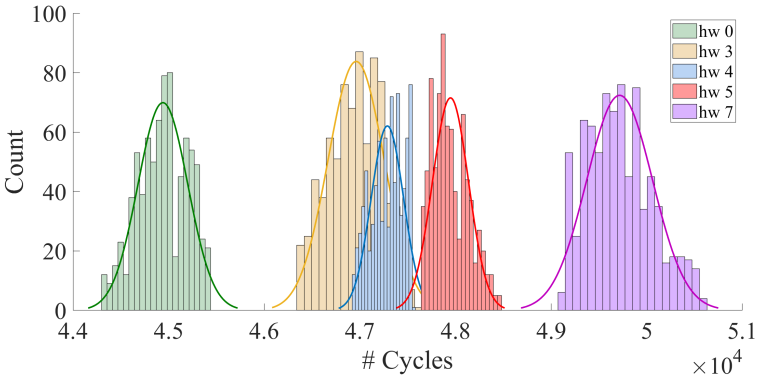


Fig. 2. 100 encryptions are performed per request on a simulated server, data HW triggers the DVFS mechanism, showing a clearly distinguishable timing signature.

3.1 Hertzbleed: DVFS is Inherently *Insecure*

Hertzbleed stands out from other remote timing attacks due to its unique exploitation of DVFS in modern processors. While other timing attacks typically focus on cache access times [28, 52], architecture structure [13] or cryptographic operation timing [4, 6, 27], Hertzbleed leverages CPU frequency adjustments based on workload intensity. Hertzbleed looks at both the overall time a certain operation takes and at the *variations in CPU frequency* during these operations, increasing its flexibility. Unlike attacks like Prime+Probe [52] or Spectre [13], Hertzbleed does not require the attacker to run any code on the victim's machine. Instead, the attack relies entirely on timing data observed from normal network based interactions, making it more accessible for remote exploitation.

Feature	**Hertzbleed**	**Other Remote Timing Attacks**
Attack Mechanism	Exploits DVFS	Exploit memory access, cache, or speculative execution timing
Remote Feasibility	Fully remote without specialized access	Many require local code execution or shared hardware resources
CPU Dependency	Targets universal CPU feature (DVFS) in Intel, AMD and RISC-V devices	Often rely on specific CPU architecture features (e.g., caches, speculative execution)
Granularity of Timing Info	Measures CPU frequency changes during various (e.g,) cryptographic operations	Often measures specific memory/cache access times or instruction execution delays
Target	Cryptographic operations (e.g., SIKE, RSA, ECDSA, ECDH)	Cryptography, memory access patterns, speculative execution
Mitigation Difficulty	*Difficult to mitigate without disabling DVFS (which affects performance)*	Mitigations typically involve firmware patches or software fixes (e.g., flushing caches, speculative barriers, const. time)

Table 1. Comparison of Hertzbleed to various timing attacks.

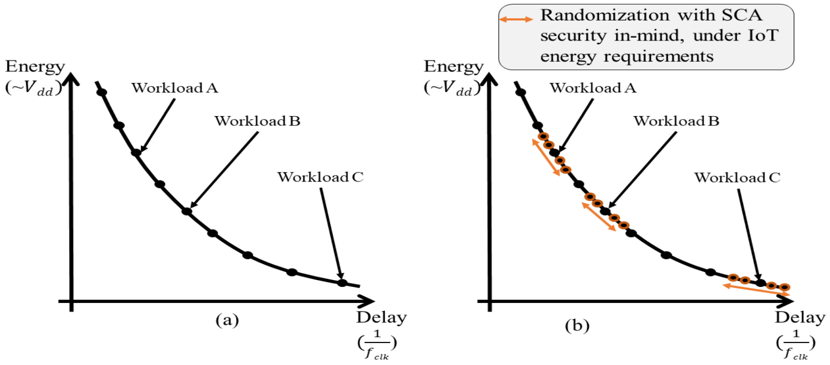


Fig. 3. DVFS conceptual implementation (a) insecure, single frequency per workload vs. (b) our proposed secure implementation, with multiple frequencies assigned per workload.

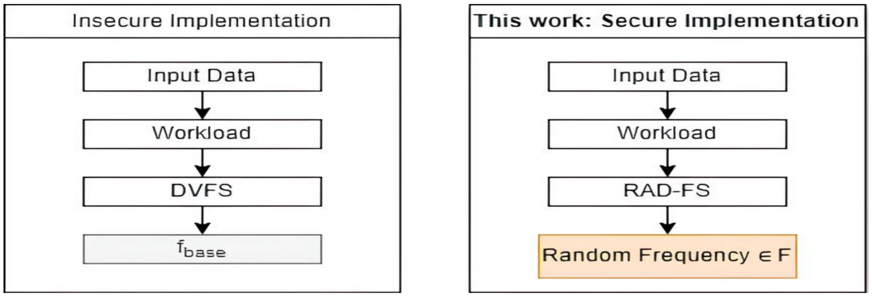


Fig. 4. Flowchart describing an insecure implementation vs our secure implementation for DVFS operation

4 THE PROPOSED APPROACH

4.1 Notations and Symbols

Below, brief summary of notation and symbols used in this study for clear readability. Sets are shown in uppercase italic, elements are shown in lowercase italic, $||\cdot||$ denotes size or cardinality and subscript notes enumeration.

(1) F' - the generally available set of operating frequencies.
(2) f_i - some frequency from the generally available set of frequencies i.e $f_i \in F'|i \in \mathbb{N}$
(3) F - a chosen subset of frequencies such that $F \subset F'$
(4) f_n, f_m - different frequencies such that $f_n \neq f_m \in F|n \neq m \in \mathbb{N}$
(5) f_{base} - base operating frequency set by the DVFS for some workload.
(6) $||F||$ - cardinality of the subset F.
(7) f_{max}, f_{min} - the fastest and slowest frequencies in some subset.
(8) BW - Bandwidth of a subset $BW = f_{max} - f_{min}$.

4.2 Randomized Aliasing Dynamic Frequency Scaling

We introduce Randomized Aliasing Dynamic Frequency Scaling (RAD-FS), as a side-channel security mechanism. Our method suggests replacing f_{base} with F as illustrated in Fig. 3(b). F is centered in a BW around f_{base}:

$$\mathrm{F} = [\mathrm{f}_{\mathrm{min}}, \ldots, \mathrm{f}_{\mathrm{base}}, \ldots, \mathrm{f}_{\mathrm{max}}] \tag{1}$$

Allowing for power optimization in relation with workload. For each $f_n, f_m \in F | n \neq m$ we achieve **aliased** distributions of the leakage induced by some internal value manipulation. Optimally, we aim for these disturbances to: (1) distribute uniformly to maximize the leakage entropy, (2) overlap significantly to increase the noise level for proximate time samples.

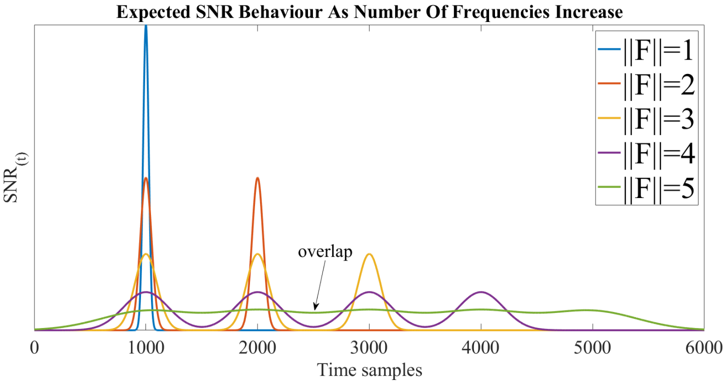


Fig. 5. After introducing RAD-FS implementation we expect that the various univariate tests will ideally demonstrate additional peaks. These will appear for each frequency in the new set, with decreasing magnitudes (ρ, SNR etc.) and increased variance as $||F||$ increases.

As we increase $||F||$, adding more frequencies to our set F, we expect security metrics e.g., the SNR, to show new distribution curves across different time samples, in correlation to various $f_n \in F$. These high leakage correlation Points Of Interest (POI), reflect the now *shifted* value manipulation corresponding with some f_n's. The overall noise increases due to cross-interference between different leakages stemming from different $f_n \in F$, as abstractly illustrated in Fig. 5. This should manifest in all univariate security metrics such as CPA, SNR, Template attacks and detection tests (we relate to other attack settings in later sections). Additionally, this randomness in frequency, increases σ_{noise} for various timing attacks, blurring the relations between data and P-state in both time and frequency domains.

4.3 Practicality and Viability

In practice, for every internal hypothesized calculation during, e.g., an encryption, we find various POI's associated with some $f_n \in F$.

In our evaluation environment (detailed below) implementing the AES cipher, it is possible to see 'ghost' peaks appearing with lesser amplitudes as $||F||$ goes up, shown in Fig. 6. The SNR value decreases, clearly owing to traces cross-interference of $f_n \neq f_m \in F$. Already, at this early stage, we can hint a significant order of magnitude improvement with only $||F|| = 5$.

Fig. 7 shows the FFT of several scenarios. In Fig. 7(a) we show the log-scale power of the single-sided spectrum when a single frequency is set ($||F|| = 1$). In Fig. 7(b) we construct F with the same frequencies used in Fig. 7(a) applying RAD-FS ($||F|| = 7$). The energy per frequency is significantly reduced as expected (note the logarithmic

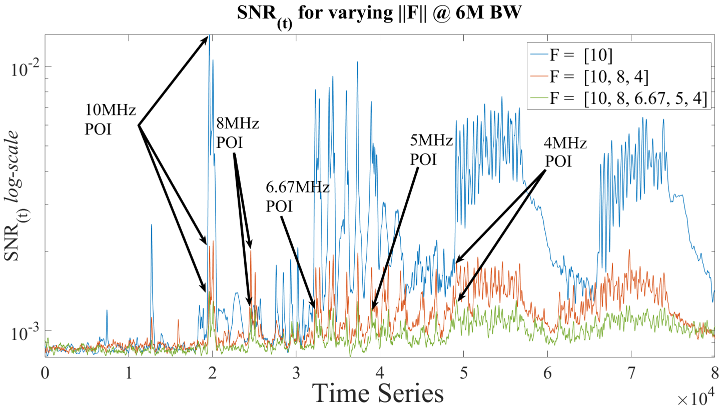


Fig. 6. Displaying arrows to indicate when peak SNR values occur relative to the chosen $f_n \in F$, for each case with different $|F|$. The Δ between the POI and the nearest secondary peaks decreases, making it more challenging to select a time sample for further analysis (i.e., template attacks).

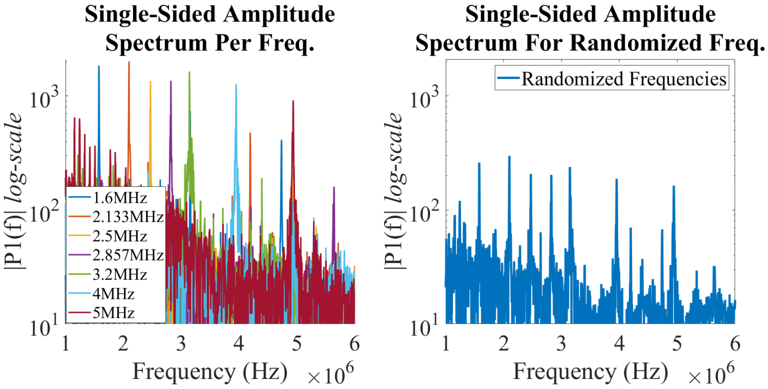


Fig. 7. FFT of different scenarios: (a) No randomization, single frequency $f_n \in F'$, $\|F\| = 1$ (b) Randomized scenario, $f_n \in F$, $\|F\| = 7$.

scale) and is distributed quasi-uniformly among $f_n \in F$. This implies that any filtering attempt will eliminate information, induce overlaps in the time domain, or alternatively, increase the noise floor.

We regard this approach as an ultra low-cost signal hiding countermeasure, as analyzed in sections below. We also conjecture it should be combined with algorithmic countermeasures such as hiding/masking with low orders d for increased efficiency [31], depending on the required security level. As the results show below, this approach alone, is found to be ultra efficient in mitigation of both remote timing and power SCA per its cost.

5 THE EVALUATED DEVICE, TESTING MODIFICATIONS AND MEASUREMENT SETUP

5.1 The Evaluated Device

Our tests were performed on the PLS15, an advanced chip made by PLSense. The PLS15 is an ULP MCU with multiple analog and digital interfaces and other capabilities, like machine learning inference engine, encryption cores, RISC-V processor and other ULP features making it a very interesting candidate for our experiment due to desirable features for IoT. The PLS15 is manufactured on the 40 *nm* TSMC process. Usually, this process node has a nominal working voltage of 1.1 *V*. The 40 *nm* low-power process was chosen due to device leakage current and dynamic power consumption savings of up to 51% compared to the 65 nm counterpart. By using mixed-threshold voltage (V_t) transistors and adaptive DVFS in the PLS15, the chip reduces operating voltage, bulk biasing, and process variation sensitivity, achieving a sub-threshold voltage of 0.45-0.6 *V* based on workload conditions. Relevant blocks on the chip are a RISC-V Core, NXP AES accelerator (unprotected), direct memory access controller and adaptive DVFS Logic. Though some other devices exist on the market incorporating DVFS and ULP process towards IoTs, we did not come across competitors reaching such deep near-threshold voltages in such a complex System-on-Chip (SoC). We have evaluated the voltage-frequency map of the PLS15 device as illustrated in Fig, 8 showing 19 discrete possible frequencies.

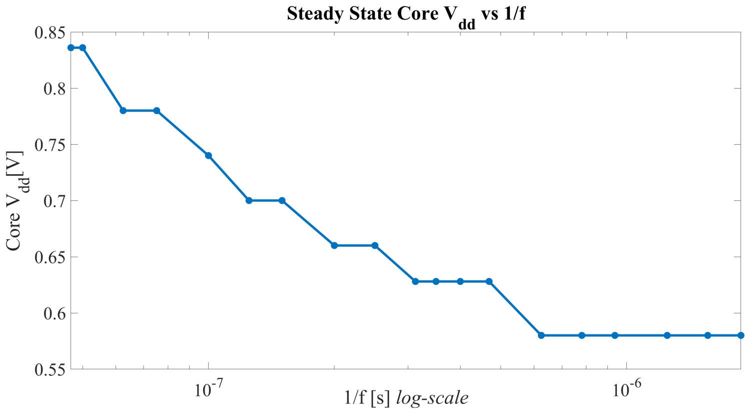


Fig. 8. Core voltage per 1/freq. available on our target device, PLS15

5.2 Testing Modifications

The PLS15 chip has a DVFS controller module that affects the chip's core voltage. The DVFS controls the input voltage to the core by utilizing an external Op-Amp in a negative feedback loop, connected to the power supply pin. We modified the PLS15 test kit board such that the DVFS is semi-enabled; The DVFS operation is in a mode where it operates based on SW commands only, and not according to workload. This gives us full control over data dependency and p-states, allowing us to reduce algorithmic noise. For power measurement, we added a jumper in series between the external core voltage regulator and the core IO pin for induction-based current sensing, utilizing the Tektronix CT1 current probe, connected through an amplifier to a Picoscope Oscilloscope as illustrated in Fig. 9.

5.3 Measurement Setup

Our test bench is composed of a PC, Picoscope 5424D, Rohde & Schwartz signal amplifier, Tektronix CT1 and PLS15 target connected as shown in Fig.10. The measurement setup is shown in Fig. 11. Note the EM probe, we aimed to perform EM SCA as well, yet the PLS15 is a flip-chip, it only exposes the backside (grounded silicon substrate) to the adversary, EM emanation is highly attenuated. Moreover, as this is an ULP device, it is indeed expected that anyway such emanation (even from the front side) would be quite weak.

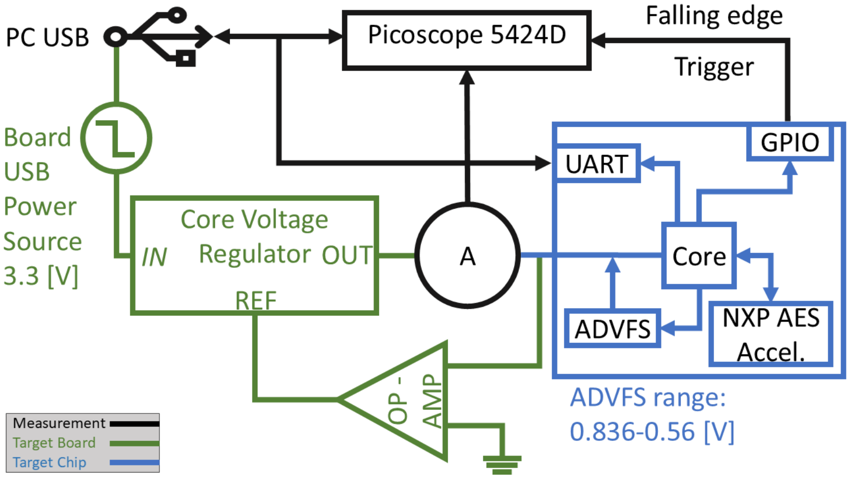


Fig. 9. Connection scheme of the core power supply and voltage regulation for the PLS15, with our current probe location.

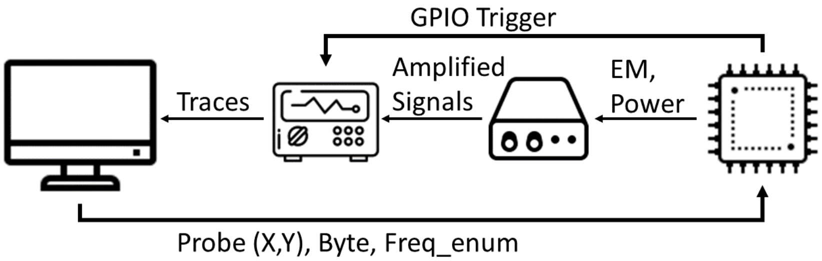


Fig. 10. Illustration of the test bench we assembled comprising a PC running a python script, Picoscope which records measurements of power traces, signal amplifier, Riscure probing station, PLS15 chip target. Icons from [24]

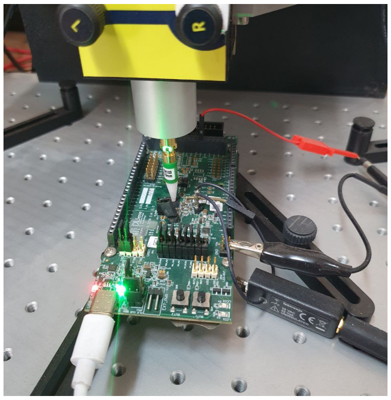


Fig. 11. Measurement setup with the PLS15 evaluation board, the CT-1 current probe connected to a power jumper.

Our experiment method pseudo-code is described in Fig. 23 of Appendix 10. For the basic SCA evaluation we wanted to eliminate algorithmic noise and to show the adversary best case scenario. Hence, we evaluate the leakage stemming from one byte manipulation leakage at a time (e.g., byte 7 in the aforementioned figure). We also control the operating frequency driven by the regulator through our python shell triggering the device, and we have added NOP cycles inline assembly to our code (implemented in C) loaded to the device, to reduce noise after a triggering event.

6 MEASUREMENTS, RESULTS AND ANALYSIS

6.1 Security Analysis Metrics

6.1.1 Low Complexity Adversary - Estimators. As discussed above we evaluate (as a starting step) Mangard's SNR [34] & Brier's CPA [10] correlation as defined by:

$$\mathrm{SNR(t)} = \frac{\mathrm{Var}_{x_i,k}(\mathrm{E}[l^t_{x_i,k}])}{\mathrm{E}_{x_i,k}(\mathrm{Var}_i[l^t_{x_i,k}]} \tag{2}$$

$$\rho_{\mathrm{l^t_{x_i,k},h^t_{x_i,k^*}}}(\mathrm{t}) = \frac{\mathrm{Cov}(l^t_{x_i,k^*}, h^t_{x_i,k^*})}{\sigma_{l^t_{x_i,k}} \sigma_{h^t_{x_i,k^*}}} \tag{3}$$

where, Var and E are the variance and expected estimators, $l^t_{x_i,k}$ is the leakage trace l in point in time samples t, taken from a cryptographic operation processing key k and plaintext (e.g.,) byte x_i.
In accordance with the correlation CPA distinguisher, we enumerate all possible (sub-) keys hypothesis k^* so as to generate the leakage hypothesis h. Then the k^* which maximizes the correlation is estimated to be the correct key. We then compare:

$$SNR_{t=POI} = max_t(|SNR|) \tag{4}$$

$$\rho_{t=POI} = max_t(|\rho|) \tag{5}$$

From now referred to as POI for the SNR & ρ accordingly. Both estimators were computed over $0.5{\cdot}10^6$ to $10{\cdot}10^6$ traces (as needed) or queries. Our results include both a RISC-V implementation of tiny-AES 128-bit code, verified against NIST [19], and an NXP crypthash hardware accelerator, embedded in the PLS15 SoC, running 128-bit AES as well. Fig. 12 shows side-by-side the SNR's of the NXP AES accelerator and the SW AES on the RISC-V processor in a comparative view, to the left and right, respectively. Note the very high SNR value achieved and the ultra fast operation of the accelerator owing to the fact that it is *seated* in its own power domain in a tailored IP block and not as part of a *sea-of-gates* as is typically the case for processor-cores.

6.1.2 Detection Test - Test Vector Leakage Assessment. As a detection test, we evaluate Test Vector Leakage Assessment (TVLA) based on the two-sided Welch's t-test [18], to show the difference in populations and the shift of data to higher statistical momentum.

$$\mathrm{T-value} = \frac{\mu_{S_f} - \mu_{S_v}}{\sqrt{\frac{\sigma_{S_f}}{\#S_f} + \frac{\sigma_{S_v}}{\#S_v}}} \tag{6}$$

It is a fast univariate method in which in our fixed versus random experiments, the test populations are leakages from two sets S_f & S_r, the compared populations are constant plaintext & a varying plaintext, this experiment set is run several times under different conditions. TVLA reflects a well known and standardized metric; as described by Eq. 6 where μ is the mean value and # denotes the size of the set. Below, we evaluate examples with T-values computed directly over the leakages (i.e., the raw first statistical moment), and we also show results of the test computed over the 2^{nd} central moment, $CM^{2,t}_s = E((l^t_s - \mu)^2)$, for completeness

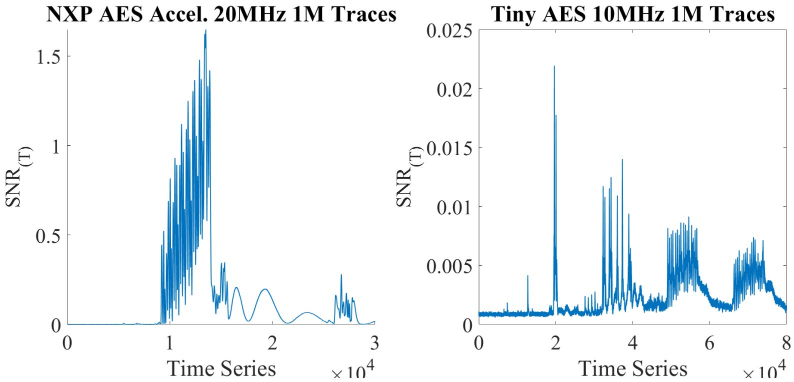


Fig. 12. SNR of the NXP AES accelerator **to the left** and the software RISC-V AES **to the right** under evaluation.

6.1.3 High complexity adversary - Templates. Template attacks [16] are performed in two consequent (or interleaved) phases of *profiling* and *attack*. It is assumed that the adversary got hold of one device for which he can program (or control) the secret key and therefore profile the leakage, and another target device from which he tries to extract information on the underlying key. We built a probability density function, for an internal manipulation, represented by function F, $y = \mathrm{f}(x_i, k)$. A set of $\mathcal{L}_p$ profiling traces of size N_p was used in order to estimate distributions, denoted as $\hat{M}_y$. Specifically, $\mathrm{f}\,(l \mid y) = \mathcal{N}\left(\widehat{\mu}_{l|y}, \widehat{\sigma}_{l|y}\right)$. For the attack phase, we utilize $\mathcal{L}_{att}$ of size N_{att} traces. The secret key k^* which maximizing the univariate maximum Likelihood (denoted by LH) is chosen: $k^* = \underset{k}{\operatorname{argmax}}\, \mathrm{LH}(k)$, i.e., $k^* = \underset{k}{\operatorname{argmax}} \prod_{j=1}^{N_{att}} \left(\mathrm{f}\,(l_i \mid y_i)\right)$. As standard, owing to practical computational reasons and numerical errors, the log-likelihood (LLH) was used [20] $k^* = \underset{k}{\operatorname{argmax}}\, \mathrm{LLH}(k) = \underset{k}{\operatorname{argmax}} \sum_{j=1}^{N_{\mathrm{att}}} \log\left(\mathrm{f}\,(l_i \mid y_i)\right)$.

6.1.4 Timing Attacks. We aim to show the relevance of our technique to timing attacks that rely on the vulnerability introduced by the very same DVFS mechanism we rely on. As an example, [49] relies on the time variance induced by DVFS, which can be manipulated by an adversary to gleam secret information via the time channel. Our technique should make it harder for an adversary to see the different distributions and to separate information about secret computation from time measurement.

6.2 Sterile Analysis - Ideal View

First, we aimed to establish that our devised method works in a sterile clean scenario, i.e., by gradually increasing the % of traces taken with altered core frequency (i.e., randomized). This is analogous to uneven weights in a distribution function. For example, 10% altered frequencies means we operate 90% of the times, for example: $f_{base} = 20\ MHz$ and 10% of the time with some other frequency in the set $f_n \in F$:

$$P(f_{base} = 20MHz) = 0.9, P(f_n \in F \setminus f_{base}) = 0.1$$

Fig. 13a, 13b show a proximate linear decrease in estimator value as the distribution nears uniform weights. Secondly, we observe differences with different BW's upon which we will rigorously discuss below. At first glance these results are easy to dismiss since the reduction is negligible, as discussed below, several degrees of freedom are available for design to enhance these results.

Setting a base line for the viability of remote timing attacks, we performed a scenario similar to [49], where plaintext HW affects the P-state. The target chip is programmed to act as a server, performing some calculation (e.g, encryption) and communication, with the adversary under the guise of a legitimate interaction, Fig. 14(a)

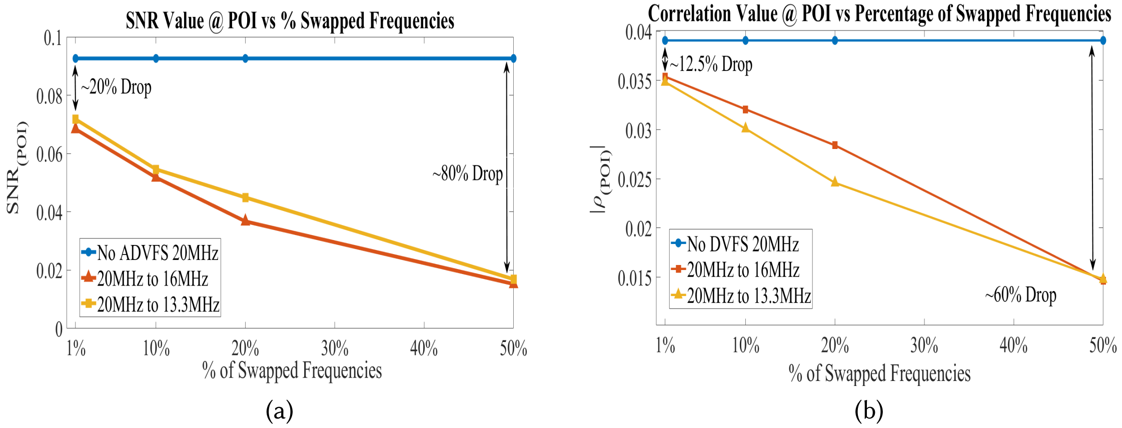


Fig. 13. Swapping from 20 *MHz* to a different frequency for a percentage of the measured queries led to a reduction in the (a)SNR or (b)ρ POI value.

shows easily distinguishable distributions in the time domain due to DVFS, we aim to introduce randomness to this process using the same scheme, proving its double effectiveness. i.e., in Fig. 14(b-c), we gradually increase $||F||$ showing how such timing distributions become gradually harder to distinguish.

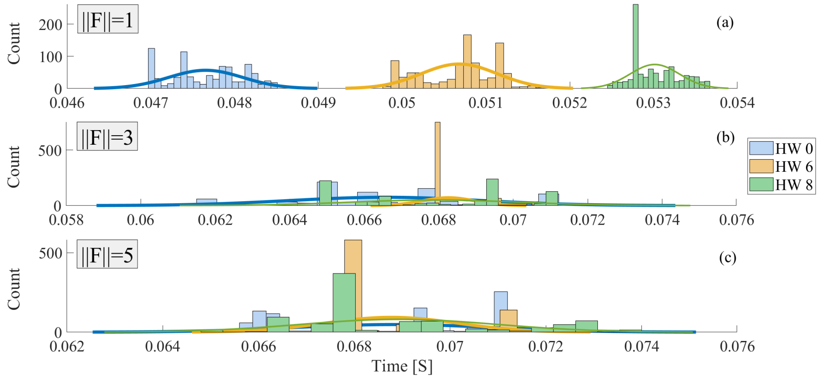


Fig. 14. Timing measured over 1000 encryptions, performed with the same HW/freq./voltage. For a worst-case analysis, all plaintext bytes are the same for minimal noise. Measured with 10-100k traces per HW for increased $||F||$.

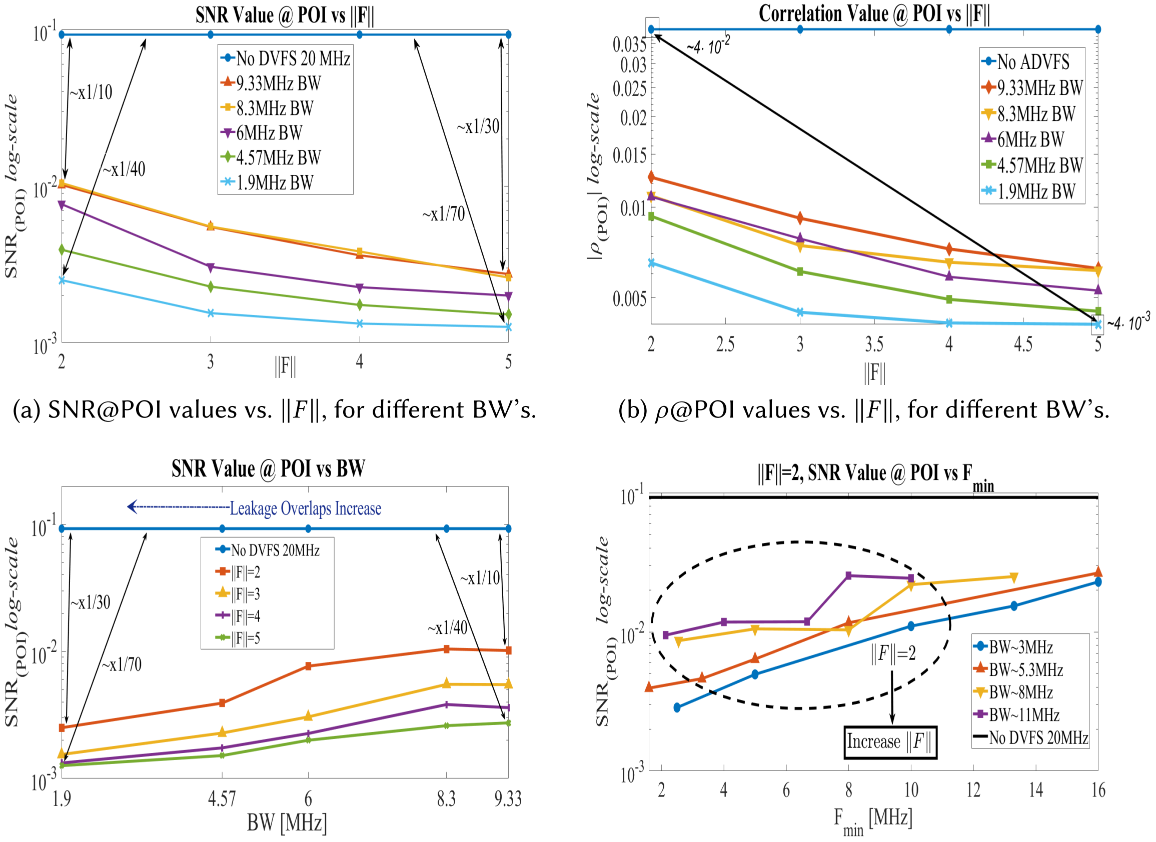
6.3 Low Complexity Adversary - Optimization parameters for RAD-FS

Under system restrictions (i.e., the discrete DVFS values described in Fig.8) we grouped frequencies within F' to isolate the parameters of BW and $||F||$ as classified in Table 2. In addition, we grouped the frequencies from F' to isolate the effects of f_{min} per given BW as partitioned in Table 5 of Appendix 10.

$||F||$**:** Randomizing frequencies from F. described in Table 2 in a uniform distribution. A reduction by 2 orders of magnitude is observable just from increasing $||F||$ to 5 frequencies (increasing $||F||$ even further is clearly possible (as outlined earlier and depends on the system under evaluation).

Table 2. Classification of frequency map of F to different BW with varying $\|F\|$s

$\|F\|$\BW	9.33[MHz]	8.3[MHz]	6[MHz]	4.57[MHz]	1.9[MHz]
$\|F\| = 2$	[16 6.67]	[13.3 5]	[10 4]	[6.67 2.1]	[4 2.1]
$\|F\| = 3$	[16 10 6.67]	[13.3 8 5]	[10 8 4]	[6.67 5 2.1]	[4 3.2 2.1]
$\|F\| = 4$	[16 13.3 10 6.67]	[13.3 10 8 5]	[10 8 6.67 4]	[6.67 5 3.2 2.1]	[4 3.2 2.5 2.1]
$\|F\| = 5$	[16 13.3 10 8 6.67]	[13.3 10 8 6.67 5]	[10 8 6.67 5 4]	[6.67 5 4 3.2 2.1]	[4 3.2 2.857 2.5 2.1]



(a) SNR@POI values vs. $\|F\|$, for different BW's.

(b) ρ@POI values vs. $\|F\|$, for different BW's.

(c) SNR@POI values vs BW

(d) SNR@POI values vs f_{min} for different BW.

Fig. 15. Various measurements results comparing the POI from optimizing various variables

A similar phenomenon is observable in the correlation graph, albeit the scale of reduction is smaller. Note that the reduction in both SNR and correlation is inversely proportional to the data/time/computation-complexity of an attack. *Therefore, two orders of magnitude are quite significant, leading to a noteworthy security level, attack complexity, etc.*

BW: By reordering the data to look at BW influence on the SNR, we show the decrease of $SNR_{t=POI}$. As demonstrated in Fig. 15c, decreasing the BW increases the overlaps (i.e., aliasing) of leakages between sets of traces from different f_n's in the time domain. A smaller BW implies interference in the frequency domain, thus the expected value of univariate analysis mixes up different time samples or internal computations. **The more the leakages overlap the larger the effect is on the estimator (e.g, SNR).**

f_{min}: To isolate the effect of f_{min}, we groupedtwo frequencies, keeping the BW constant to the best of our abilities under system limitations (listed in Table 5). Comparing Fig.15d to Fig. 15c,15a we can see that although f_{min} has an effect, it is considerably weaker than the effect of $\|F\|$ and BW.

Generally, the experiment set highlights that both the ρ and SNR estimators performed quite similarly. However, results (security gains) were slightly (by an order of magnitude) with CPA since with correlation the signal is not scaled to the noise compared to the SNR.

6.4 Detection Test - Results

In Fig. 16 we performed t-test on $0.5{\cdot}10^6$ to $10{\cdot}10^6$ traces to evaluate the security of the proposed mechanisms for different $\|F\|$, RAD-FS lowers the final value in both statistical orders. Fig. 17 shows convergence calculated over 10^6 traces, with different $\|F\|$ for both the SW implementation(a-b) and NXP hardware accelerator(c-d). It is evident from Fig. 17(a-b) that using RAD-FS shifts the data to higher statistical moments, observing $\|F\| = 1$ we see no leakage in the 2^{nd} moment, wherein $\|F\| = 3$ and above the T-value becomes significant. Even when encryption is performed on an accelerator nested with a dedicated power grid (high power signature), Fig. 17(c-d) show a steady increase in the amount of traces needed to converge to a relevant value. To summarize, a potential adversary requires higher computational effort to execute a successful attack. It is important to note that information evidenced by a detection test does not practically imply that an attack is known or easy; we show below that with the best univariate template attack success rate, the protection level provided by RAD-FS is quite remarkable. On any account, even with the t-test, adversary complexity increases by orders of magnitude. Fig 19 shows the number of traces required to disclose a secret N_{td} with $\|F\| = 2$ is 10 times higher, reaching up to 80 times more with $\|F\| = 5$. In modern DVFS systems with high frequency resolution [51] this can be further optimized, resulting in exponential gains.

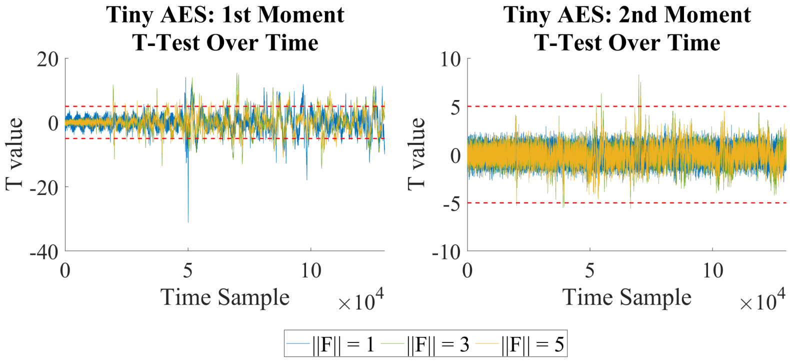


Fig. 16. **RISC-V SW:** Example TVLA over time, 10^7 traces, different $\|F\|, BW = 6MHz$.

6.5 High Complexity Adversary - Results

Our goal is to show how hard it is to make use of the information leakage measured by TVLA. As discussed in subsection 4.3, RAD-FS approach reduces attack-based metrics by orders of magnitudes. Thus, our goal was to perform model-less (profiled) evaluation utilizing templates. First, to reduce computational effort we have found

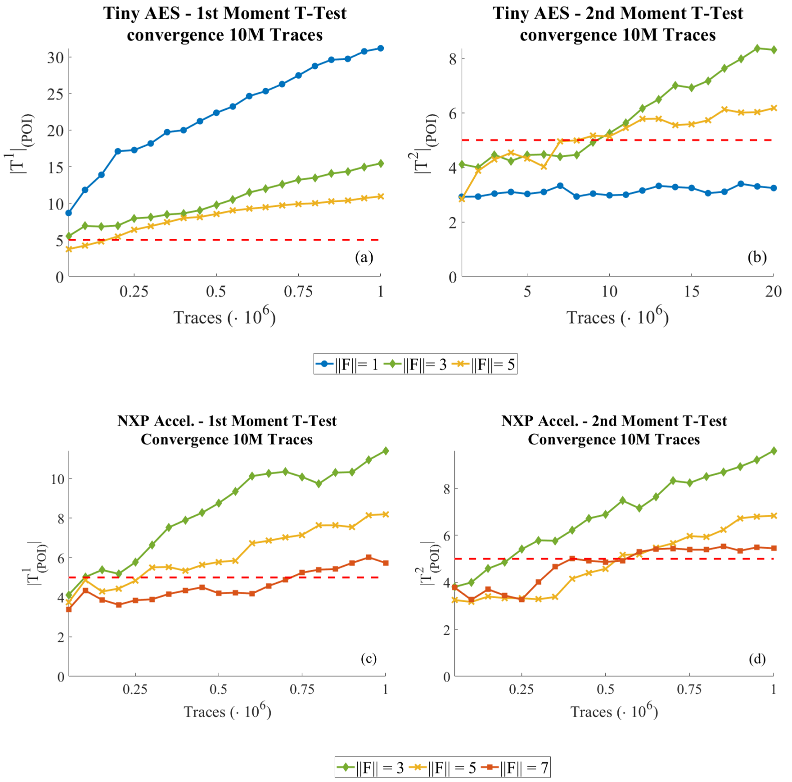


Fig. 17. T-test convergence for both SW and hardware implementations, note the significance of the 2^{nd} moment when introducing RAD-FS.

POI's using SNR. Then, we profiled leakages in a subset of time samples using a Gaussian template model. As shown in Fig. 18 for a 6-1.9 *MHz* BW the attack's Success-Rate (SR) drops rapidly both with increased $||F||$ and reduced BW. As shown in Fig. 19, higher $||F||$ requires exponentially more traces for a successful extraction of key values, while the BW inversely affects the exponential growth multiplicative constant. Whereas approximately $5 \cdot 10^5$ traces are needed to achieve a meaningful attack (SR > 0.5) with $||F|| = 1$, increasing $||F||$ to only 7 at a BW of 4.57 *MHz* raises the requirement to over $7 \cdot 10^6$ traces. Further, with $||F||$ set to 5 and a BW of 1.9 *MHz*, more than $8 \cdot 10^6$ or approximately $40 \cdot 10^6$ traces are needed, respectively. Clearly, even a small increase in $||F||$, say to 8, can exponentially increase the data complexity required for a successful attack, making it exceptionally high.

6.6 Timing Attacks & P-States

In this subsection we show RAD-FS is very relevant for such network timing attacks mitigation. Results are demonstrated via an ideal (optimal) oracle modeling the RAD-FS parameters. The constructed oracle is very *generous* with how much control is given to the adversary: prior-art discusses how bit-states influence DVFS algorithms selections. i.e., different combinations of 'on' and 'off' bits (HW), in essence generate different workloads and allow adversaries to manipulate the DVFS power control (and therefore also the timing) of a

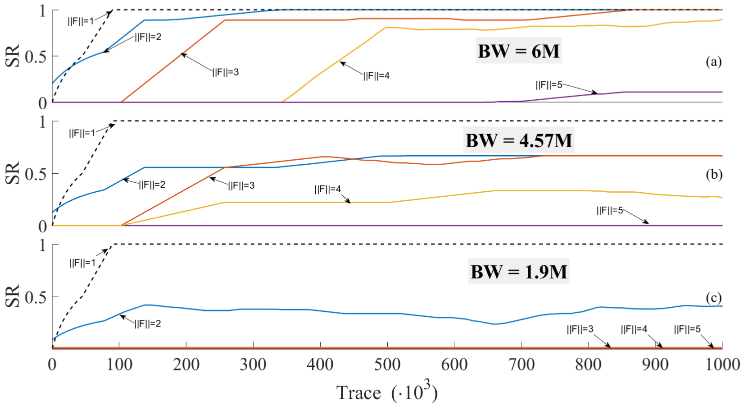


Fig. 18. **RISC-V SW Tiny-AES:** the success rate of a Gaussian template attack with varying $||F||$. Black dashed line - no DVFS at 20 *MHz*; blue - $|F|$ = 2; orange - $|F|$ = 3; yellow - $|F|$ = 4; purple - $|F|$ = 5.

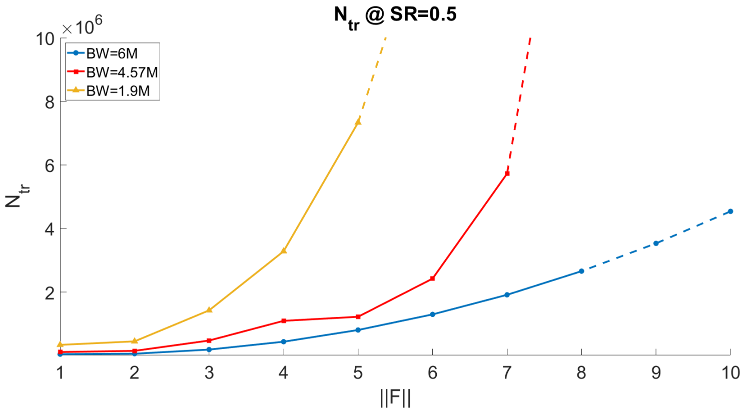


Fig. 19. **RISC-V SW Tiny-AES:** Number of traces required to achieve SR = 0.5 vs $||F||$ across different BW, dashed line denotes curve extrapolation

function call or code. All is needed is an eavesdrop adversary on the protocol (remote) in a known plaintext scenario. This is a manifestation of the mechanism relied upon in [49], the adversary manipulates the HW of the plaintext, the resulting encryption time is measured. As our defense mechanism, we emulate a firmware based solution, that soft overrides the frequency input from the Operating System (OS) and randomizes it with our proposed RAD-FS in mind: the OS requests that the DVFS switches to some f_{OS} that matches a specific HW plaintext, serving as the *generous* (most sensitive) SCA oracle while the firmware chooses a random frequency f_{RAD-FS} that has some relation to f_{OS}. This technique makes the vulnerability harder to exploit by introducing uniformly distributed noise (ideally), making the measured computation time (side-channel) harder to analyze. In the first experiment, the adversary measures the time to perform a sequence of steps: UART communication, change_clock_freq and 1000 encryptions. Fig. 14 shows that when RAD-FS is introduced at the firmware level, increasing $||F||$ causes the distributions that are easily separable with $||F|| = 1$, to alias unto one, and make it increasingly difficult for the adversary to discern them from one another. In Fig. 14(a-c) we gradually increase $||F||$ showing that even in this hard scenario of 1000 encryptions distributions start to completely overlap. Expanding upon this, in the next scenario we measured the clock cycles required to perform a set number of encryptions

(100), while changing the clock and performing UART communications. We compared changing the clock in three different intervals; 100, 50 and 10, meaning as we reduce the interval the adversary is weaker and the scenario is more realistic. 10 encryptions with the same P-state is clearly a reasonable scenario. Several interesting phenomena are visible in Fig. 20,Fig. 21,Fig. 22: **(1)** in Fig. 20 even for an interval of 10 (many clock frequency changes), the distributions are easily distinguishable with $\|F\| = 1$; Demonstrating that without RAD-FS, the DVFS mechanism introduces a strong timing bias. **(2)** introducing RAD-FS shows a significant improvement for any interval. **(3)** going from Fig. 20 to Fig. 21 we can clearly see an increasing aliasing of the distributions & confirming our intended use case. **(4)** in Fig. 22 we achieve almost uniform distributions, with full aliasing, hampering the adversaries attempt to gain any significant amount of information via the side channel. Clearly, in this respect a uniform distribution (maximum entropy) is the best one can hope for.

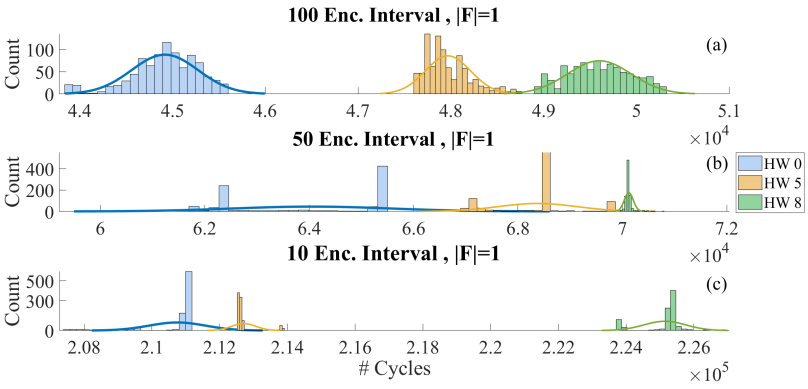


Fig. 20. Clock cycles measured over 100 encryptions, frequency changed with varying intervals. $\|F\| = 1$.

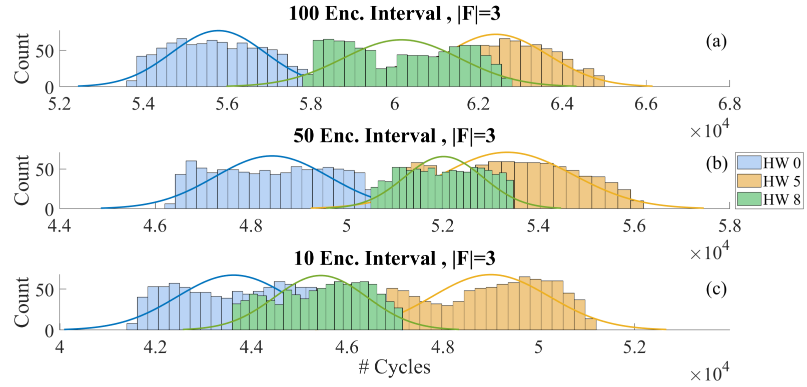


Fig. 21. Clock cycles measured over 100 encryptions, frequency changed with varying intervals. $\|F\| = 3$.

6.7 Impact on Performance

In the ideal RAD-FS scenario, performance and security are optimized when the encryption engine operates at the highest possible mean frequency (f_{base}), given the computation-intensive and high-workload mapping,

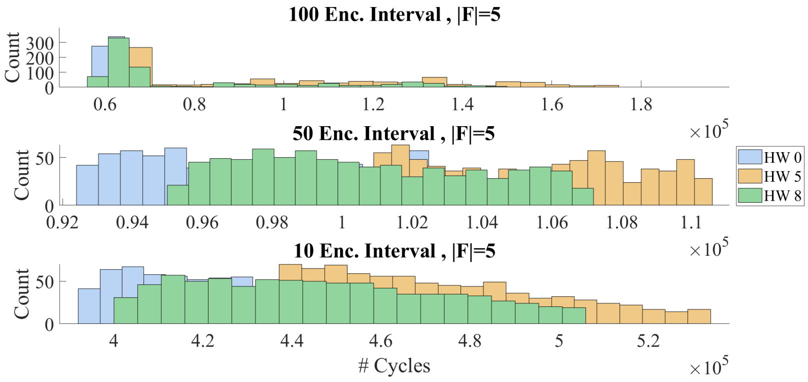


Fig. 22. Clock cycles measured over 100 encryptions, frequency changed with varying intervals. $||F|| = 5$.

concurrently with a large $|F|$ within a narrow BW. Considering IoT applications, the main parameters for optimization are area and power. For some notations, we denote by **(1)** Overhead (OH) - the performance overhead, that is, the increase in either computation time (L^{OH}), area ($Area^{OH}$) or power (P^{OH}). **(2)** m - # encryptions performed at a given f_n. **(3)** T_{switch} - the time to switch between $f_n, f_m \in F | n \neq m$ for $||F|| = k$ a total of $\binom{k}{2}$ options. **(4)** f_i - a frequency in F. That is, the total encryption time can be written by $\frac{1}{f_i} \cdot \#_{cycles}$, considering #cycles clock cycles.

$$\mathrm{T_{av}} = \frac{E[T_{switch_i}] + m \cdot E[\frac{1}{f_i} \cdot \#_{cycles}]}{m} \tag{7}$$

Generally, considering the specifications of commercial devices (and the characteristics of PLS15 itself), the switching time and the encryption time are of the same scale, $E_i[T_{switch}] \approx E_i[\frac{1}{f_i} \cdot E\#_{cycles}]$, denoted by E_T. Clearly, taking $\lim_{m\to\infty}$, $\mathrm{T_{av}} = E_T$ with 1 over m convergence. For example, taking the m=10 case:

$$\mathrm{L^{OH}}\% = \frac{T_{av}}{T_{f_{base}}} = \frac{1.1 \cdot E_T}{T_{f_{base}}} \tag{8}$$

Under our chip's limitation, for example, we can calculate the overhead using two frequencies inside a 1.33 MHz bandwidth with the following parameters: $F = \{21.33MHz, 20MHz\}, ||F|| = 2, BW = 1.33MHz$ we achieve:

$$\mathrm{L^{OH}}\% = \frac{1.1 \cdot \frac{1}{k} \cdot \sum_{i=0}^{k-1} \frac{1}{f_i}}{T_{f_{base}}} = 1.136 = 13.6\% \tag{9}$$

For this configuration,RAD-FS endures a 13.6% latency overhead, which is rather efficient compared to prior-art as discussed below. Considering the energy overhead, due to the quadratic dependency of the operation frequency, F, on the operating voltage, V_{dd}. We can approximate the voltage OH by $\Delta\mathrm{V} \sim \sqrt{\frac{1}{\Delta\mathrm{T}}} = 0.938$, concluding in:

$$\mathrm{P^{OH}}\% = \frac{\Delta\mathrm{P}}{\mathrm{P}} = \frac{\Delta\mathrm{f} \cdot \mathrm{C_{eff}} \cdot \Delta\mathrm{Vdd}^2}{\mathrm{f} \cdot \mathrm{C_{eff}} \cdot \mathrm{Vdd}^2} = \frac{1.136 \cdot 0.938^2}{1} = 99\% \tag{10}$$

Under our chip's limited options, and taking the above estimations, we estimate the P^{OH} to be improved by 1%. Actual commercial devices numbers may even be better with more granular freq./voltage steps.

6.8 Results Summary

To summarize our results: **(1)** we start by generally demonstrating that with a simple distinguisher our concept has merit as shown by Fig. 13a and Fig. 13b. The more uniform the frequency distribution, the less information is leaked **(2)** We continue to deep dive into the different parameters such a technique would utilize, using various estimators and SCA methodologies. From Fig. 15b we conclude that CPA is weak in our scenario and focus on TVLA & SNR. **(3)** We isolated the different variables, $||F||$, BW and f_{min}, into different sets of measurements to determine which is the most prudent. Comparing results from Fig. 15c with Fig. 15a to Fig. 15d, from the difference in the slope and distribution of the measurements, we can conclude that the best scenario requires as many frequencies as possible within the smallest BW possible to generate *overlaps* in the frequency domain, which in turn are harder to filter out in the time domain. **(4)** Using TVLA (Fig. 17), our objective was to show two things. First, security increases; that is to say, the two compared populations (fixed and randomized) are harder and harder to distinguish to a meaningful extent, as evident by the higher number of traces needed to converge and the final convergence value. Secondly the shift of information to higher statistical moments, seen in Fig. 17(a-b) the only measurement showing no information in the second moment is the *non RAD-FS* measurement set, proving RAD-FS introduces computational complexity for an adversary. **(5)** Gaussian template based attacks - we chose a strong(er) attack model to show that even in a scenario where the adversary is knowledgeable can profile a device and take large amount of profiling traces, a successful attack is still not trivial (regardless of the univariate TVLA results). Fig. 19 shows that $N_{tr} \sim \exp\{BW \cdot ||F||\}$. **(6) Identifying the strength of RAD-FS in protecting against remote timing-based attacks**, we show clear aliasing in the time domain in a scenario where the adversary has a way to manipulate the DVFS mechanism either directly (via SW) or via an oracle (as done in [49]). By randomizing the frequency, we reduce the direct relation between workload and frequency, making it harder to discern information about the encryption from the run time. This is *important as several industry standard encryption schemes, both symmetric and asymmetric public key encryptions are vulnerable to such attacks.*

7 COMPARISON WITH EXISTING LITERATURE

To fully compare RAD-FS performance, as calculated in Subsection 6.7, with the existing literature, it is important to discuss both masking and hiding countermeasures in both hardware and software.

7.1 Masking

Masking is a powerful technique used in cryptography in general and in SCA security specifically by reducing the dependency of a device's physical emissions on sensitive data. In a masked implementation, sensitive variables are split into multiple $d + 1$ random looking shares, where d denotes the security order. It is done in a way which fulfills the independence assumption where the leakage of each share is independent in all others. For linear operations each share is processed separately, ensuring that no single intermediate variable contains useful information about the original data. And when non-linear operations take place special *refresh* mechanisms, which require more randomness, are required to ensure shares do not recombine. Masking is effective against DPA and similar attacks. However, masking is costly and typically resulting in quadratic cost (either latency or area) with the security order d, driven by the cost of non-linear operations. this in turn translates into OH in terms of area utilization, power, computation time and additional randomness required, as more resources are required to handle and combine the shares securely.

Table 3. Comparison of various masking methods. N_{td} notes the number of traces required to disclose information above the threshold of the d^{th} statistical order

Source	Variant	$Area^{OH}$ (kGE)	L^{OH}	P^{OH} mW	$N_{td}@d^{th}$ order
[22, 23][1]	d=2	$\geq x2$	$\geq x3$[2]	Not provided	$\sim 0.001 \cdot 10^6$
[22, 23][1]	d=3	$\geq x3.25$	$\geq x3$	Not provided	$0.01 \cdot 10^6$
[15][3]	d=2	$\geq x1.9$	$\geq x3$	7	$0.24 \cdot 10^6$
[15][3]	d=3	$\geq x3.5$	$\geq x3$	10	$6 \cdot 10^6$
RAD-FS[4]	$\|\|F\|\| = 3$	0	$\geq x1.14$	0	$0.25 \cdot 10^6$
RAD-FS[4]	$\|\|F\|\| = 7$	0	$\geq x1.14$[5]	0	$0.5 \cdot 10^6$

Table 3 compares security in terms of the number of traces needed to capture information from the d^{th}-order t-test where the RAD-FS comparison point is in the scale of $1 \cdot 10^6$ traces needed to capture information, which is pushed to the 2^{nd} order. We chose this comparison parameter because: **(1)** RAD-FS shifts data to the 2^{nd} order (no information with less traces from the first moment), **(2)** higher statistical orders (3^{rd} and above) don't hold significance with RAD-FS in the univariate setting if $||F||$ is uniformly randomized. As shown, when compared with Domain-Oriented Masking (DOM) and Hardware Private Circuits (HPC) Masking techniques which are quite SOTA in hardware: RAD-FS achieves far better cost-per-security trade-off as compared to masking with $d = 2$, in area, power and latency within an acceptable range.

When comparing to high order masking with $d \geqslant 3$ RAD-FS clearly still outperform in all electrical cost factors, however only borderline meets the same security level. This can be improved for example with increasing $||F||$. It is important to note that high-order masking solutions will be quite challenging fo fit into ULP system constraints, due to extremely large area overhead whereas RAD-FS practically have no area or power overheads (in average). As a last highlight, masking does not offer a proven (or even analyzed) solution to remote timing attacks and P-state leakage **per se**, as we have with RAD-FS: the shared masked design will trigger and be under the influence of the DVFS mechanism, and power SCA leakage in higher orders can be translated to higher orders in timing owing to the DVFS power management. As such we believe our technique holds merit and applicability for both servers (vulnerable to timing attacks) and edge devices (vulnerable to DPA).

7.2 Cost of Shuffling

Table 4. *SW implementation, **Hardware analog implementation, ***Hardware analog & custom memory implementation.

Source	$Area^{OH}$	L^{OH}	P^{OH}	$Effort^{OH}$
[44]	+6.6%	+17.4%	-3.5%	**
[21]	+20%	0	+24%	**
[29]	+10%	+0.07%	+8%	***
RAD-FS	0	+13.6%	-1%	*

[1] Results from Domain-Oriented Masking (DOM) masking

[2] In these reports the multiplicative depth of the S-boxes is two meaning minimum pipeline of three stages which will, to the very minimum, increase latency by a factor of three (though it is a lower-bound and in practice values reported are much worse)

[3] Results for Hardware Private Circuits (HPC1 gadgets) masking

[4] NXP hardware implementation, 2nd order t test

[5] Latency depends on BW, not $||F||$

Table 4, compares the latency and power overheads calculated in subsection 6.7 to existing literature dealing with frequency modulation, considering implementation effort, which roughly estimates resources (dedicated design, engineering, verification, etc.) needed to implement different solutions and the flexibility offered by different approaches; SW implementation which requires some SW code, hardware analog implementation which requires additional silicone area and dedicated hardware design, hardware analog design with dedicated memory design that requires custom memory cells in addition to analog design. As shown above our solution aims at ULP MCU's and IoT chips, where area and power are the main optimization concerns. Due to using a block that preexists within such devices, we present 0 area OH, with low implementation effort, requiring only a SW implementation. Using a SW solution, we sacrifice latency within an acceptable margin even when comparing to SOTA. This can be further improved upon, implementing RAD-FS in assembly and not in high level C code.

7.3 Post Quantum Cryptography

Post-Quantum Cryptography (PQC) refers to cryptographic algorithms designed to secure data against the potential threats posed by quantum computers. While current cryptographic methods, such as RSA and ECC, are considered secure against classical computers, the advent of quantum computing poses a threat to these systems due to quantum algorithms that can factor large numbers exponentially faster. To establish a standard for PQC, NIST has selected CRYSTALS-Kyber [3] as its key encapsulation mechanism, which is a lattice-based scheme for quantum-safe public key encryption. In a work published [53] while this study was under review, the PQC relevance of our work was proven. Hints from Hertz show that under standard DVFS conditions, CRYSTALS-Kyber and implementations of it using NTTRU [33] are highly vulnerable to timing attacks and leak data correlative to HW in the frequency domain, highlighting the potential benefit of RAD-FS to PQC primitives.

8 ASSUMPTIONS AND REAL-LIFE

We will now outline the features of our protection mechanism by identifying the essential requirements for a successful attack:

1. Perfect synchronization and measurement triggering - this is how the analysis is done in this paper/analysis. However, in real-life scenarios, achieving synchronization and establishing a measurement-triggering mechanism is more challenging, introducing slight overhead for the adversary.
2. Pre-knowledge on *Trace Length* - when we randomize frequencies using RAD-FS, the adversary clearly does not know the number of samples needed to be captured, as it depends on the randomized frequency. Therefore, the best scenario is to take some fixed number of samples which will imply mixtures of frequencies appearing in the captured leakage even if f_n is only randomized once per several encryptions.
3. Isolating the relevant leakages is harder in a parallel computation scenario. Running on multiple cores may increase the algorithmic noise and will linearly harden the adversary's ability to filter out leakages not correlating with the hypothesis data manipulation.

In addition, it is important to emphasize that real-life applications with more conventional DVFS mechanisms are differ from ours:

1. DVFS resolution is very high - even hundreds of P-states [1, 36, 46, 51]. That is, the security parameters such as $\|F\|$, BW, f_{min} can be significantly optimized.
2. Modern DVFS solutions optimize each core individually depending on the workload, adding a plethora of protection flexibility.
3. As discussed in Subsection 4.3, filtering attempts may conceal information and induce distributions overlaps thus increasing the noise.

9 CONCLUSIONS AND FUTURE RESEARCH

In this paper we demonstrate RAD-FS, a new security technique that is scalable, software-based, and easy to implement, that applies to most if not all modern microchips. Improving protection against DPA and timing SCA attacks in orders of magnitudes. Discussing several different estimators under DPA, we show the radical effect achieved by RAD-FS on the analyzed estimator and its' convergence. We show the effects of our parameters, and conclude that the ideal scenario revolves around large $||F||$ within the narrowest BW considering system limitations, as to increase aliasing. Moreover, it substitutes the naive solution against timing attacks (shutting off the DVFS controller, e.g., Hertzbleed) and enables an SCA secure chip coexisting with DVFS optimization. We discussed a countermeasure for the inherent weakness in power/freq.-data dependency.

In future work, we would like to test adding “fuzziness” to the RAD-FS process though uniformly distributed amount of “NOP” asm commands before/within the encyption operation, to touch upon multi-DVFS-model security analysis, and combine this ultra low-cost solution with additional layers of security, and proceed with in depth electromagnetic SCA analysis with the proposed mechanism.

ACKNOWLEDGEMENTS

We sincerely thank the *PLSense* team (today acquired by NXP), namely, Uzi Zangi, Arnon Kaminsky, Shahar Dinar, Shaked Matzner and Tzach Hadas: for their invaluable support and guidance while working with their development environment and beta (development and testing) boards and setups, such assistance and dedication towards innovative science is not trivial for such a small start-up.

DECLARATIONS

Itamar Levi and Daniel Dobkin were partially funded by the Israel Innovation Authority (IIA), Bio-Chip Consortium grant file No. 75696, Israel, and Israel Science Foundation (ISF) grant 2569/21.

10 APPENDIXES

Table 5. Classification of frequency map of F' to different BWs with a varying F_{min}

~const BW\\F_{min}	F_{min}	~$2F_{min}$	~$3F_{min}$	~$4F_{min}$	~$5F_{min}$	~$6F_{min}$	~$13F_{min}$
BW~3 *MHz*	[5, 2.5]	[8, 5]	[13.3, 10]	[16, 13.3]		[20, 16]	
BW~5.3 *MHz*	[6.67, 1.28]	[8, 2.857]	[8,3.3]	[10, 5]			[21.3, 16]
BW~8 *MHz*	[10, 2.5]	[13.3, 5]	[16, 8]	[20, 10]	[21.33, 13.3]		
BW~11 *MHz*	[13.3, 2.13]	[16, 4]	[16, 6.67]	[20, 8]	[21.33, 10]		

+

REFERENCES

[1] Acun, B., Chandrasekar, K., and Kale, L. V. Fine-grained energy efficiency using per-core dvfs with an adaptive runtime system. In *2019 Tenth International Green and Sustainable Computing Conference (IGSC)* (2019), pp. 1–8.

[2] Agrawal, D., Archambeault, B., Rao, J. R., and Rohatgi, P. The em side—channel (s). In *International workshop on cryptographic hardware and embedded systems* (2002), Springer, pp. 29–45.

[3] Alagic, G., Alagic, G., Apon, D., Cooper, D., Dang, Q., Dang, T., Kelsey, J., Lichtinger, J., Liu, Y.-K., Miller, C., et al. Status report on the third round of the nist post-quantum cryptography standardization process.

[4] AlFardan, N. J., and Paterson, K. G. Lucky thirteen: Breaking the tls and dtls record protocols. In *2013 IEEE Symposium on Security and Privacy* (2013), IEEE, pp. 526–540.

[5] Bao, W., Hong, C., Chunduri, S., Krishnamoorthy, S., Pouchet, L.-N., Rastello, F., and Sadayappan, P. Static and dynamic frequency scaling on multicore cpus. *ACM Transactions on Architecture and Code Optimization (TACO) 13*, 4 (2016), 1–26.

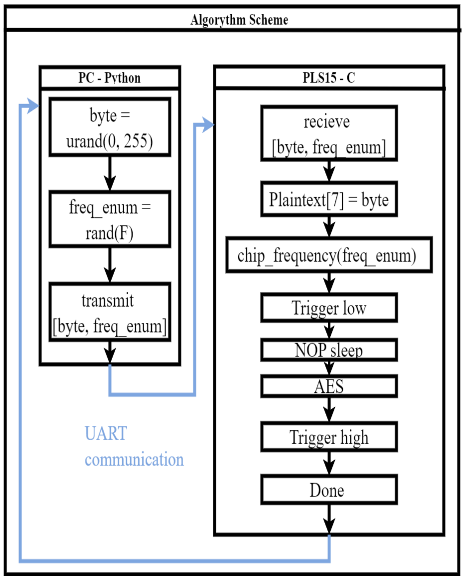


Fig. 23. Experiment pseudo-code Flowchart

[6] Bleichenbacher, D. Chosen ciphertext attacks against protocols based on the rsa encryption standard pkcs #1. In *Annual International Cryptology Conference* (1998), Springer, pp. 1–12.

[7] Bos, J. W., Halderman, J. A., Heninger, N., Moore, J., Naehrig, M., and Wustrow, E. Elliptic curve cryptography in practice. In *Financial Cryptography and Data Security: 18th International Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014, Revised Selected Papers 18* (2014), Springer, pp. 157–175.

[8] Breuer, R., Standaert, F.-X., and Levi, I. Fully-digital randomization based side-channel security—toward ultra-low cost-per-security. *IEEE Access 10* (2022), 68440–68449.

[9] Brier, E., Clavier, C., and Olivier, F. Correlation power analysis with a leakage model. In *International workshop on cryptographic hardware and embedded systems* (2004), Springer, pp. 16–29.

[10] Brier, E., Clavier, C., and Olivier, F. Correlation power analysis with a leakage model. In *Cryptographic Hardware and Embedded Systems-CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings 6* (2004), Springer, pp. 16–29.

[11] Brumley, B. B., and Tuveri, N. Remote timing attacks are still practical. In *European Symposium on Research in Computer Security* (2011), Springer, pp. 355–371.

[12] Brumley, D., and Boneh, D. Remote timing attacks are practical. *Computer Networks 48*, 5 (2005), 701–716.

[13] Canella, C., Bulck, J. V., Schwarz, M., Lipp, M., von Berg, B., Ortner, P., Piessens, F., Evtyushkin, D., and Gruss, D. A systematic evaluation of transient execution attacks and defenses, 2019.

[14] Carts, D. A. A review of the diffie-hellman algorithm and its use in secure internet protocols. *SANS institute 751* (2001), 1–7.

[15] Cassiers, G., Grégoire, B., Levi, I., and Standaert, F.-X. Hardware private circuits: From trivial composition to full verification. *IEEE Transactions on Computers* (2020).

[16] Chari, S., Rao, J. R., and Rohatgi, P. Template attacks. In *International Workshop on Cryptographic Hardware and Embedded Systems* (2002), Springer, pp. 13–28.

[17] Chari, S., Rao, J. R., and Rohatgi, P. Template attacks. In *Cryptographic Hardware and Embedded Systems-CHES 2002: 4th International Workshop Redwood Shores, CA, USA, August 13–15, 2002 Revised Papers 4* (2003), Springer, pp. 13–28.

[18] Cooper, J., Mulder, E. D., Goodwill, G., Jaffe, J., Kenworthy, G., and Rohatgi, P. Test vector leakage assessment (TVLA) methodology in practice (extended abstract). ICMC 2013.

[19] Dworkin, M. Recommendation for block cipher modes of operation. methods and techniques. Tech. rep., National Inst of Standards and Technology Gaithersburg MD Computer security Div, 2001.

[20] Fei, Y., Ding, A. A., Lao, J., and Zhang, L. A statistics-based fundamental model for side-channel attack analysis. *Cryptology ePrint Archive* (2014).

[21] Ghosh, A., Das, D., and Sen, S. Physical time-varying transfer functions as generic low-overhead power-sca countermeasure. *arXiv preprint arXiv:2003.07440* (2020).

[22] Gross, H., and Mangard, S. Reconciling masking in hardware and software. In *International Conference on Cryptographic Hardware and Embedded Systems* (2017), Springer, pp. 115–136.

[23] Gross, H., Mangard, S., and Korak, T. Domain-oriented masking: Compact masked hardware implementations with arbitrary protection order. *Cryptology ePrint Archive* (2016).

[24] Icons8. Scientific icons, 2023. https://icons8.com/ [Accessed: (1.06.2023].

[25] Inc, A. Arm big. little, 2022. https://www.arm.com/technologies/big-little [Accessed: (10.07.2023].

[26] Kocher, P., Jaffe, J., Jun, B., and Rohatgi, P. Introduction to differential power analysis. *Journal of Cryptographic Engineering 1*, 1 (2011), 5–27.

[27] Kocher, P. C. Cryptanalysis of diffie-hellman, rsa, dss, and other systems using timing attacks. In *Extended abstract* (1995), Citeseer.

[28] Kogler, A., Juffinger, J., Giner, L., Gerlach, L., Schwarzl, M., Schwarz, M., Gruss, D., and Mangard, S. Collide+power: Leaking inaccessible data with software-based power side channels. In *USENIX Security* (2023).

[29] Kumar, R., Liu, X., Suresh, V., Krishnamurthy, H. K., Satpathy, S., Anders, M. A., Kaul, H., Ravichandran, K., De, V., and Mathew, S. K. A time-/frequency-domain side-channel attack resistant aes-128 and rsa-4k crypto-processor in 14-nm cmos. *IEEE Journal of Solid-State Circuits 56*, 4 (2021), 1141–1151.

[30] Le Sueur, E., and Heiser, G. Dynamic voltage and frequency scaling: The laws of diminishing returns. In *Proceedings of the 2010 international conference on Power aware computing and systems* (2010), pp. 1–8.

[31] Levi, I., Bellizia, D., Bol, D., and Standaert, F.-X. Ask less, get more: Side-channel signal hiding, revisited. *IEEE Transactions on Circuits and Systems I: Regular Papers 67*, 12 (2020), 4904–4917.

[32] Levi, I., Bellizia, D., and Standaert, F.-X. Beyond algorithmic noise or how to shuffle parallel implementations? *International Journal of Circuit Theory and Applications 48*, 5 (2020), 674–695.

[33] Lyubashevsky, V., and Seiler, G. Nttru: truly fast ntru using ntt. *Cryptology ePrint Archive* (2019).

[34] Mangard, S. Hardware countermeasures against dpa–a statistical analysis of their effectiveness. In *Topics in Cryptology–CT-RSA 2004: The Cryptographers' Track at the RSA Conference 2004, San Francisco, CA, USA, February 23-27, 2004, Proceedings* (2004), Springer, pp. 222–235.

[35] Mangard, S., Oswald, E., and Popp, T. *Power analysis attacks: Revealing the secrets of smart cards*, vol. 31. Springer Science & Business Media, 2008.

[36] Mei, X., Wang, Q., and Chu, X. A survey and measurement study of gpu dvfs on energy conservation. *Digital Communications and Networks 3*, 2 (2017), 89–100.

[37] Menezes, A. J., Van Oorschot, P. C., and Vanstone, S. A. *Handbook of applied cryptography*. CRC press, 2018.

[38] Messerges, T. S., Dabbish, E. A., and Sloan, R. H. Investigations of power analysis attacks on smartcards. *Smartcard 99* (1999), 151–161.

[39] Nassi, B., Iluz, E., Cohen, O., Vayner, O., Nassi, D., Zadov, B., and Elovici, Y. Video-based cryptanalysis: Extracting cryptographic keys from video footage of a device's power led. *Cryptology ePrint Archive* (2023).

[40] Padoin, E. L., Pilla, L. L., Castro, M., Boito, F. Z., Alexandre Navaux, P. O., and Méhaut, J.-F. Performance/energy trade-off in scientific computing: the case of arm big. little and intel sandy bridge. *IET Computers & Digital Techniques 9*, 1 (2015), 27–35.

[41] Pering, T., Burd, T., and Brodersen, R. The simulation and evaluation of dynamic voltage scaling algorithms. In *Proceedings of the 1998 international symposium on Low power electronics and design* (1998), pp. 76–81.

[42] Quisquater, J.-J., and Samyde, D. Electromagnetic analysis (ema): Measures and counter-measures for smart cards. In *International Conference on Research in Smart Cards* (2001), Springer, pp. 200–210.

[43] Salomon, D., and Levi, I. Masksimd-lib: on the performance gap of a generic c optimized assembly and wide vector extensions for masked software with an ascon-p test case. *Journal of Cryptographic Engineering* (2023), 1–18.

[44] Singh, A., Kar, M., Mathew, S. K., Rajan, A., De, V., and Mukhopadhyay, S. Improved power/em side-channel attack resistance of 128-bit aes engines with random fast voltage dithering. *IEEE Journal of Solid-State Circuits 54*, 2 (2019), 569–583.

[45] Standaert, F.-X. Introduction to side-channel attacks. In *Secure integrated circuits and systems*. Springer, 2010, pp. 27–42.

[46] Su, B., Gu, J., Shen, L., Huang, W., Greathouse, J. L., and Wang, Z. Ppep: Online performance, power, and energy prediction framework and dvfs space exploration. In *2014 47th Annual IEEE/ACM International Symposium on Microarchitecture* (2014), pp. 445–457.

[47] Turan, M. S., McKay, K., Chang, D., Bassham, L. E., Kang, J., Waller, N. D., Kelsey, J. M., and Hong, D. Status report on the final round of the nist lightweight cryptography standardization process.

[48] Veyrat-Charvillon, N., Medwed, M., Kerckhof, S., and Standaert, F.-X. Shuffling against side-channel attacks: A comprehensive study with cautionary note. In *International Conference on the Theory and Application of Cryptology and Information Security* (2012), Springer, pp. 740–757.

[49] Wang, Y., Paccagnella, R., He, E. T., Shacham, H., Fletcher, C. W., and Kohlbrenner, D. Hertzbleed: Turning power {Side-Channel} attacks into remote timing attacks on x86. In *31st USENIX Security Symposium (USENIX Security 22)* (2022), pp. 679–697.

[50] Wang, Y., Paccagnella, R., Wandke, A., Gang, Z., Garrett-Grossman, G., Fletcher, C. W., Kohlbrenner, D., and Shacham, H.

Dvfs frequently leaks secrets: Hertzbleed attacks beyond sike, cryptography, and cpu-only data. In *2023 IEEE Symposium on Security and Privacy (SP)* (2023), pp. 2306–2320.

[51] Xinxin Mei, Ling Sing Yung, K. Z., and Chu, X. Gpu dvfs, 2013. https://dl.acm.org/doi/pdf/10.1145/2525526.2525852 [Accessed: (10.07.2023].

[52] Yarom, Y., and Falkner, K. {FLUSH+ RELOAD}: A high resolution, low noise, l3 cache {Side-Channel} attack. In *23rd USENIX security symposium (USENIX security 14)* (2014), pp. 719–732.

[53] Yu, T., Cheng, C., Yang, Z., Wang, Y., Pan, Y., and Weng, J. Hints from hertz: Dynamic frequency scaling side-channel analysis of number theoretic transform in lattice-based kems. *Cryptology ePrint Archive* (2024).

[54] Zhai, B., Blaauw, D., Sylvester, D., and Flautner, K. Theoretical and practical limits of dynamic voltage scaling. In *Proceedings of the 41st Annual Design Automation Conference* (2004), pp. 868–873.