Efficient Key Extraction from the Primary Side of a Switched-Mode Power Supply

Sami Saab, Andrew Leiserson, and Michael Tunstall

Rambus Cryptography Research
425 Market St Fl 11, San Francisco CA 94105-2496, United States
{sami.saab, andrew.leiserson, michael.tunstall}@cryptography.com
http://www.cryptography.com/

Abstract. In this paper we detail techniques that can be used to analyze and attack an AES implementation on an FPGA from the primary (i.e., external) side of a switched-mode power supply. Our attack only requires measurements of the duty cycle of the power supply, and then increases the signal-to-noise ratio (SNR) though averaging, deconvolution and wavelet based detrending. The result is an exploitable source of leakage that allows a secret key to be determined from low-frequency power measurements. The techniques and procedures provide a general approach to performing differential power analysis (DPA) from a single point of information for any single hypothesized intermediate value, suggesting their potential for improving other types of side-channel analysis as well.

Keywords: Side-Channel Analysis, DPA, Switched-Mode Power Supply, Deconvolution, Detrending, Wavelets

1 Introduction

Side-channel analysis was proposed as a method of extracting cryptographic keys by Kocher [11], who noted that the time required to compute an RSA signature could reveal a private key. Further work demonstrated that one could determine cryptographic keys by observing the power consumption over time [12]. Two types of attacks were proposed. The first was inspecting a single power consumption trace, referred to as Simple Power Analysis (SPA), and a statistical treatment of a set of traces, referred to as Differential Power Analysis (DPA). It was later shown that one could use the same treatment on traces taken using electromagnetic probes [7, 17], where the equivalent attacks are typically referred to as Simple Electromagnetic Analysis (SEMA) and Differential Electromagnetic Analysis (DEMA), respectively. With adequate sampling rates, proximity, and absence of countermeasures, side-channel attacks have been practically demonstrated on a wide variety of devices ranging from small single purpose chips [2, 6, 10] to large general purpose SoCs and CPUs [8, 16, 20].

In this paper, we consider an implementation in a tamper resistant enclosure such that no meaningful power or electromagnetic measurements can be made.

While examples of such metal cases that leak information are present in the literature [8], we shall assume that this avenue of attack is not available to an attacker. In such a case, an attacker would seek to determine the power consumption of the entire device by focusing on the external power supply. For example, the primary (i.e., external) line of its internal switched-mode power supply (SMPS) that supplies its internal components. The nature of an SMPS means that the power consumption over time is not directly correlated with the power consumption of a device it is powering. One might assume that the combination of a secure enclosure and an SMPS would be sufficient to protect a device from side-channel analysis.

In this paper, we demonstrate a key extraction attack using only duty cycle information taken from the primary side of an SMPS. The measurements reduce the information from a trace to a single value that summarizes the leakage for a period of time. We show how this leakage can be detected and then exploited to allow cryptographic keys to be retrieved. Furthermore, we also describe how the same techniques can improve other types of side-channel analysis.

The remainder of this paper is organized as follows. In Section 2 we describe how a switched-mode power supply operates, why leakage is visible on the primary side of the power supply, and the basic phenomenon that allows one to detect power usage on its secondary side. In Section 3, we describe how we can validate the efficacy of different measurement techniques and how to work around the constraints imposed on how we measure power usage on the secondary side of an SMPS from its primary side. In Section 4, we discuss results on validating the efficacy of the proposed measurement techniques. In Section 5, we discuss results on performing a key extraction using the proposed measurement technique in addition to other signal processing techniques required to make the attack successful. We conclude in Section 6.

2 Switched-Mode Power Supplies

Switched-mode power supplies (SMPS) have gained popularity over the past 35 years because of their higher efficiency, lower operating temperature, compactness and light weight (when compared to other power supplies such as linear regulators). Their efficiency stems from their switching characteristic, by connecting/disconnecting the input, or primary, to the load, or secondary. Many SMPS configurations have been developed to achieve various performance characteristics. By modulating the width of a pulse of the full input voltage, and low-pass filtering that pulse, an SMPS can control the secondary voltage level and adjust according to changing load characteristics. The trade-offs associated with using an SMPS include a more complex design and high-amplitude, high-frequency spikes caused by switching. The injected spikes on the power line need to be filtered, adding further complexity.

The nature of an SMPS might lead one to conjecture that power measurements from its primary side contain only minimal information about the power consumed on its secondary side. One strategy might be to integrate over time a measurement of power on the primary side, and incorporate an accurate efficiency characterization of the SMPS, to approximate the energy consumed on the secondary side. Yet another argument could state that the ground, or return line, on the primary side, never disconnects from the secondary side, hence measuring the return power on the primary side could contain power usage information from the secondary side, albeit with reduced resolution because of the required capacitance on the primary side.

While the above arguments do not allow for effective DPA-based key extraction in practice, monitoring the return line has been demonstrated to expose leakage with potential to extract keys in low-noise environments. High-amplitude, high-frequency spikes on the input lines, that typically manifest as noise in a side-channel attack, nonetheless provide power usage information through their locations in time. Furthermore, given that the spikes are high in amplitude and frequency, they can be observed through noisy environments, filtering, over distance, and through shielding. In fact, the spike locations in time provide enough information for successful side-channel attacks.

2.1 Voltage Spike Interpretation and Detection

When operating, an SMPS generates voltage spikes on its primary side. The cause of these voltage spikes stem from a combination of the fundamental mode of operation of an SMPS, and unavoidable practical physical attributes found in all electrical circuits (specifically, switching and inductance). SMPS switching causes sudden changes in current flow through parasitic inductance of the circuit that, in turn, induces a voltage. The magnitude of the induced voltage follows the relationship

$$V(t) = L\frac{dI(t)}{dt} \ ,$$

where V is voltage, t is time, L is inductance and I is current. Obviously, a sudden change in current, such as the switching from an SMPS, generates a very large multiplier applied to the inductance that in turn generates a very large induced voltage spike.

While electrical circuits contain parasitic inductance, a major contributor to the inductance responsible for inducing voltage spikes on the primary side comes from the required capacitance on the primary side, which is used to quickly provide current when the SMPS switches on. The equivalent series inductance of the resulting capacitor becomes part of the circuit that passes the sudden inrush and sudden stops in current. Figure 1 shows a common electrically equivalent model of a capacitor with its parasitic contributors.

By deducing the direction of current flow immediately following an SMPS switching event, one can determine, or predict, more information about the resulting voltage spike. Consider when an SMPS switches the primary from disconnected to connected, or *on*, as in Fig. 2. While disconnected, the secondary side will drop its potential as energy flows out of the low-pass filter into the load on the secondary side. On the primary side, current from the source powering

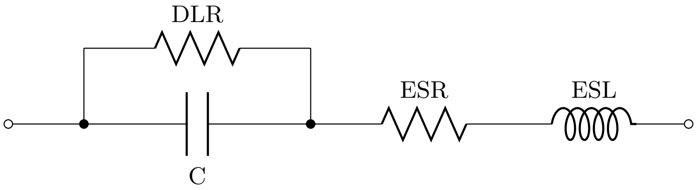

Fig. 1. Electrical model of capacitor where C is the capacitance, DLR is the dielectric leakage resistance, ESR is the equivalent series resistance, ESL is the equivalent series inductance.

the SMPS flows through the ESL into the capacitor. When the connection to the primary side reestablishes itself, the lower potential secondary side causes a sudden draw of current from the primary side that in turn suddenly reverses the flow of current through the ESL. The sudden change in current through the ESL induces a voltage opposite, or against, the voltage applied on the primary side. As such, an observer monitoring voltage upstream of the SMPS on the primary side will measure a sudden drop in voltage followed by a recovery based on the capacity of the capacitor and the capabilities of the source powering the SMPS.

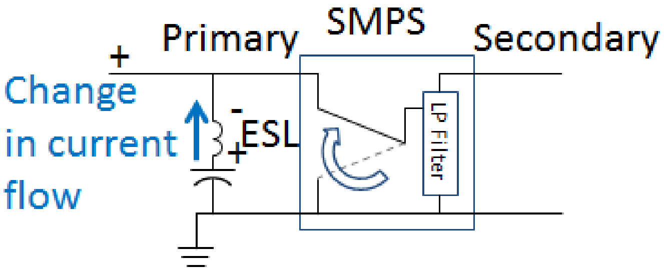

Fig. 2. SMPS with primary connecting.

Similarly, consider when an SMPS switches the primary from connected to disconnected, or *off*, as in Fig. 3. While connected, the secondary side draws current from the capacitor and the source powering the SMPS. When the SMPS disconnects from the primary, the current flowing out of the capacitor and through the ESL suddenly changes direction when all the current from the source feeds, or replenishes, the capacitor. The sudden change in current in turn generates a voltage aligned, or in addition to, the voltage applied on the primary side. As such, an observer monitoring voltage upstream of the SMPS on the primary side will see a sudden increase in voltage followed by a recovery based on the capacity of the capacitor and the source powering the SMPS.

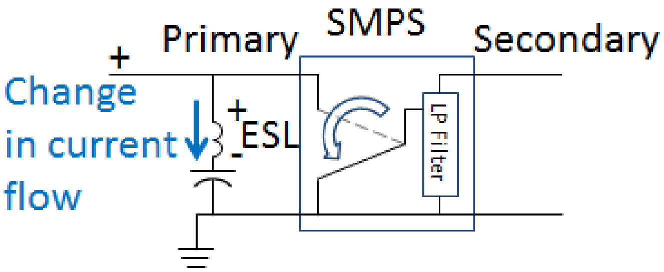

Fig. 3. SMPS with primary disconnecting.

Discriminating between on and off switching events from the primary side of the SMPS reduces to a simple check of voltage polarity swing at the beginning of the voltage spike. Furthermore, the absolute amplitude of the voltage spike and recovery typically differ between on and off switching events for multiple reasons, including input voltage level, load and capacitance. The resulting signal allows for very straightforward triggering logic to capture voltage traces synchronous to an on or off switching event.

Our measurements taken do not employ any strategic tap location or additional conditioning on the power going to the primary side of the SMPS. While some data conditioning can clean the voltage signal to simplify measurements, such modifications entail more intrusive modifications than a simple direct tap at the source for the SMPS. Only minimal signal conditioning was used on the tapped line fed to a signal recorder. In fact, to emphasize the ability to monitor power usage from a distance, all measurements taken come from a tap six feet away from the SMPS and co-located at the output of the power source feeding its primary side.

Figure 4 represents a sample voltage signature of a tapped power source feeding an SMPS. As can be seen at the beginning of each spike, the voltage either suddenly dips or rises, followed by ringing due to the resonance of the circuit feeding the SMPS. The tapped line connects to a high-pass filter before connecting to an oscilloscope to remove the DC bias and any ripple caused by the SMPS. Based on the previous discussion, one can easily identify the first spike as an on event and the second spike as an off event. One can also notice the magnitude difference, allowing an oscilloscope trigger to be set on either an on or off spike.

3 Using Test Vector Leakage Assessment

Test Vector Leakage Assessment (TVLA), proposed by Goodwill et al. [9], is a powerful approach for determining if a device's power consumption relates to processed secret information. In summary, TVLA combines, or magnifies, potentially leaky states of a cryptographic algorithm by calculating specific input

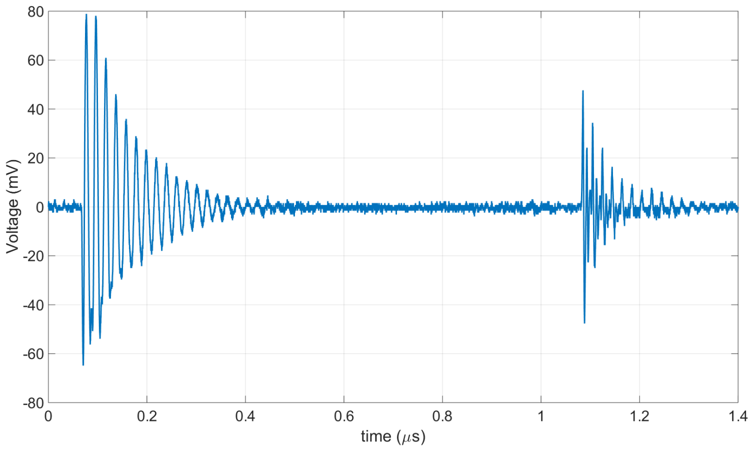

Fig. 4. Voltage trace on primary side of an SMPS showing an on spike followed by an off spike.

vectors and keys. The resulting states create leaks with higher SNR compared to attack vectors meant to isolate sub-keys. As such, when evaluating signals with a loose correlation to the power of a device under test (DUT), the extra sensitivity provided by TVLA allows for efficient and practical assessments of different signal acquisition and processing techniques.

One of the tests in the TVLA methodology is to determine whether there are statistically significant differences in the means of traces collected with a fixed set of data, versus randomly generated data. In applying this, one would take two sets of data, and conduct Welch's t-test to determine whether there is evidence that a null hypothesis that the sets are the same is false. We note that one would typically randomly interleave acquisitions so that environmental effects are the same for both sets and there are no erroneous indications of leakage, caused, for example, by the least significant bit of a variable used to count the number of acquisitions.

Consider two sets of acquisitions, of n_1 and n_2 samples, respectively. We can compute sample means, $\bar{X}_1$ and$\bar{X}_2$, and sample standard deviations, σ_1 and σ_2, respectively. One can then compute a t-statistic using Welch's t-test:

$$\alpha = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{{\sigma_1}^2}{n_1} + \frac{{\sigma_2}^2}{n_2}}} \ , \tag{1}$$

where the result is distributed over a t-distribution with ν degrees of freedom, i.e., $\alpha \sim t(\nu)$. In practice, one would use the asymptotic result where the t-distribution is equivalent to the standard normal distribution, so ν does not need to be defined.

Goodwill et al. use as a baseline $\alpha > 4.5$ to indicate the presence of leakage. Specifically, an $\alpha > 4.5$ gives the probability of indicating leakage where no leakage is present, often referred to as a Type I error, of $\sim 1 \times 10^{-6}$. The probability of no leakage being indicated where leakage is present, often referred to as a Type II error, is not defined. However, repeating experiments can mitigate this problem. To detect side-channel leakage that could be used to conduct an attack, one would typically use this test on a set of power consumption traces where the t-statistic is computed in a point-wise manner. That is, one can test for leakage at each index in a series of acquisitions indicating at what point in time leakage occurs if present.

In our case, large portions of a cryptographic operation can affect a single on–off switching period because of the low-frequency content of the SMPS duty cycle. Indeed, depending on the algorithm and SMPS, the execution of an entire block cipher, or multiple instances of a block cipher, including input and or output data can all affect a single SMPS duty cycle. If one were to compare the power usage between a set of random inputs and a set of fixed inputs, as described by Goodwill et al. [9], power dependencies attributed to the input and output data would add together with other leakages, making it impossible to determine if exploitable leakage is present. The multi-dimensional, but single-point, result of translating power usage on the secondary side of an SMPS to its duty cycle requires the use of targeted vectors, similar to the approach described by Mizaki and Hayashi [15].

To construct a TVLA-like test for leakage detection we would need to target the middle of a cryptographic operation, while having inputs and outputs that are indistinguishable from random values via some side-channel. For example, to evaluate an AES encryption operation one would target a middle round of the block cipher. For that round, the key and input vector maintain a minimal Hamming distance between the round's input and output states as well as maintain a minimal Hamming weight of the round input and output states.

One would generate such a key and input vector by setting the input state of a given middle round to all zeros and propagating the state through one round of operations, excluding the AddRoundKey operation. The result determines the given round key. As such, the state XORed with a round key of the same value provides an output state of all zeros. From this round key one can invert the key schedule to determine the secret key, and then compute the required plaintext by deciphering the round input state of all zeros the required number of rounds. Generating additional inputs require slightly changing the targeted round input state to keep its Hamming weight low, and computing the relevant plaintexts. To generate 2^{16} distinct plaintexts with the required properties, one needs to modify two bytes, and one can try to choose the bytes that will have a minimal impact. For example, by choosing two bytes such that only one column changes in the state matrix when a MixColumns operation is computed. We give a more explicit description of this method in Appendix A.

The resulting set of plaintexts can be randomly interleaved with random plaintexts to provide a good method of detecting leakage using the methods described by Goodwill et al. The test accentuates any leakage caused by Hamming weight or Hamming distance in the implementation of a round of AES. We note that a specific secret key must accompany the plaintexts, so such a test requires the ability to set the key in the DUT. Furthermore, if this version of TVLA reveals a leak, the results do not specifically reveal a particular type of leak, or indicate how to exploit the leakage to determine a secret key. However, by using the same technique, one can generate more specific TVLA vectors, with their associated secret key, to help isolate where and how a leakage manifests.

4 TVLA on a DUT From the Primary Side of its SMPS

The rest of the paper focuses on a specific example and the practical implications and solutions required to quickly detect leakage using TVLA, and exploiting the observed leakage to extract cryptographic keys. We used a SASEBO-GII [18] as our DUT with the on-board FPGA powered by an SMPS. The only side-channel we observed was the primary side of the SMPS. Table 1 provides the specifications of the DUT and associated SMPS. The SMPS sets its output voltage based on a connected resistor R_{SET}. An R_{SET} of 27.4 Ohms was used to set the SMPS output to 1.2 Volts. The SMPS was then configured to mate with the CN1 connector on the SASEBO-GII board, providing power to the FPGA core. The sense resistors R1 and R2 were bypassed, and the resulting voltage between test points TP2 and TP4 was 1.17 Volts. Figure 5 shows the setup of the oscilloscope, computer, power supply, SASEBO-GII and SMPS.

Table 1. DUT and associated SMPS

DUT	
System	SASEBO-GII
FPGA	Xilinx Virtex 5 LX50
Algorithm	AES-128

SMPS Providing Power to FPGA Core	
Manufactuer	Texas Instruments
Part Number	PTH08000W
Input Voltage Range	4.5 Volts – 14 Volts → set to 4.5 Volts Input
Output Voltage Range	0.9 Volts – 5.5 Volts
Switching Frequency	300 kHz
Input Capacitor	100 uF Electrolytic
Set Resistor (R_{SET})	27.4 kOhms Metal Film → 1.2 Volts Output

4.1 Initial TVLA Experiments

The first experiments we conducted used a trigger to determine when an AES encryption operation occurred. For each AES encryption operation, data was

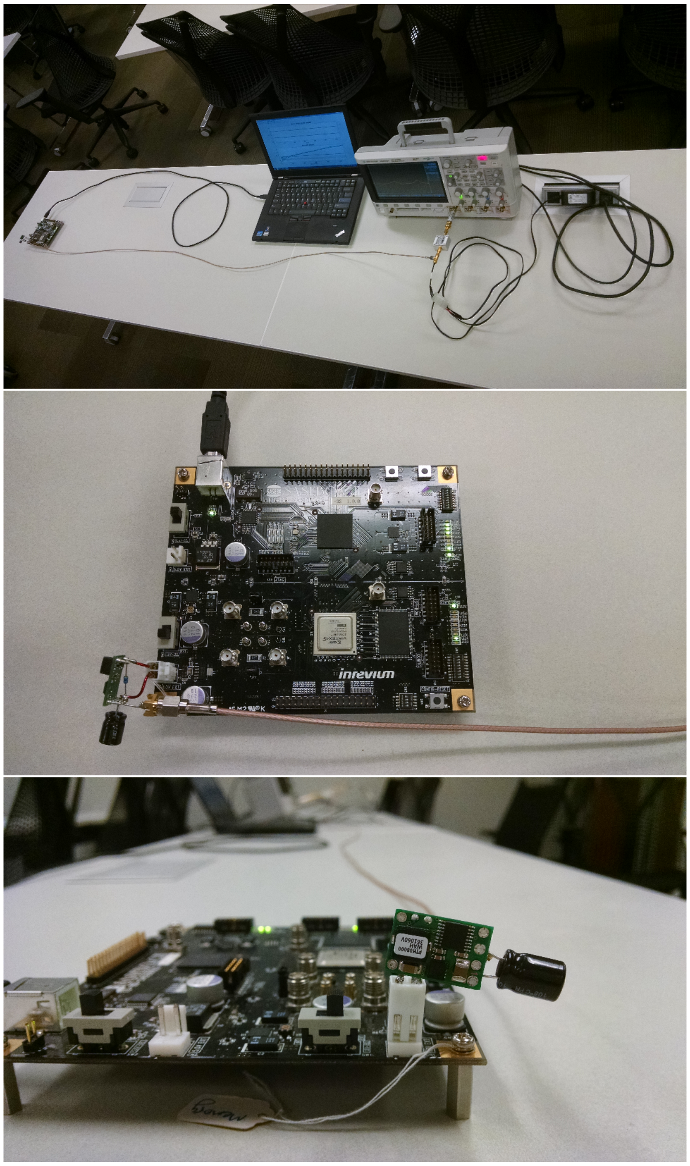

Fig. 5. The setup including the SASEBO-GII, SMPS, computer and oscilloscope (upper), the SASEBO-GII (middle) and how the SMPS connected to the SASEBO-GII (lower).

randomly chosen as either a targeted or random vector, and sent as input to the DUT. Using the trigger, an oscilloscope was used to acquire a voltage trace from the source powering the SMPS. Predictably, the SMPS switching events did not synchronize with the AES encryption execution. However, any load variation on the SMPS secondary side can only affect subsequent duty cycles of the SMPS, so the oscilloscope was set to collect a trace long enough to record multiple duty cycles after the AES encrypt execution. Initially, over 25 cycles were recorded.

Three variables were defined as:

$$\begin{aligned} c &= \text{number of duty cycles directly after trigger} \\ \mathbf{w} &= \text{weight vector of } c \text{ elements} \\ d &= \text{delay after trigger before first on event} \end{aligned}$$

The variables were varied such that c determined the number of duty cycles to process in each trace, $\mathbf{w}$ determined the weight of each duty cycle, and d determined a maximum delay threshold before excluding the trace and associated data. As d increases, the information in the duty cycle decreases. That is, the duty cycle is less coupled to the power draw from the targeted AES encrypt operation.

For each trace, if less than d, a $\mathbf{w}$ weighted sum of c duty cycles provided a single number representing a relative power draw of the load. Each number contributed to one of two sets, representing input data from either the targeted or random vectors. Finally, as explained in Goodwill et al. [9], the two sets generated a single number using Welch's t-test.

Using targeted vectors, described in Section 3, one experiment collected a total of 1.3×10^5 traces comprising approximately 2^{16} targeted inputs and 2^{16} random inputs. Setting $d = 2.08\,\mu$s reduced the targeted and random sets to approximately 4×10^4 traces. Setting $c = 4$ and $\mathbf{w} = [0.83 \;\; 0 \;\; 0.17 \;\; 0]^T$ on the reduced sets, (1) gave a t-statistic of 17.83, yielding a confidence level of $1 - 5.639 \times 10^{-71}$ that the measured duty cycles reflect a change in cryptographically sensitive information within an AES encryption operation.

Variants of the above approach can be considered, such as setting a threshold to determine inclusion for each duty cycle within each trace and weighting appropriately, or setting an early threshold for duty cycles that potentially represent too much activity before an AES encryption operation, etc. However, the approach described in Section 4.2 provided a more substantial increase to the t-statistic comparatively, allowing the transition from leakage detection to practical AES key extraction.

4.2 Subsequent TVLA Experiments

The experiments described in Section 4.1 provided confidence that measuring duty cycles from the primary side of an SMPS can expose leakage from a device on the SMPS secondary side. Subsequent steps involve various approaches to increase the SNR for practical key extraction. The following approach focuses on reducing the time to extract an AES key rather than reducing the number of AES encrypt operations. A subsequent section will discuss an approach that focuses on the latter. In order to strengthen the coupling between the SMPS duty cycle and the power required to encrypt a specific input, an introduced modification in control requested the AES engine to encrypt the same input data in a continuous loop. In addition, an oscilloscope configuration was set to trigger when an SMPS on event occurred, rather than when an AES encrypt executed. Furthermore, rather than measuring the full duty cycle from a recorded voltage trace, the triggered on event was shifted to allow only the off event to be recorded at a high time resolution. Given that this provided a fixed reference in time for the on event, an oscilloscope could generate an average trace on-the-fly to provide an average location of the off event.

Using on-scope averaging mode affords much faster trace collection rates by removing the data transfer time to a computer. The larger the number of traces that are averaged, the greater the efficiency, but at the cost of a lower number of unique inputs. The low frequency information of the SMPS duty cycle translates to convolving high frequency leakage information on the secondary side with temporal noise. As such, on-scope averaging has the potential to provide improvements in increasing a t-statistic in a fixed amount of time.

The first experiments incorporated a low-pass filter in an attempt to stretch the voltage spike oscillation period out greater than the duty cycle jitter. With such a signal, the average voltage trace converges faster due to less interfering voltage oscillations after the initial spike. Figure 6 shows a single low-pass filtered signal. Figure 7 shows the beginning of 100 off events at a higher time resolution along with the mean off event.

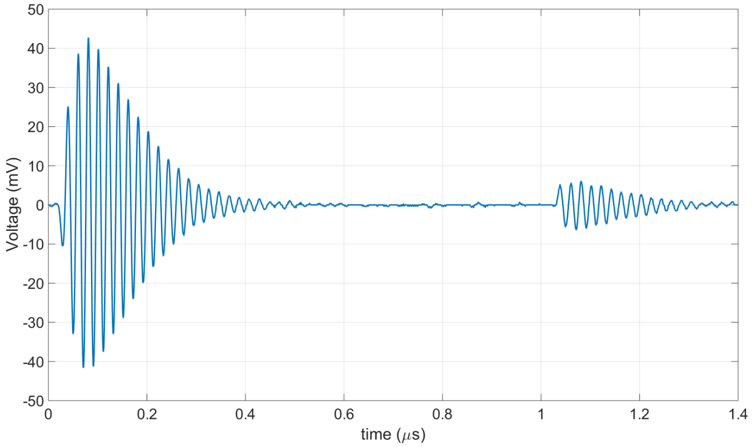

Fig. 6. Low frequency content of Voltage trace on primary side of SMPS.

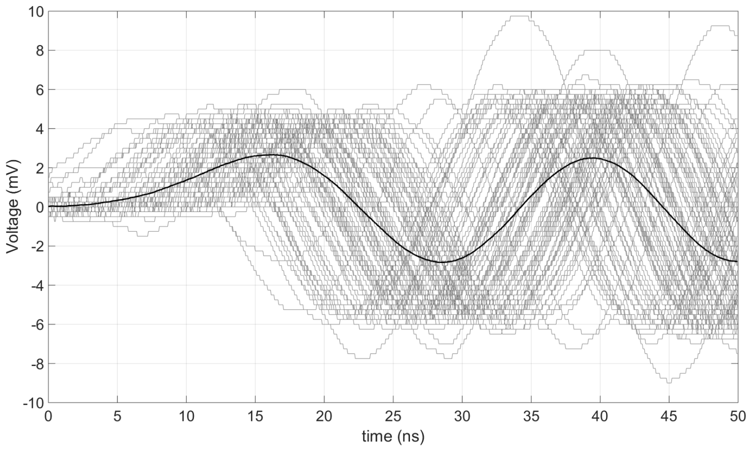

Fig. 7. One hundred off switching events and their mean.

Performing the same TVLA as before, substituting weighted sums of duty cycles with mean off events, yielded a t-statistic of $92.88\,\sigma$. Figure 8 shows the resulting means of the two sets and the resulting t-statistic over time. At this point we had enough confidence to perform an AES key extraction attack.

5 Side-Channel Attack on Primary Side of SMPS

In order to construct a key extraction attack one would typically need to try different targeted TVLA vectors in order to determine the specific intermediate values that leak and their leakage model. However, one may alternatively make an educated guess and attempt an attack sooner. In the case of the SASEBO-GII, the intermediate states that leak correspond to a leakage model that had already been determined, given our knowledge of the FPGA design under attack. As such, the attack proceeded with a correlation power analysis (CPA) [3] on the Hamming distance across the last round of the AES encrypt. However, using the data collection method described in Section 4.2, 185,000 unique inputs were required to extract only one sub-key byte. Therefore, the following additional signal processing approaches were used to further increase the SNR of the duty cycle.

5.1 Wavelet Based Detrending

An observation gained from collected duty cycles over time revealed a large drift. Many factors can cause duty cycle drift, such as temperature, physical stress, and other physical forces that affect the electrical characteristics of the load perceived by the SMPS. As the drift reduces the SNR of the side-channel signal,

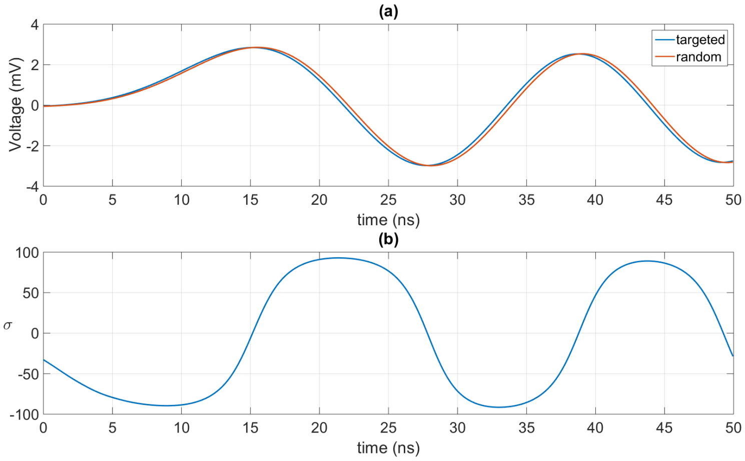

Fig. 8. Plot (a) shows the mean of the two sets of mean off events: targeted and fixed. Plot (b) shows the resulting t-statistic.

we detrended the signal before performing CPA. The detrending approach used the discrete wavelet transform (DWT) [13]. The scaling and wavelet coefficients used were the Daubechies family of wavelets [4]. The reason for using the DWT with the Daubechies wavelets stems from the desire to preserve the relative change in mean duty cycle as much as possible while removing any drift that can change slowly and/or quickly. The Daubechies wavelets provide the maximum number of vanishing moments for a given support, which translates to the flattest frequency response of the filters used in the DWT for a given number of taps [4]. The flatter the frequency response in the band of interest, the more preserved the duty cycles remain when transformed. The fewer taps used in the filters, the faster the DWT can respond to sudden duty cycle drifts. The accuracy of approximation of a signal $f(t)$, represented at a resolution j, or $f_j(t)$, follows

$$\|f(t) - f_j(t)\| \leq C2^{-jp}\|f^{(p)}(t)\| \quad \text{and} \quad \int_{-\infty}^{\infty} f(t) w_{jk}(t) dt \leq C2^{-jp}$$

where C is a constant dependent on the scaling and wavelet functions, and p is the number of vanishing moments of the wavelet function (or number of zeros at π of the DWT approximation filter) [4, 19]. Figure 9 shows the frequency response of the DWT filters when using Daubechies wavelets with $p = 2$ and $p = 8$, or db2 and db8 respectively. A compromise exists because the number of taps is equal to double the value of p. So the better the fit, the less reactive the DWT becomes to sudden changes in duty cycle drift.

The detrending technique follows a simple algorithm, inspired by the wavelet shrinkage techniques as described in [5]. Perform the DWT, or analysis, with a chosen filter order for an L number of levels. Set all of the resulting approximation coefficients to zero, then perform an inverse DWT, or synthesis, on the

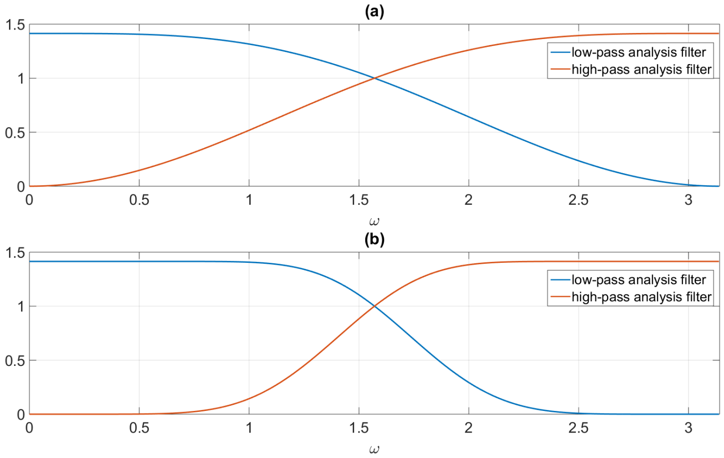

Fig. 9. Plot (a) shows the amplitude of the frequency response of the db2 DWT filters. Plot (b) shows the amplitude of the frequency response of the db8 DWT filters.

remaining detail coefficients. Further enhances could probably be made by performing thresholding, again, akin to what's described in [5], but on the scaling rather than detail coefficients.

After detrending the signal using the above method with various ordered filters at various levels, we were able to extract the last round key with 125,000 unique inputs over 4 days. However, each unique trace comprised of 2^{16} averaged traces. Using the same data without detrending, we could only extract one subkey byte from the last round key. After various key extraction experiments, the db3 filters analyzed at 3 levels provided the best consistent results. In general analyzing $2-4$ levels with various ordered filters worked well. Figure 10 shows the typical duty cycle drift observed before and after detrending.

5.2 Deconvolution

The next approach to increasing the SNR of the measured duty cycles focused on a more accurate PDF of the off events mean. The method follows a straightforward deconvolution by observing that a waveform shifted and added to itself maps to a phase change in the frequency domain. The property comes directly from the linearity and time shifting properties of the Laplace, Fourier and z transforms [1]. For the z transform,

$$\mathcal{Z}\left[\frac{1}{M}\sum_{m=0}^{M-1} f(nT + k_m T)\right] = \frac{1}{M}\sum_{m=0}^{M-1} z^{k_m}\mathcal{Z}\left[f(nT)\right] \ ,$$

where $f(nT)$ is a single off event waveform, T is the sampling period, M is the number of off events averaged, and $\mathbf{k} = [k_0 \ \ k_1 \ \ \cdots \ \ k_{M-1}]^T$ is a vector of samples that the m^{th} off event is shifted relative to $f(nT)$.

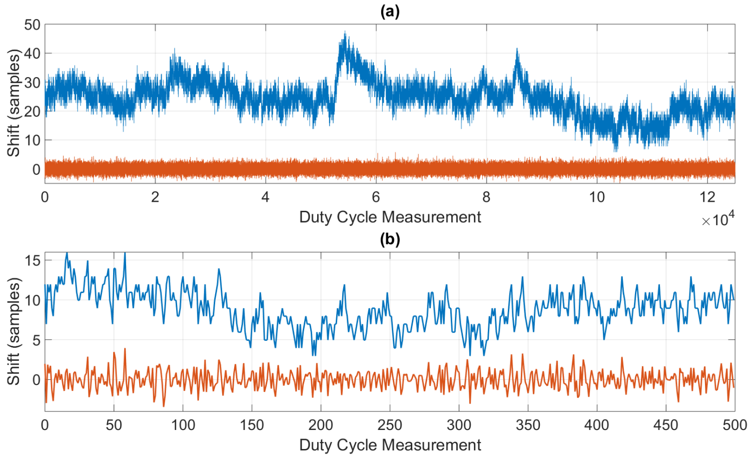

Fig. 10. Plot (a) shows long-term detrending. Plot (b) shows short-term detrending.

The deconvolution technique follows a simple series of steps. First, align a collection of single off event waveforms to generate a low-noise template $f(nT)$. Take a series of single off event waveforms and average them without alignment. Divide the frequency response of the averaged non-aligned waveforms by the frequency response of the template, and calculate the group delay of the resulting spectrum. Look for frequencies that consistently and accurately approximate the true delay compared to the template. Finally, perform the same division and group delay calculations on the off event waveform means and detrend. Also, because deconvolution lifts the concern of voltage spike oscillations interfering with itself shifted in time, we removed the low-pass filter added earlier for a more general solution that applies to any type of signal.

By combining the deconvolution and detrending techniques, we were able to extract the last round key with 130,000 unique inputs over 2 days and 20 hours, with 1,000 averaged traces for each unique trace. Using the same data with only deconvolution or detrending yielded one correct sub-key byte from the last round key. No deconvolution or detrending yielded no correct sub-key bytes.

6 Conclusions

In this paper we demonstrated that an attacker can derive information from the duty cycle of the primary side of an SMPS. Moreover, the developed techniques were used to derive information from an integrated measurement, greatly reducing the SNR. As such, the same techniques could also provide effectiveness on other types of signals. For example, by detrending sample points within a trace over multiple traces, one can remove unwanted power uses operating asynchronously to the signal of interest. For signals difficult to align due to amplitude noise, yet changing in frequency due to timing jitter, one can collapse the trace, or portions of the trace, encapsulating a portion of a cryptographic operation within a single point. For algorithms with masking, one can derive methods for collapsing each trace, or portions of each trace, sacrificing SNR for a reduced search space. In many instances, the approach entails sacrificing of SNR for some other gain, then performing techniques outlined in this paper in an attempt to recoup the SNR while maintaining the sought after advantage.

The approach in this paper focused on reducing the time to extract an AES key rather than reducing the number of AES encrypt operations. One approach that focuses on the latter involves performing a single AES encrypt operation for each input data, and allowing multiple different encrypts to perform with each SMPS duty cycle. One would then use the same collection of duty cycles for multiple and different AES encrypt operations, but with different weights that would shift and slide in time at the same rate. Using techniques such as principal component analysis (PCA), one could then find optimal weighting functions to maximize the information of each duty cycle for multiple inputs.

References

1. Antoniou, A.: Digital Signal Processing: signals systems and filters. McGraw-Hill (2006)
2. Balasch, J., Gierlichs, B., Verdult, R., Batina, L., Verbauwhede, I.: Power analysis of Atmel CryptoMemory – recovering secret keys from secure EEPROMS. In: Dunkelman, O. (ed.) CT-RSA 2012. LNCS, vol. 7178. Springer
3. Brier, E., Clavier, C., Olivier, F.: Correlation power analysis with a leakage model. In: Joye, M., Quisquater, J.J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 16–29. Springer (2004)
4. Daubechies, I.: Ten Lectures on Wavelets. CBMS-NSF Regional Conference Series in Applied Mathematics, Society for Industrial and Applied Mathematics (1992)
5. Donoho, D.L., Johnstone, I.M.: Ideal spatial adaptation by wavelet shrinkage. Biometrika 81(3), 425 – 455 (1994)
6. Eisenbarth, T., Kasper, T., Moradi, A., Paar, C., Salmasizadeh, M., Shalmani, M.M.: On the power of power analysis in the real world: A complete break of the KeeLoq code hopping scheme. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157. Springer
7. Gandolfi, K., Mourtel, C., Olivier, F.: Electromagnetic analysis: Concrete results. In: Koç, C.K., Naccache, D., Paar, C. (eds.) CHES 2001. LNCS, vol. 2162, pp. 251–261. Springer (2001)
8. Genkin, D., Pipman, I., Tromer, E.: Get your hands off my laptop: Physical side-channel key-extraction attacks on pcs. In: Batina, L., Robshaw, M. (eds.) CHES 2014. LNCS, vol. 8731, pp. 242–260. Springer (2014)
9. Goodwill, G., Jun, B., Jaffe, J., Rohatgi, P.: A testing methodology for side-channel resistance validation (September 2011)
10. Kasper, M., Kasper, T., Moradi, A., Paar, C.: Breaking KeeLoq in a flash: On extracting keys at lightning speed. In: Preneel, B. (ed.) AFRICACRYPT. Springer
11. Kocher, P.: Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In: Koblitz, N. (ed.) CRYPTO '96. LNCS, vol. 1109, pp. 104–113. Springer (1996)

12. Kocher, P., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M.J. (ed.) CRYPTO '99. LNCS, vol. 1666, pp. 388–397. Springer (1999)
13. Mallat, S.G.: A theory for multiresolution signal decomposition : the wavelet representation. IEEE Transactions on Pattern Analysis and Machine Intelligence 11(7), 674 – 693 (July 1989)
14. Mangard, S., Oswald, E., Popp, T.: Power Analysis Attacks — Revealing the Secrets of Smart Cards. Springer (2007)
15. Mizuki, T., Hayashi, Y.: AES cipher keys suitable for efficient side-channel vulnerability evaluation. Cryptology ePrint Archive, Report 2014/770, `http://eprint.iacr.org/`
16. Moradi, A., Barenghi, A., Kasper, T., Paar, C.: On the vulnerability of FPGA bitstream encryption against power analysis attacks: extracting keys from Xilinx Virtex-II FPGAs. In: CCS. pp. 111–124 (2011)
17. Quisquater, J.J., Samyde, D.: Electromagnetic analysis (EMA): Measures and counter-measures for smart cards. In: Attali, I., Jensen, T.P. (eds.) E-smart 2001. LNCS, vol. 2140, pp. 200–210. Springer (2001)
18. Research Center for Information Security: Side-channel attack standard evaluation board (sasebo). `http://www.risec.aist.go.jp/project/sasebo/` (2002), online; accessed January-2015
19. Strang, G., Fix, G.J.: An Analysis of the Finite Element Method. Automatic Computation, Prentice-Hall, Inc., Englewood Cliffs, N.J. (1973)
20. Uno, H., Endo, S., Hayashi, Y., Homma, N., Aoki, T.: Chosen-message electromagnetic analysis against cryptographic software on embedded OS. In: EMC '14. pp. 314–317. IEEE (2014)

A Building Targeted Plaintext Vector and Secret Key

In this section we give a more explicit description of the method we detail in Section 3 to derive a plaintext vector and secret key. In the attack described in this paper the leakage from a device during the computation of one, or more, instances of a block cipher is summed to a single value. The Test Vector Leakage Assessment (TVLA), proposed by Goodwill et al. [9], uses a t-test to determine if a fixed input is distinguishable from a random input, as described in Section 3.

For a block cipher, one would want to construct test vectors where, at a chosen point in the computation, there is a minimal Hamming weight and a minimal Hamming distance from some previous state. These correspond to the leakage models typically observed in microprocessors [3], and will maximize the observed leakage. The chosen vectors, and the resulting ciphertexts, should also be indistinguishable from random values to avoid erroneous leakage detection.

We consider 128-bit AES as shown in Fig. 11 where a 128-bit plaintext $P = (p_0, p_1, \ldots, p_{15})$ is used to compute a ciphertext $C = (c_0, c_1, \ldots, c_{15})$. These are typically expressed as matrices:

$$P = \begin{pmatrix} p_0 & p_4 & p_8 & p_{12} \\ p_1 & p_5 & p_9 & p_{13} \\ p_2 & p_6 & p_{10} & p_{14} \\ p_3 & p_7 & p_{11} & p_{15} \end{pmatrix} \quad \text{and} \quad C = \begin{pmatrix} c_0 & c_4 & c_8 & c_{12} \\ c_1 & c_5 & c_9 & c_{13} \\ c_2 & c_6 & c_{10} & c_{14} \\ c_3 & c_7 & c_{11} & c_{15} \end{pmatrix} .$$

The plaintext is initially XORed with a first round key k_0, equivalent to the secret key, and then a round function R is applied iteratively with a sequence of round keys. Each function R takes a state matrix, conducts a series of round operations (ShiftRows, ByteSub and MixColumns operations) and XORs the result with a round key (AddRoundKey operation).

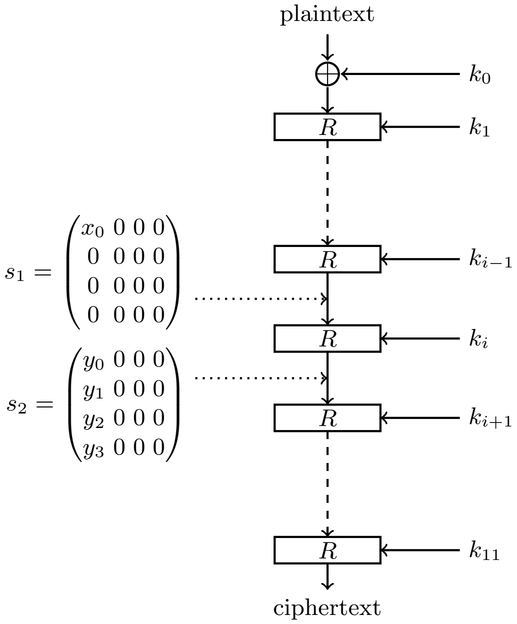
plaintext
k_0
R
k_1
$s_1 = \begin{pmatrix} x_0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$
R
k_{i-1}
R
k_i
$s_2 = \begin{pmatrix} y_0 & 0 & 0 & 0 \\ y_1 & 0 & 0 & 0 \\ y_2 & 0 & 0 & 0 \\ y_3 & 0 & 0 & 0 \end{pmatrix}$
R
k_{i+1}
R
k_{11}
ciphertext

Fig. 11. The structure of AES, and the states chosen to determine plaintext vectors and a secret key for leakage detection.

In order to generate plaintext vectors and a secret key that can be used to detect leakage in the middle of a block cipher, one would proceed as follows:

1. Select one of the middle rounds, denoted i.
2. Choose a state s_1, see Fig. 11, where $x_0 = 0$.
3. Determine a k_i, such that the output of the next round s_2 is also equal to s_1.
4. Compute k_0, the secret key, from k_i, given that the key schedule of AES is invertible.
5. Given k_0 and s_1, decipher s_1 the required number of rounds to determine the plaintext P.
6. Note the plaintext P as the first test vector.
7. Generate remaining entries in the test vector set as follows:
   - Choose a state s_1, where $x_0 \in \{1, \ldots, 255\}$.

- Given k_0 and s_1, decipher s_1 the required number of rounds to determine the plaintext P.
- Note the plaintext P as another test vector.

The above would allow for 256 distinct plaintext elements to be generated. In the work described in this paper, two bytes of s_1 are varied to allow 2^{16} distinct plaintext elements to be generated.