# On-Device Power Analysis Across Hardware Security Domains

## Stop Hitting Yourself.

Colin O'Flynn, Alex Dewar

Dalhousie University

**Abstract.** Side-channel power analysis is a powerful method of breaking secure cryptographic algorithms, but typically power analysis is considered to require specialized measurement equipment on or near the device. Assuming an attacker first gained the ability to run code on the unsecure side of a device, they could trigger encryptions and use the on-board ADC to capture power traces of that hardware encryption engine.

This is demonstrated on a SAML11 which contains a M23 core with a TrustZone-M implementation as the hardware security barrier. This attack requires $160 \times 10^6$ traces, or approximately 5 GByte of data. This attack does not use any external measurement equipment, entirely performing the power analysis using the ADC on-board the microcontroller under attack. The attack is demonstrated to work both from the non-secure and secure environment on the chip, being a demonstration of a cross-domain power analysis attack.

To understand the effect of noise and sample rate reduction, an attack is mounted on the SAML11 hardware AES peripheral using classic external equipment, and results are compared for various sample rates and hardware setups. A discussion on how users of this device can help prevent such remote attacks is also presented, along with metrics that can be used in evaluating other devices. Complete copies of all recorded power traces and scripts used by the authors are publicly presented.

**Keywords:** Side channel analysis, TrustZone, cross-domain attacks

## 1 Introduction

One popular method of preventing security attacks on embedded devices is to have a Trusted Execution Environment (TEE), where only well-validated code executes, and only code in the TEE is allowed to access sensitive resources such as encryption keys or cryptographic accelerators. There are many implementations of this idea – automotive devices typically include a Hardware Security Module (HSM) based on EVITA specification, consumer devices may have a proprietary secure enclaves (such as Apple secure enclave), and implementations of microprocessor vendors such as Arm include general implementations such as Cortex-A TrustZone.

TrustZone for Cortex-M (Armv8-M) contains a variant of TrustZone targetting resource-constrained applications, such as IoT. TrustZone for Cortex-M is unique as it targets the Cortex-M devices, which are currently found in a variety of applications due to a reasonable spread of devices covering low to medium performance with low to medium power consumption and cost. These devices are found extensively in consumer IoT products, but also industrial and automotive products which have similar requirements. While TrustZone-M does not specify cryptographic accelerator requirements, devices in the market implementing TrustZone-M are at minimum including symmetric encryption as

part of TrustZone-M. Considering that TrustZone-M is designed to allow secure boot and secure firmware update, the inclusion of cryptographic primitives is effectively required.

Side-channel power analysis is a well-known method of breaking such cyptographic primitives, first introduced by Kocher et al.[KJJ99]. TrustZone-M does not mandate side-channel power analysis resistance for any included cryptographic cores, so the ability of an attacker to perform side-channel power analysis is reasonably expected on these implementations. This means developers relying on TrustZone-M must not expect that security will be guaranteed when physical access to the device is possible. Future Arm cores look to include such protection, for example the M35P specifically mentions physical-layer protection against these attacks.

These devices often include an ADC, which allows physical sampling of various voltages on the system. By configuring this ADC to perform sampling operations before calling cryptographic operations, an attacker can perform side-channel analysis using entirely on-board resources as introduced by Gnad et al.[GKT19]. This effect will be used to break cryptographic implementations inside of the TrustZone-M secure world from the non-secure world which has access to the ADC configuration. This attack will be referred to as a *cross-domain* attack, as it cannot be applied remotely by itself, but could be used to break hardware security domain barriers that would otherwise preven the remote attack from extracting secrets.

The paper will begin in Section 1.2 with an introduction to TrustZone-M as a TEE, with specifics of the TrustZone-M Core type described in Section 1.3, and the specifics of the microcontroller used in this example in Section 1.4. An example application to better frame TrustZone-M is presented in Section 1.5. We then describe in Section 1.6 previous work on remote side-channel attacks and remote side-channel power analysis attacks.

From there in Section 2 will perform a basic CPA attack on the AES hardware accelerator to build a baseline performance measurement. The baseline measurement will be manipulated to explore the effect of reduced sample rate (Section 2.1) and ADC bit depth (Section 2.2) on the success of the attack.

An attack using internal resources will then be presented in Section 3, culminating with the results of an attack on the SAML11 evaluation board in Section 4. This attack requires no special physical modification of the board, and demonstrates how an attacker could perform an attack against the AES accelerator by running code from the non-secure world. We then discussion countermeasures in Section 5 before concluding in Section 6.

## 1.1 Contributions

This work contains the following contributions:

1. The first side-channel attack on a device with TrustZone-M[1], performed using an on-board ADC controlled from the non-secure world to capture power measurements to recover secrets processed in secure world of the TrustZone-M, making the attack possible to perform without having physical access to the device power rail.

2. Comparison of effect of sampling rates considerably lower than the target device clock rate for side-channel power analysis of symmetric algorithms.

3. Discussion on use of security features within the M23/M33 architecture to reduce the ability of a remote attacker to perform these attacks.

4. Presentation of an architecture for extremely fast trace recording from the on-board ADC to simplify further experimentation on Arm devices.

[1]Note this claim is specific to the Cortex-M TrustZone, which is separate from the existing TrustZone for Cortex-A which has a number of published attacks.

The authors have also made available the entire toolchain as a public project at https://github.com/colinoflynn/xdomain-dpa-m23, including the example firmware, scripts and code used for processing recorded data, and also over $500 \times 10^6$ recorded power traces from the target devices. These power traces include several modes (such as random key and random plaintext) that were not used by the authors themselves in this paper, but may be useful for future research such as training neural networks. All results presented in this paper can be recreated with provided code, scripts, and raw data.

## 1.2 Trust Zone for Armv8-M (TrustZone-M)

The TrustZone architecture is specifically designed to ensure an attacker that obtains code execution privileges on a device cannot perform a complete compromise. As devices become more highly interconnected, considerable trust is placed into the security of the TrustZone architecture. While many devices are not designed to run untrusted code, software exploits may allow an attacker to gain remote code execution. Assuming the design had correctly partitioned all of the network and I/O functions into non-secure space, such execution will occur in the non-secure space, thus fundamentally preventing read-out of secure secrets. In addition as even certain peripherals and I/O pins can be only controlled from the secure code, physical security of connected devices can also be protected.

TrustZone was initially introduced as a component of the larger Cortex-A devices, with the objective of providing a hardware barrier between untrusted code that could include malicious components, and critical functions that include cryptographic secrets.

As Cortex-M devices are typically used in environments with constrained power, cost, and real-time requirements, TrustZone for Armv8-M (TrustZone-M) does not use the same low-level architecture as TrustZone for Cortex-A (TrustZone-A). The higher-level setup is similar: an application is partitioned into a "secure" and "non-secure" world[2], where certain memory segments and peripherals can only be accessed from a secure application. This prevents an application exploit from reading out memory from the secure section of the device. Other features include abilities to mark memory as execute-only (cannot be read out) or no-execute (only used for data storage). Some features differ to allow better power and real-time performance, such as allowing non-secure interrupts to be serviced while running secure code. A full overview is provided by Arm in [Lim17].

With TrustZone-M, the CPU registers are shared between secure and non-secure states, with the exception of the stack pointer (R13) which has separate secure and non-secure copies. In addition protection such as stack-pointer limits are designed to detect and catch stack overflow and other potential attacks in both non-secure and secure states. Various peripherals can be configured for the secure or non-secure code access, allowing a user to determine what should be considered as a sensitive peripheral.

## 1.3 M23/M33 Core

The Cortex-M23 and Cortex-M33 cores are new cores targetting small applications where security is a key requirement. These cores are described as being the first devices with TrustZone for Armv8-M (TrustZone-M). The Cortex M23/M33 cores are one of the most recent cores available in released silicon. The M23 core was first available in the Microchip SAML11 (used in this paper) released in July 2018, and then in the Nuvoton M2351 released in September 2018.

The first hardware available for the M33 core is the Nordic Semiconductor nRF9160 which contains a LTE cellular modem, and was released December 2018. For stand-alone microcontrollers, the M33 core is available in the NXP LPC55S6x which was available in the market as of March 2019, and the STM32L5 which is not yet available. Based on

[2]The notation of "secure" and "non-secure" is used in this paper instead of Trusted and Untrusted execution environments, as the M23/M33 documentation follows this notation.

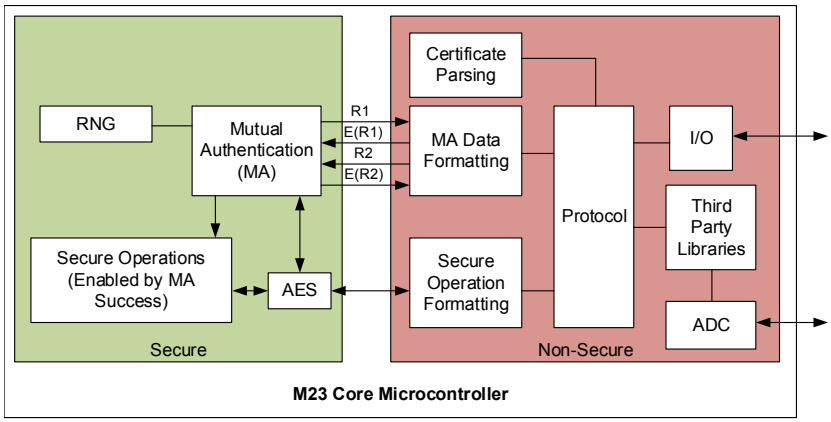


**Figure 1:** TrustZone-M allows development of applications where a break in application-level security cannot reveal secrets held by the secure code.

pricing and marketing information these devices are likely to be found in various IoT and similar products, but the authors are unaware of their use in commercial products yet.

## 1.4 SAML11 and TrustZone-M

The SAML11 configures a peripheral as secure or non-secure via the `NONSEC{A,B,C}` registers, the ADC in particular being set via the `NOSECC` register. Setting a bit to '1' configures the peripheral as non-secure, setting a bit to '0' sets it to secure. The default value for NONSEC is loaded from part of the user flash configuration ('NVM User Row'), which by default has all peripherals as secure until changed by the designer.

This selection of secure vs. non-secure code and peripheral sets requires knowledge of possible attacks and threat models. Functions should be in the non-secure space by default, as any code inside the secure space could be a possible attack vector to bypass the security mechanisms. In many applications it would be expected the ADC will be part of non-secure code, since typical ADC usage would be associated with user-level information. This would mean that the ADC itself is then likely moved to non-secure space. The SAML11 datasheet even uses the ADC as an example non-secure peripheral when discussing common configuration examples[3].

A designer wishing to move the ADC into secure code space would need to ensure all code accessing the peripheral is moved into the secure code-space. If only a basic stub is moved into the secure code-space and it still allows for relatively arbitrary configurations of the ADC from non-secure space, the attack described in this paper would not be impeded. Moving complex functions into secure code-space would also be undesirable, as these complex functions that may interact with a user present higher risks of code flaws that give an attacker access to code execution within the secure code space.

Certain applications may specifically allow user code to be executed on the non-secure application. This could include devices which are user extensible, such as digital wallets that are designed to allow users to add processing routines for new digital currencies.

## 1.5 Example Application

To help frame the work in this paper, an example application making use of the Armv8-M TrustZone is shown in Figure 1. An example application is presented to understand the

[3]Section 13.2.5.1 of DS60001513B says: 'Below is a typical configuration examples where all peripherals except the ADC, TC0, and Event System (EVSYS) are reserved to the Secure application'

attack model, as to the authors knowledge no publicly available end products use the M23/M33 devices[4], and the authors based this example on published Armv8-M TrustZone use-cases. The ASSURED framework for secure IoT firmware updates presented by Asokan et al.[ANR+18] uses the TrustZone-M via the Arm FPGA prototyping platform, and the design assumptions for ASSURED include that TrustZone-M provides *Protection of Code & Secret Keys on Device*. This assumption is thus reasonable to include in our demonstration application.

This application has an I/O interface, where one of the capabilities of this device is to perform mutual authentication requests. In this example the authentication request comes over an I/O interface, and is formatted and processed in the non-secure code. This would be typical since I/O processing is often a source of attacks such as buffer overflows or reading beyond the end of an array. The non-secure code passes the random number to be encrypted into the secure code, which performs the encryption as part of the mutual authentication. The non-secure code then sends the response to an external user, along with the secure codes random number to be encrypted. Note the secure to non-secure interfaces are designed to minimize complex information exchange, in this case variables are fixed-length for example to reduce the likelihood of introducing complex parsing errors.

This mutual authentication example would allow secure code inside the Armv8-M TrustZone to authenticate another device, without needing to fully trust the firmware operating on the non-secure code. In this example some portions of the protocol are encrypted, and this encryption is also done inside the secure code. Commonly exploited code such as complicated certificate parsing and third-party libraries (which may only be delivered in binary form) are kept strictly in the non-secure code. An attacker with access to the non-secure code could not recover the encryption key used for mutual authentication, nor could they bypass the authentication logic inside the secure code, or bypass the encryption on the secure commands.

## 1.6 Remote and Cross-Domain Side-Channel Attacks

As this work concentrates on TrustZone-M as the hardware security barrier, we will first discuss attacks specific to TrustZone before more general remote side-channel attacks. This paper will use the phrase *cross-domain* attack where a remote attack is *possible*, but the attack itself requires an existing remote attack to be performed. This means a cross-domain attack requires code execution on a device or in a domain which the attacker is not normally expected to have access.

### 1.6.1 TrustZone-A Side Channel Attacks

Previous work on TrustZone has been demonstrated that side-channel power analysis of TrustZone-A cryptographic primitives is possible[BLLB+18], as TrustZone makes no requirements on side-channel power analysis prevention.

Remote side-channel attacks were first presented in general using cache-timing effects by Bernstein[Ber05]. This was a true remote attack as it could be applied across a network interface (i.e., no code execution assumptions). The fertile work in cache-timing attacks was primarily applied to x86 systems, with attacks on Arm-based systems first extensively analyzed by Lipp et al. [LGS+16]. The work by Lipp et al. [LGS+16] did not however concentrate on TrustZone-A specifically, with the first concrete example against TrustZone-A being an attack to recover encryption keys from an OpenSSL implementation of AES using T-Tables by Zhang et al. [ZSS+16, ZSS+18]. Demonstrations against a specific vendor-supplied implementation was first shown by Lapid et al. with an attack on Arm assembly implementations of AES-256/GCM present in a Samsung Keymaster

[4] As of April 2019.

Trustlet[LW18]. More general exploration of cache attacks on TrustZone-A platforms was made possible with the release of cachegrab by Keegan Ryan[NCC18].

Devices based on the M23/M33 core normally do not contain a data cache. When a cache is present it will typically be only an instruction cache (ICACHE) to improve access speed to the flash memory. Side-channel cache attacks based only on instruction caches are more limited, and have been demonstrated to be useful in attacking DSA by Acıiçmez et al.[ABG10], but the previous work on TrustZone-A has assumed a more complete cache implementation.

Some implementations of the M23/M33 cache are not specifically an ICACHE, but instead a cache on *accesses to the flash memory*. This distinction is important as data accesses to flash memory may have exploitable cache-like effects as described by Gallais et al.[GKT11]. This cache is part of the low-level architecture and cannot be disabled (such as the 'ART Accelerator' on the STM32F2 series). To the authors knowledge this remains an open area of research, as no full attacks have been demonstrated against this feature.

Remote attacks on TrustZone-A cryptographic functions have also been performed using fault injection techniques. The RowHammer technique[KDK+14] has been demonstrated to be usable on TrustZone-A attacks as shown by Carru[Car17]. Recently the CLKscrew attack by Tang et al. was able to perform extensive attacks on a TrustZone-A implementation including using cross-domain fault attacks to extract cryptographic keys[TSS17].

### 1.6.2 Cross-Domain Power Analysis Attacks

The Cortex-M devices may lack a full cache making cache-timing attacks inapplicable. For these devices cross-domain power analysis attacks present a worthwhile attack vector.

Cross-domain power analysis attacks against FPGAs have been presented, where the assumed environment is one in that an attacker is able to reconfigure part of the FPGA fabric. Cross-domain power-analysis type attacks were first demonstrated on an FPGA against AES by Schellenberg et al.[SGMT18b], by measuring the delay through a series of buffers. The delay varies with voltage, allowing localized measurement of voltage within the FPGA without relying on specific ADC circuitry. Ramesh et al.[RPD+18] also demonstrated a similar cross-domain attack against AES on an FPGA. Zhao et al.[ZS18] demonstrated a cross-domain power analysis attack using a ring oscillator as the sensor instead of a delay line, and performed SPA against RSA running in a Arm co-processor located in the FPGA.

Moving from an attacker residing on the same FPGA to an attacker residing on a separate device located on the board was also performed by Schellenberg et al.[SGMT18a]. This attack used one FPGA as a monitoring platform to monitor a power rail of another FPGA sharing the same power rail. Schellenberg et al.[SGMT18a] attacked both AES and RSA using this platform. Considering that a board may consist of both more secure and less secure devices, a remote attack which does not even require access to the exact device performing the sensitive operation is particularly powerful. While a FPGA was used for the demonstration by Schellenberg et al.[SGMT18a], fundamentally other devices could be used for the power measurement.

Schmidt et al.[SPK+10] demonstrated that the I/O pins of a device couple sufficient information to allow side-channel power measurements to be performed on a device I/O pin. This attack does not require modifications to the target board, and as many I/O pins may be routed to higher-level connectors of a device, it allows attacks to be performed without opening a device enclosure.

The question of band-limited and noisy measurements being successful in attacking symmetric algorithms was demonstrated by Saab et al.[SLT16]. The work of Sabb et al. is highly relevant to our work, as Saab et al. perform an attack against AES on the supply side of a switch-mode power supply.

A quasi-remote power analysis attack against AES was presented by Camurati et al.[CPM+18], where the attack is performed using sampling of an emitted radio signal. This attack still required physical equipment within 10 metres of the device, but demonstrates an attack which does not require the same level of physical access as classic power analysis attacks. This was attacking an Arm mbedTLS implementation – while still a software implementation, it does demonstrate the leakage is sufficient to perform DPA-style attacks at large distances.

Recent work has shown that an ADC on-board a microcontroller [GKT19] configured to measure an I/O pin can perform side-channel power analysis on-board a microcontroller. This work attacked a software AES implementation, which lays the groundwork for our attack by demonstrating such on-board attacks are possible. We extend this to include more details on the specific effect of variable sample rate and bit depth changes, as well as attacking a state of the art TEE with a hardware seurity implementation.

Asymmetric algorithms with simple power analysis vulnerabilities can be attacked with very band-limited signals. These attacks have also been demonstrated to be possible using less than 100 kHz of bandwidth using radio setups by Genkin et al.[GPPT15], and attacks can even be performed that use acoustic side-channel analysis shown by Genkin et al.[GST14].

### 1.6.3 Practical Applicability

The cross-domain power analysis attack may suggest that the primary goal is to find a remote attack, and then perform the cross-domain attack. Due to the large data traces recorded this may remain impractical. But these cross-domain attacks are useful where physical access is available but limited. This would include for example where attacks are to be 'commercialized', such as developing a commercialized attack that disables an engine control unit (ECU) security in a car to allow tuning or key cloning. If the attack can be performed using on-board peripherals of the device, it simplifies packaging of the attack since it requires only a debugger access which may already be available for other purposes, and no special side-channel power analysis setup or equipment.

## 1.7 Partial Guessing Entropy

The majority of results in this paper will be presented with a partial guessing entropy (PGE)[Mas94], based on magnitude of the correlation output. PGE provides a clear metric around the question of number of traces required to break an implementation, and in particular it allows comparison of number of traces to reduce the guessing entropy to some tractable number. As the authors have chosen to publish full data-sets for this paper, external research groups can freely use these traces to calculate metrics that may be more relevant for their specific attack. More details of PGE are provided in Appendix B.

# 2 External Power Analysis of AES Accelerator

The Microchip SAML11 provides a hardware cryptographic accelerator, which is accessible by the secure code in the TrustZone-M. The datasheet describes ROM-code functions which are responsible for use of the hardware accelerator. These functions are located in execute-only memory, and are claimed to prevent "misuse" of the accelerator.

We can use these functions to perform a basic side-channel power analysis attack on the cryptographic accelerator to provide baseline results. The baseline measurements are taken with a ChipWhisperer-Lite, which samples at 29.48 MS/s (which is four times the 7.3728 MHz clock the SAML11 is configured to operate on). These samples are taken synchronous to the SAML11, meaning the jitter is minimized between the sample location

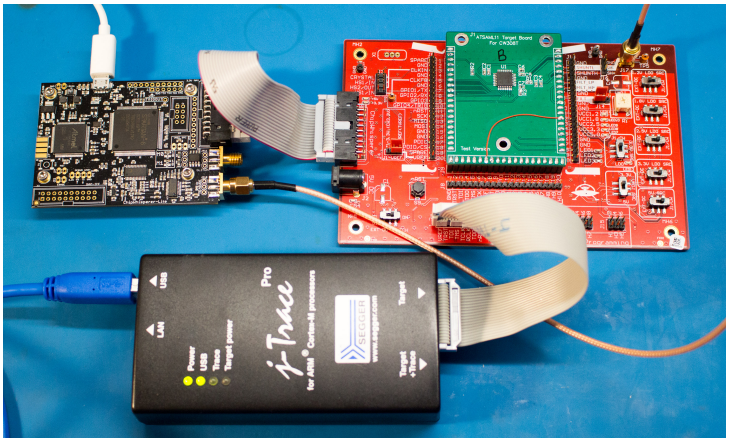


**Figure 2:** Power analysis using a ChipWhisperer-Lite as a sampling apparatus.

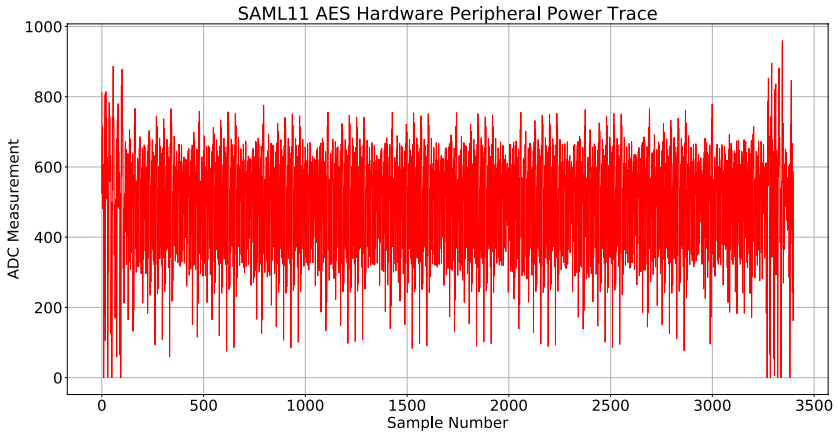


**Figure 3:** Power trace measurement of SAML11 AES operation, sampled at 4x the device clock frequency using a 10-bit ADC in the ChipWhisperer-Lite.

and internal clock of the SAML11. The use of synchronous sampling as a method to demonstrate the attack can be successful even at a low sample rate was first presented by O'Flynn and Chen[OC13].

The power analysis is performed using the ChipWhisperer capture architecture[OC14], and the CPA attacks are performed using Lascar as the analysis engine by San Pedro et al.[SPSG18]. The general setup is shown in Figure 2, which shows the following: (1) the base-board which has the SAML11 target mounted on it, (2) the ChipWhisperer-Lite for external power analysis, and (3) a Segger J-Trace Pro used for JTAG programming.

While the SAML11 contains a hardware accelerator, it is clear from the power trace in Figure 3 that this accelerator appears to consist of only partial acceleration. The entire encryption takes 800 device clock cycles, and has clear round-by-round power signatures.

A Correlation Power Analysis (CPA) attack is performed[BCO04], where the leakage model is assumed to be the Hamming weight (HW) of the S-Box input from the last round of AES-128. This model requires an attacker to have access only to the ciphertext after an encryption occurs. For an output ciphertext byte $c$, the leakage model for a given last-round key guess $g$ is given in (1). Due to the `ShiftRows()` in the last round, the key recovered by this leakage model also needs to be passed through `ShiftRows()` for the correct byte ordering. The original key can then be found from the key scheduling

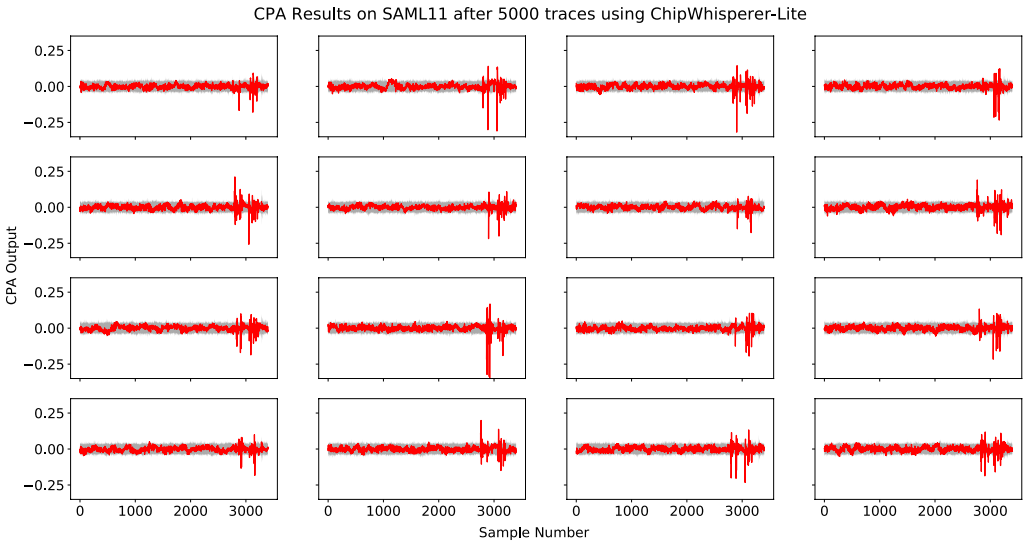


**Figure 4:** CPA attack result of SAML11 AES accelerator.

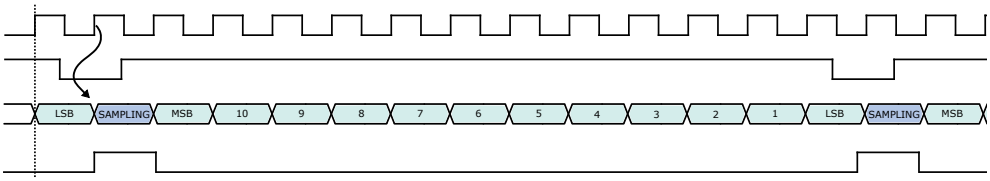


**Figure 5:** The SAML11 ADC acquires a sample every 13 ADC clock cycles.

algorithm.

$$l_g = HW\left(SBox^{-1}\left(c \oplus g\right)\right) \tag{1}$$

This CPA attack shows the entire encryption key can be recovered in 1500 traces, see the top-left graph in Figure 6. This level of success suggests that no side-channel power analysis countermeasures are present in the device. The correlation plots are shown after 5000 traces in Figure 4 for all bytes.

## 2.1 Sample Rate Considerations

Performing the remote attack will take advantage of the ADC within the SAML11. This ADC is assumed to be allowed to operate from the non-secure code. This ADC will remain synchronized to the device clock, but with substantial limitations on the sample rate. The ADC clock itself runs at half the system clock (meaning an immediate 0.5x sample reduction), and when operating the ADC in 12-bit mode, an output sample is generated every 13 clock samples, as shown in Figure 5. This means the sample rate of the ADC is 0.03846x the device clock, or, at our 7.37 MHz clock, the sample rate is 0.284 MS/s.

This result can be replicated with the ChipWhisperer-Lite samples by performing a downsampling operation that keeps only every $n^{th}$ sample. Note this is not done using a decimation operation (which typically would also suggest a low-pass filter is applied to respect the Nyquist limits of the new sample rate), but instead by directly keeping only every 26th sample. The decimation operation would still be using the higher sampling

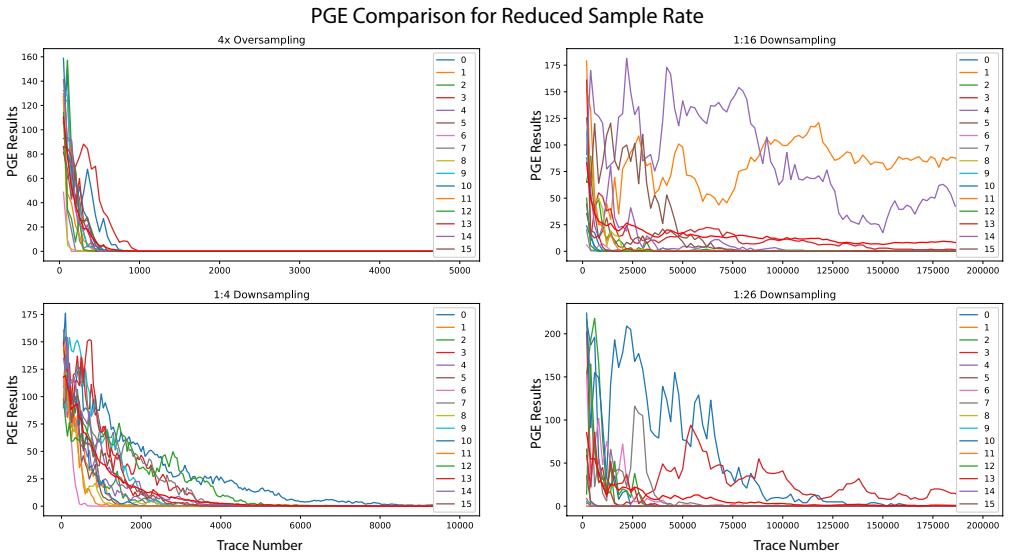


**Figure 6:** Downsampling the external power measurements allows us to compare the effect of very low sample rates on the success of the attack. Note the scales on number of traces differs for these plots.

rate as an input, and it would be expected that some data would be spread between multiple samples. This spreading would also occur with an analog filter[MM12], but for the exact work under question we wish to analyze only the reduction in sample rate and not bandwidth limiting.

As the samples with the ChipWhisperer-Lite were taken at 4x the device clock, achieving the equivalent of the 1:26 decimation that happens with the on-board ADC requires a decimation of 1:104 samples. From Figure 6 it can be seen that severe undersampling does not fully remove information. Some of the bytes become more difficult to recover than others, and interestingly the results from the 1:16 downsampling to the 1:26 downsampling does not represent a significant difference. All results in Figure 6 are downsampling rates relative to the device clock.

## 2.2 ADC Bit Depth

As the cross-domain attack will assume to have a small amount of information present, we also record the effect of a changing bit depth of the ADC on attack results. The number of bits in an ADC has a relationship to the SNR as given by (2).

$$SNR_{dB} = 6.02N + 1.78dB \tag{2}$$

The ChipWhisperer-Lite has a 10-bit ADC, and we can again process the signal to simulate a variety of bit-depths and confirm the effects of our analysis. This assumes that the input signal has been appropriately scaled to the ADC range, such that for example going from a 10-bit ADC to a 8-bit ADC is achieved by removing the lower two bits. In practice this perfect scaling may not be achieved, and for example our 2-bit dataset contained only the values $1, 2, 3$ giving it an effective bit depth of $log_2(3) = 1.58$. Table 1 shows the maximum and minimum values present in a single trace of the various datasets, along with the effective bit-size of the measurement based on the number of digitized steps.

**Table 1:** Imperfect scaling of the input means the actual measurement does not take advantage of the full dynamic range of the ADC.

| ADC Bits | Max Value | Min Value | Effective Bits |
|---|---|---|---|
| 10 | 929 | 429 | 8.97 |
| 8 | 232 | 107 | 6.98 |
| 4 | 14 | 6 | 3.17 |
| 3 | 7 | 3 | 2.32 |
| 2 | 3 | 1 | 1.58 |

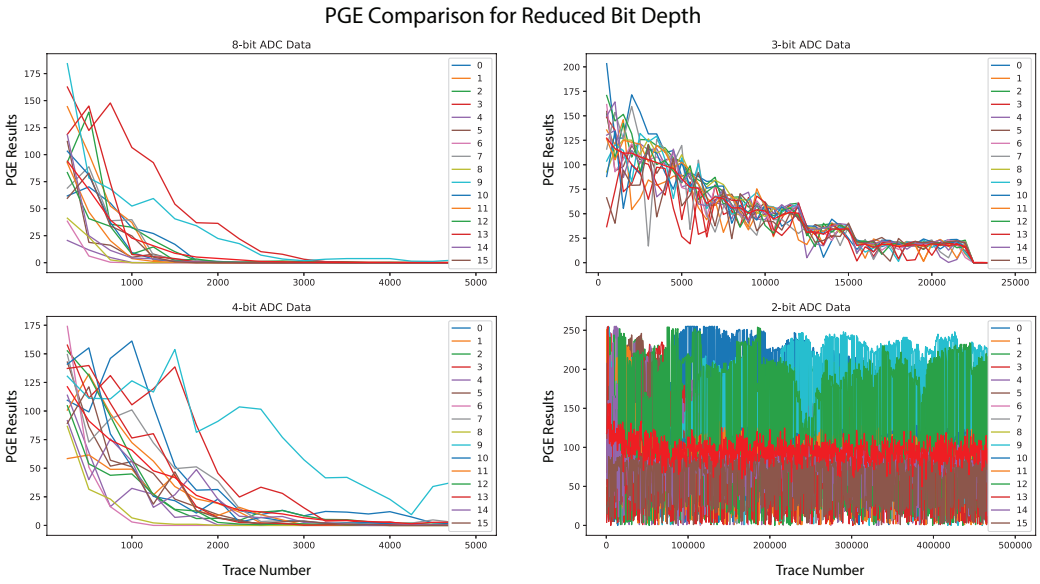


**Figure 7:** Adjusting the bit depth of the externally recorded power traces is done by right-shifting the original 10-bit ADC data, with resulting changes in the guessing entropy.

This effective bit-size shows only the scaling of the data raw, and is not related to the information contained about the leakage.

The results of bit depth on average PGE is shown in Figure 7, where the sampling rate is 7.37 MS/s (1x device clock rate). In this case the attack remains successful down to a 3-bit ADC signal. The 2-bit signal (which has only a 1.58-bit effective depth) does not recover the key within the 500 000 traces recorded.

# 3 Internal Power Analysis on Custom Boards

In the previous section, the effect of changes in sample rate and bit depth on the AES accelerator CPA attack were described. In this section, we will look at how this applies to power analysis performed on-board the device.

Four test boards are used during this first experiment, shown in Figure 9. These boards are labeled (A) which has a shunt resistor and an external amplifier, Board (B) which has a shunt resistor in the V-Core pin but no decoupling capacitor, Board (C) which has a 0-ohm shunt (solder blob short) and partial decoupling cpacitors mounted, and Board (D) which has a 0-ohm shunt (solder blob short) and all the decoupling capacitors mounted. More detail of these are in Appendix A. The internal 16-MHz oscillator is used in the

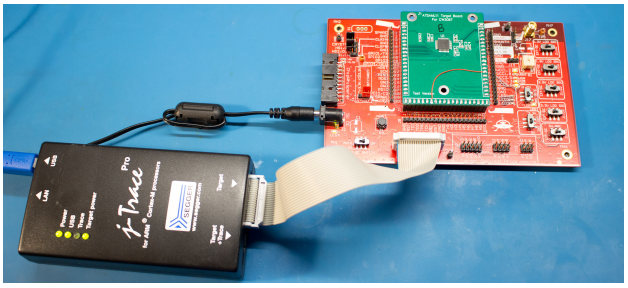

**Figure 8:** Power analysis using a the internal ADC on the test boards.

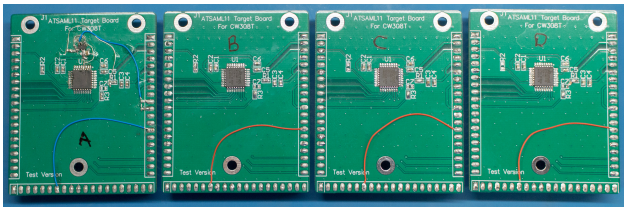


**Figure 9:** Four test boards are marked A through D.

following sections.

The initial tests are performed on Board A with external circuitry to assist with this power analysis. A block diagram of this is shown in Figure 10. An external amplifier provides an amplified and AC-coupled $\text{VDD}_{\text{Core}}$ power measurement that is presented to the on-board ADC. As previously mentioned, this on-board ADC will be sampling at 1/26th the rate of the core clock (also running the AES accelerator). This setup allows us to confirm the attack success rate in a best-case scenario.

The triggering for the ADC capture is based on enabling the ADC before performing a call to the AES encryption function. This precise triggering is assumed to exist where a non-secure application has some ability to cause the secure code to perform an encryption operation. Such an assumption is entirely in-line with the normal usage of TrustZone-M – for example the non-secure code which has the I/O functions would expect to be passing raw received data to be decrypted and processed by the secure code, or would be receiving an encrypted packet for transmission. This was discussed in Section 1.5.

Recreating this experiment on a commercial product would likely require code execution

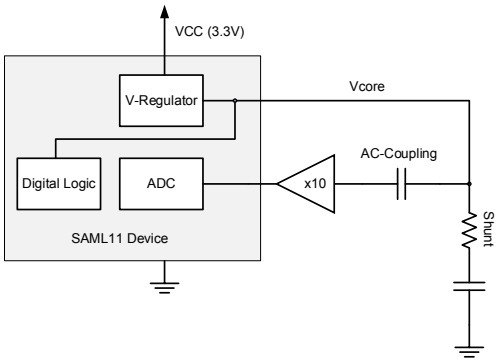


**Figure 10:** The 'A' test board uses an external amplifier to condition the shunt measurement.

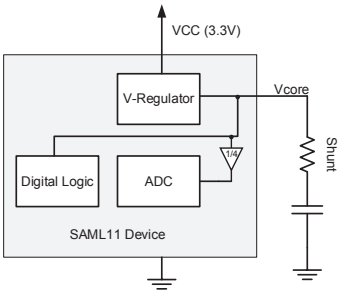


**Figure 11:** The 'B' board has an external shunt resistor to improve the signal strength.

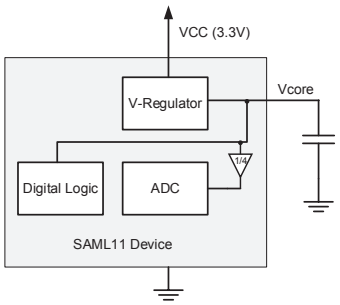


**Figure 12:** The 'D' board does not rely on external modifications compared to regular usage.

permission, but again this permission would only be on the non-secure code. An attacker gaining such access is assumed to not compromise the system based on the security design of TrustZone-M, but these permissions are sufficient to perform power analysis as they could receive encrypted data (to record the text-out needed for the leakage model), along with the ADC sample data.

For Boards B, C, and D there is no connection from the shunt resistor to the ADC input. Instead we take advantage of the fact that one of the internal ADC input options is to measure the voltage $\frac{VDD_{core}}{4}$. We anticipate that sufficient leakage will be present in the lower bits of the 12-bit ADC measurement to allow our power analysis attack to still succeed. The block diagram for Boards B and D is shown in 11 and 12 respectively.

As no external measurement hardware is needed for the remainder of the attacks, a different capture architecture is used. This architecture uses the capability of the Segger J-Link to perform data transfers to internal memory from JTAG or SWD. In this case the entire communications protocol is performed using the Segger J-Link, where trace data that was recorded with the internal ADC is transferred using the Segger J-Link RTT function. This allows us to achieve capture rates of approximately 1100 traces/second. Capturing very large data-sets can be easily achieved in reasonable times at these capture rates.

## 3.1 Results of Board A

We initially used Board A to better understand the impact of the on-board ADC on the power analysis results. The PGE for each of the 16 bytes can be seen in Figure 13, noting it took approximately 200 000 traces to recover the complete encryption key. These results are roughly as expected based on the experiments performed with the ChipWhisperer-Lite and downsampling the signal.

### 3.1.1 TrustZone-M Secure and Non-Secure Code

The majority of results in this paper will perform the ADC and DMA operations entirely from the secure code. This setup simplifies development and configuration by keeping all peripherals in one world, and also mirrors the situation where the ADC driver is located in secure space. This does not however follow the example application from Figure 1.

To validate the use-case where an attacker in the non-secure world must extract secrets from the secure world, we built an application that controls the ADC from non-secure code. To perform an encryption, a call is made using the 'security veneer' which provides explicit interfaces between the non-secure and secure code. These power traces were able to recover the encryption keys, which demonstrated that there is no blackout of the autonomous

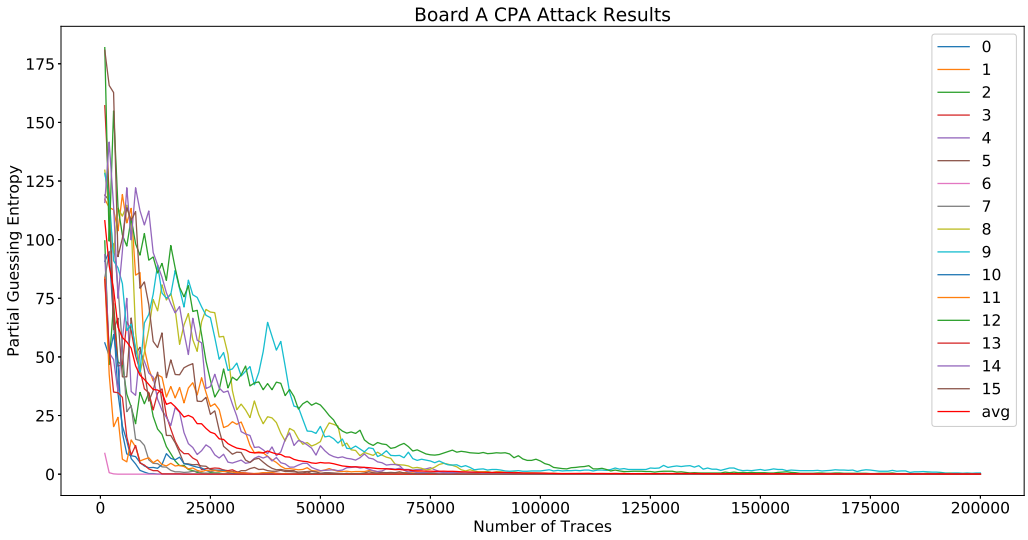


**Figure 13:** PGE Results for each of the 16 key bytes, along with average of all bytes when using an external amplifier to condition the $\text{VDD}_{\text{Core}}$ measurement but sampled using the internal ADC.

ADC or DMA operation (both of which were marked as non-secure peripherals) during a call into the secure world. There is no difference in an attackers capabilities by running from secure or non-secure worlds. The source code for both projects (running in secure world, and running in separate worlds) is available as part of the supporting material.

### 3.1.2 Effect of Phase Offset

From Figure 5, it can be seen that the ADC sampling is done every 26th clock cycle of the core clock. We can modify the location of this sample by delaying the start of the ADC sample operation relative to our call to enable the cryptographic accelerator. Note due to the Arm architecture, precise cycle-level delays cannot be achieved with only `nop` instruction padding[5].

Table 2 show the results of various phase offsets on all bytes. Note that certain offsets appear to favour certain bytes. Our experiments did not demonstrate that any optimization of the offset made practical sense; capturing 200 000 traces for four different offsets (where each offset was selected as optimal for some bytes) did not result in better results than directly capturing 800 000 traces at a fixed offset and using across the bytes.

When not otherwise mentioned, the offset for all results in this paper is equal to delay = 19 cycles. This applies to the earlier Board A results, along with result for the Board B/C/D sections and the development kit. This offset was chosen somewhat arbitrarily, and can be seen from Table 2 represents relatively normal results.

## 3.2 Results of Board B

The Board B results demonstrated a strong signal can still be recovered, as seen in the PGE graph of Figure 14. Board B uses the internal ADC connection, but exaggerates the

[5] `nop` instructions are not guaranteed to be time-consuming, as may be removed from the instruction pipeline by the core. The pipeline state must be initialized with the `isb` before any delay instructions. See reference code for more details

**Table 2:** Average partial guessing entropy of CPA attack after 200 000 traces using Board 'A' for differing cycle offsets from start of ADC capture to start of encryption.

| | Key Byte Targeted | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Offset | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| **5** | 0.1 | 116.2 | 0.1 | 20.0 | 109.8 | 0.0 | 0.0 | 0.0 | 0.0 | 27.5 | 0.0 | 26.0 | 0.1 | 0.0 | 0.1 | 0.0 |
| **6** | 0.0 | 0.4 | 0.0 | 29.9 | 0.0 | 0.2 | 0.1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| **7** | 0.0 | 0.2 | 0.0 | 12.8 | 0.0 | 0.1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| **8** | 0.0 | 0.2 | 0.0 | 17.1 | 0.0 | 0.4 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| **9** | 9.9 | 0.0 | 0.0 | 0.0 | 0.0 | 53.8 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.1 |
| **10** | 61.5 | 0.0 | 10.4 | 30.5 | 0.0 | 40.1 | 0.0 | 0.0 | 0.0 | 32.6 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| **11** | 3.4 | 0.0 | 0.0 | 82.1 | 0.0 | 0.0 | 0.0 | 31.1 | 0.0 | 61.5 | 2.1 | 0.0 | 0.0 | 3.6 | 0.0 | 0.0 |
| **12** | 1.1 | 2.1 | 0.8 | 0.0 | 7.8 | 83.0 | 0.0 | 5.6 | 0.0 | 0.0 | 0.1 | 3.6 | 0.0 | 10.9 | 6.6 | 0.0 |
| **13** | 0.8 | 3.5 | 0.0 | 0.0 | 0.0 | 174.9 | 0.0 | 47.8 | 0.0 | 0.0 | 3.5 | 0.0 | 0.0 | 5.2 | 0.6 | 0.0 |
| **14** | 0.1 | 0.4 | 0.0 | 0.0 | 0.0 | 179.2 | 0.0 | 33.2 | 0.0 | 0.0 | 1.2 | 0.5 | 0.0 | 20.4 | 0.2 | 0.0 |
| **15** | 0.0 | 0.0 | 0.0 | 0.0 | 38.9 | 20.8 | 0.0 | 0.1 | 0.0 | 0.0 | 0.9 | 7.6 | 115.1 | 10.9 | 49.9 | 0.0 |
| **16** | 102.1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 99.2 | 0.0 | 8.2 | 152.6 | 0.0 | 0.0 | 45.2 | 0.0 | 0.9 |
| **17** | 0.0 | 0.0 | 0.2 | 33.4 | 0.0 | 124.4 | 0.0 | 0.0 | 0.0 | 68.9 | 0.0 | 0.0 | 77.4 | 0.2 | 0.0 | 0.0 |
| **18** | 0.0 | 0.1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 3.5 | 0.2 | 0.0 | 0.0 | 10.9 | 0.0 | 0.4 | 0.0 |
| **19** | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.2 | 2.5 | 2.2 | 7.2 | 0.0 | 37.0 | 0.2 | 0.0 | 0.2 |

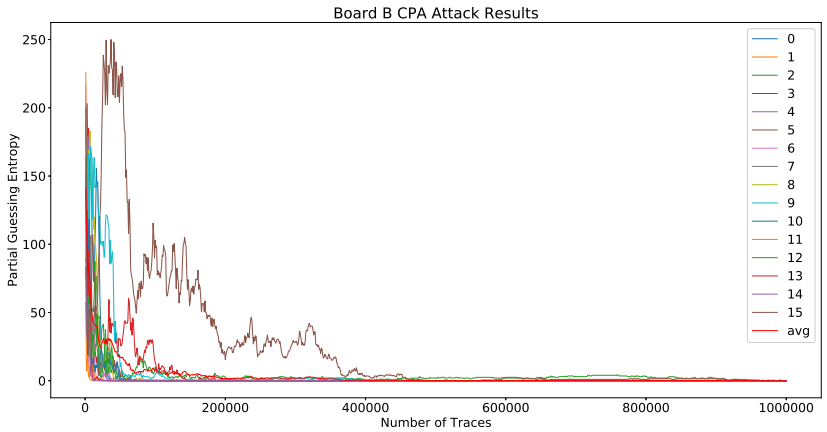


**Figure 14:** PGE Results for each of the 16 key bytes, along with average of all bytes when using the linear regulator with an external shunt resistor inline, but performing internal sampling of the $\frac{VDD_{core}}{4}$ rail

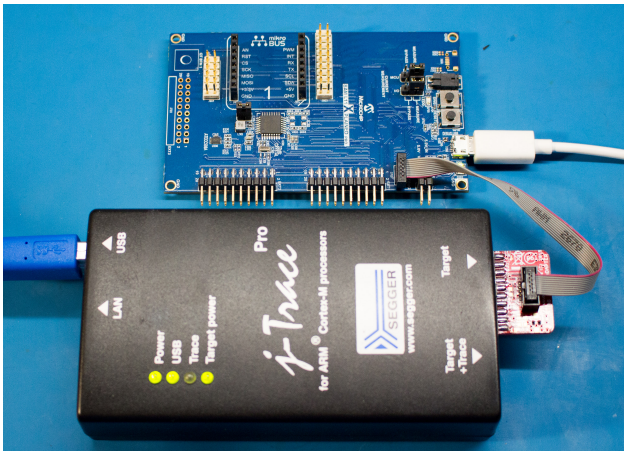


**Figure 15:** Power analysis using a the on-board ADC with the Microchip SAM L11 Xplained Pro.

leakage by placing an external shunt resistor into the $\text{VDD}_{\text{Core}}$ pin as shown in Figure 11, allowing completed recovery of the key in $1 \times 10^6$ traces. See Appendix A for more details of the board setup and recorded power traces.

### 3.3 Results of Board C & D

Boards C & D did not have significantly better results than the attack on the standard development kit, described in the next section. As described in Appendix A, the difference between boards C & D was only the number of decoupling capacitors mounted. The measurement is still taken using the internal ADC mux measuring $\frac{VDD_{core}}{4}$ as shown in Figure 12. We will directly move to discuss the results of the final attack on the standard development kit.

## 4 Internal Power Analysis on Development Kit

The final attack results will be presented using the Microchip SAM L11 Xplained Pro, a development kit for the ATSAML11. This allows an attack to proceed without any custom devices, and most closely represents the type of environment expected in production targets. The setup is similar to Boards C/D, and is shown in Figure 15. Again a J-Link is used for performing rapid data transfer on this device[6].

The PGE results in Figure 16 show that a complete stable key recovery occurs after $160 \times 10^6$ traces, but significant entropy reduction has occurred after $100 \times 10^6$ traces.

In addition to using the $\frac{VDD_{core}}{4}$ as an ADC input, we also explored various other analog input functions. The SAML11 for example has a number of on-board op-amps with various amplification modes, and an ability to perform differential measurements. We found less exploitable leakage using the analog input blocks, which is hypothesized to be because the analog circuitry is powered from the $VDD_{analog}$ domain. While it has been demonstrated that the digital I/O lines have sufficient coupling to leak side-channel information, on this platform it appears the combination of filters on the analog supplies and reduced attack efficiency due to the severe undersampling makes such attacks more difficult. This will

[6]These boards have an on-board debugger, which can be made J-Link compatible. The external debugger provides faster transfer rates, so the external debugger is used in this work. The use of the on-board debugger should also be possible to recreate the results using only the low-cost development board.

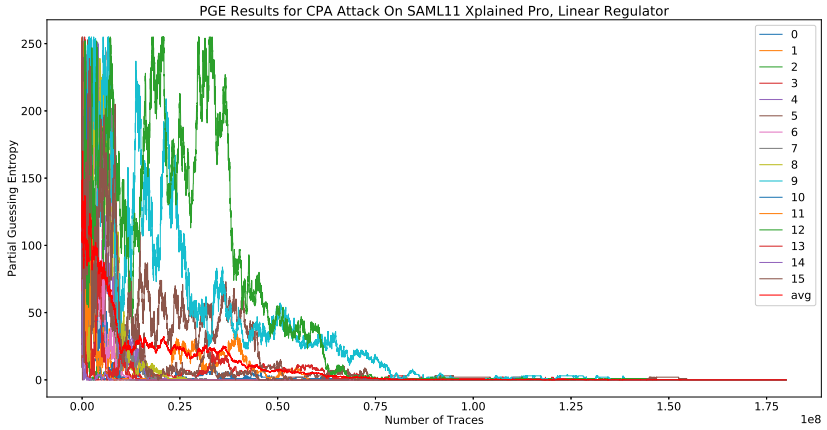


**Figure 16:** PGE Results for each of the 16 key bytes, along with average of all bytes when using the linear regulator on a standard development board and performing internal sampling of the $\frac{VDD_{core}}{4}$ rail.

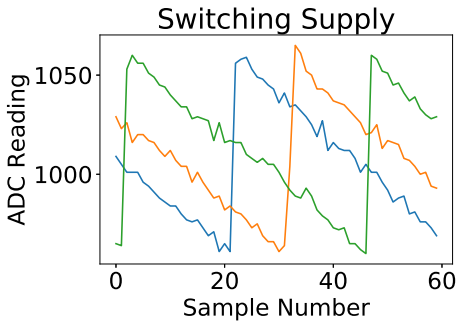


**Figure 17:** ADC measurement of switching power supply.

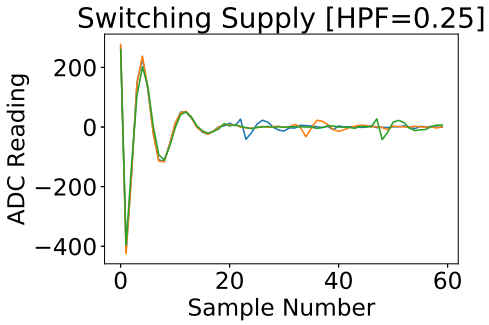


**Figure 18:** ADC measurement processed by high-pass filter.

vary widely with specifics of the platform – for example external connections may route the digital VDD to an analog I/O pin either directly or through a pull-up.

## 4.1 On-Board Regulator Mode

The SAML11 contains an on-board regulator, as the $\text{VDD}_{\text{Core}}$ operates from a lower voltage (1.2V) than the allowable I/O range. The default operation mode of this is to operate as a linear regulator, which does not require an external inductor and thus allows lower-cost designs. Where better power efficiency is desired, the device can also be operated in a switching regulator mode. The switching regulator adds considerable noise to the power traces – Figure 17 shows three overlaid power traces as measured by the on-board ADC, where the strong triangular ramp from the switching regulator can be seen. To counteract the switching regulator, the authors performed a high-pass filter ($f_c = 0.25$) on the recorded power traces, shown in Figure 18.

The effects of the high-pass filter attempt is shown in the PGE results of Figure 19. It can be seen that while some entropy reduction was achieved, the overall results did not fully recover the key in $240 \times 10^6$ traces. The work in Saab et al.[SLT16] successfully recovered a power trace from a switch-mode power supply, although the specific techniques from Saab et al. also solve jitter problems that are not present in our captures.

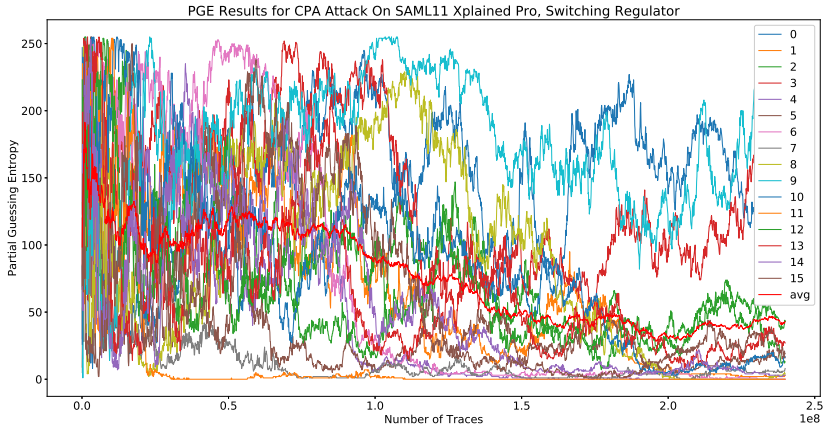


**Figure 19:** PGE Results for each of the 16 key bytes, along with average of all bytes when using the switching regulator with a high-pass filter processing the internally sampled $\frac{VDD_{core}}{4}$ traces.

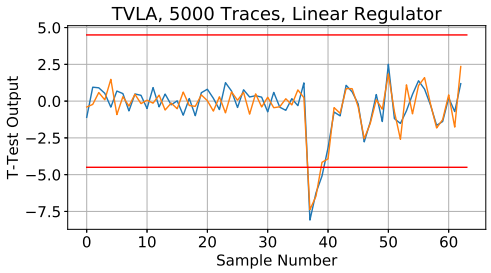


**Figure 20:** TVLA results with the linear regulator.

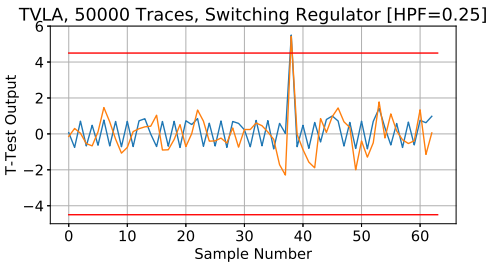


**Figure 21:** TVLA results with a high-pass filter processing the switching power supply traces.

## 4.2 TVLA Test Results

The usage of Test Vector Leakage Assesement (TVLA) provides a simple method of detecting leakage within a power trace, without determing the exact method required to exploit the leakage. This test methodology was first proposed by Goodwill et al.[GJJR11], and has gained considerable popularity due to the simple test method. In this section we use TVLA to perform such an evaluation, with the advantage that we already have information on the exploitability of this system.

Figure 20 shows the results of running the TVLA test using the development kit as the test target. A total of 10 000 traces were recorded, which are split into two groups of 5000 traces and the test repeated on both groups, as recommended in [GJJR11]. Note the TVLA test result exceeding the threshold around sample point 37 aligns with the known location of leakage from the CPA attack, verifying this leakage is from the operation of interest and not simple loading or unloading of data. Note leakage does not strongly occur elsewhere – this aligns with initial work on the SAML11, as for example other leakage models such as the Hamming weight of the first-round SBox output *did not* result in a successful attack. The leakage model in (1) was the only useful model we discovered.

The TVLA test allows rapid evaluation of the potential ability of an algorithm to be exploited using this remote side-channel. A total of 10 000 traces was needed for this evaluation, rather than the $160 \times 10^6$ required as part of the full CPA attack.

The results of the TVLA test on the switching regulator mode are shown in Figure 21. Note that without performing the high-pass filtering, the TVLA result did not suggest any leakage was present in the signal.

# 5 Countermeasures to Attack

The attack presented here demonstrates the possibility of a remote power-analysis attack. With the remote attack, certain configuration options such as using the switching-mode regulator increase the background noise considerably. This does not prevent the attack, but increases the number of observations required by an attacker. Two specific configuration options can help prevent the remote attack, and consideration of general side-channel power analysis countermeasures is also valuable.

## 5.1 Securing ADC Access

The ADC itself can be designated as a secure or non-secure peripheral. This work demonstrates that this peripheral should be constrained to the secure world due to the exploitable side-channel it can provide, and the driver must provide limited capability for an attacker to otherwise control the ADC from non-secure space. Other peripherals which may provide similar side-channel results should be suitably scrutinized.

## 5.2 Environment Validation from Secure World

Embedded systems often have the advantage of very well-known and consistent code execution paths. Validating the state of all used and unused peripherals may help with attack detection, for example by detecting the ADC in our example is enabled when it would not normally be. The secure code cannot access non-secure peripherals, however, meaning somewhat careful consideration of the partitioning of peripherals between the secure and non-secure worlds is still required.

## 5.3 Side-Channel Power Considerations for M23/M33

Devices with the M23/M33 core may not provide power analysis protection of their cryptographic core. Due to the additional resource (time, power, and cost) requirements of typical protected implementations, an unprotected cryptographic core is a reasonable market decision. Where the cryptographic core can be triggered by non-secure code, additional protocol-level restrictions should be placed *within the secure code* for systems which could be compromised by side-channel leakage. For example the secure code could limit the number of encryption operations performed with a single key, provided the key can be frequently changed. It is expected that the relatively high number of traces presented in this work using a basic CPA attack could be further reduced, as has been seen in the public DPA contest results[CDD+14].

A user may find it valuable to use a side-channel protected software implementation for certain operations which are more sensitive. Providing side-channel protection is critical for use-cases that rely on an attacker not being able to extract secrets from the device, even when they have some level of physical access. Some M33 core implementations such as the NXP LPC55S6x do advertise side-channel resistance – in this case using a special mode of AES called Indexed Code Book (AES-ICB). It is assumed AES-ICB is an implementation of the NXP 'Leakage Resilient Primitive'[NXP19], but these details are not publicly disclosed. Choosing to use AES-ICB means an incompatibility with existing libraries, as AES-ICB is not part of any standard group (IEEE, IETF, ISO).

# 6 Conclusions

Hardware security barriers are present as part of a Trusted Execution Environment (TEE), and assumed to stop leakage of critical secrets even when remote code execution privileges are obtained on one domain. The most prominent TEE for low-power devices is TrustZone for Cortex-M, available in Cortex-M23 and Cortex-M33 cores. Recently these devices have become physically available, led by Microchip with their SAML11. The TrustZone-M architecture prevents a variety of attacks by limiting the capability of an attacker who has code execution on a platform, since they will typically only gain code execution in the non-secure world. This work demonstrates that side-channel power analysis can be performed from the non-secure world against the secure world, using entirely on-board resources. This allows an attacker to perform side-channel power analysis without modifying the device, and potentially without even having physical access.

The specific examples performed here were performed against the Microchip SAML11 device. This device had the advantage of having an ADC which allows measurement of the $\text{VDD}_{\text{Core}}$ , but it would be expected that other ADC channels may provide leakage paths in many systems[GKT19]. Performing the attack on an unmodified board with a standard CPA attack required approximately $160 \times 10^6$ traces on this device. The architecture presented in this paper allows very rapid collection of data traces, and can be used to assist with evaluation of other M23/M33 devices.

Developing secure applications requires knowledge of realistic threat models. The Arm TrustZone-M technology is critical in blocking many of these threats by bringing security improvements to very low-cost and low-power devices, such as execute-only memory and partitioning of designs into secure and non-secure worlds with hardware-based enforcement. TrustZone-M does not, however, specify requirements against side-channel power analysis attacks, and thus TrustZone-M associated cryptographic cores may have side-channel power leakages as in the SAML11. This paper demonstrates that a cross-domain power analysis attack is possible, meaning this threat model needs to be carefully considered to prevent the exposure of secret data.

This is especially critical when additional algorithms are implemented in the secure world of the TrustZone-M. These algorithms may be vulnerable to side-channel power analysis, especially implementations that have SPA type leakages which are well known to be exploitable with extremely band-limited information.

# Acknowledgments

The authors are grateful for many constructive and detailed comments from external reviewers in the development of this final paper. The external reviewers are currently anonymous as the authors are waiting for permission to include their names when possible (some reviewers are anonymous by design).

# References

[ABG10] Onur Aciiçmez, Billy Bob Brumley, and Philipp Grabher. New Results on Instruction Cache Attacks. In Stefan Mangard and François-Xavier Standaert, editors, *Cryptographic Hardware and Embedded Systems, CHES 2010, 12th International Workshop, Santa Barbara, CA, USA, August 17-20, 2010. Proceedings*, volume 6225 of *Lecture Notes in Computer Science*, pages 110–124. Springer, 2010.

[ANR+18] N. Asokan, T. Nyman, N. Rattanavipanon, A. Sadeghi, and G. Tsudik. ASSURED: Architecture for Secure Software Update of Realistic Embedded

Devices. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 37(11):2290–2300, November 2018.

[BCO04] Eric Brier, Christophe Clavier, and Francis Olivier. Correlation Power Analysis with a Leakage Model. In Marc Joye and Jean-Jacques Quisquater, editors, *Cryptographic Hardware and Embedded Systems - CHES 2004*, Lecture Notes in Computer Science, pages 16–29. Springer Berlin Heidelberg, 2004.

[Ber05] Daniel J. Bernstein. Cache-timing attacks on AES. 2005.

[BLLB+18] Sebanjila Kevin Bukasa, Ronan Lashermes, Hélène Le Bouder, Jean-Louis Lanet, and Axel Legay. How TrustZone Could Be Bypassed: Side-Channel Attacks on a Modern System-on-Chip. In Gerhard P. Hancke and Ernesto Damiani, editors, *Information Security Theory and Practice*, Lecture Notes in Computer Science, pages 93–109. Springer International Publishing, 2018.

[Car17] Pierre Carru. Attack TrustZone with Rowhammer, 2017.

[CDD+14] Christophe Clavier, Jean-Luc Danger, Guillaume Duc, M. Abdelaziz Elaabid, Benoît Gérard, Sylvain Guilley, Annelie Heuser, Michael Kasper, Yang Li, Victor Lomné, Daisuke Nakatsu, Kazuo Ohta, Kazuo Sakiyama, Laurent Sauvage, Werner Schindler, Marc Stöttinger, Nicolas Veyrat-Charvillon, Matthieu Walle, and Antoine Wurcker. Practical improvements of side-channel attacks on AES: feedback from the 2nd DPA contest. *Journal of Cryptographic Engineering*, 4(4):259–274, November 2014.

[CPM+18] Giovanni Camurati, Sebastian Poeplau, Marius Muench, Tom Hayes, and Aurélien Francillon. Screaming Channels: When Electromagnetic Side Channels Meet Radio Transceivers. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, pages 163–177, New York, NY, USA, 2018. ACM.

[GJJR11] Gilbert Goodwill, Benjamin Jun, J. Jaffe, and Pankaj Rohatgi. A testing methodology for side channel resistance. 2011.

[GKT11] Jean-François Gallais, Ilya Kizhvatov, and Michael Tunstall. Improved trace-driven cache-collision attacks against embedded aes implementations. In Yongwha Chung and Moti Yung, editors, *Information Security Applications*, pages 243–257, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[GKT19] Dennis Gnad, Jonas Krautter, and Mehdi Tahoori. Leaky Noise: New Side-Channel Attack Vectors in Mixed-Signal IoT Devices. *TCHES*, 3, 2019.

[GPPT15] Daniel Genkin, Lev Pachmanov, Itamar Pipman, and Eran Tromer. Stealing Keys from PCs Using a Radio: Cheap Electromagnetic Attacks on Windowed Exponentiation. In Tim Güneysu and Helena Handschuh, editors, *Cryptographic Hardware and Embedded Systems – CHES 2015*, Lecture Notes in Computer Science, pages 207–228. Springer Berlin Heidelberg, 2015.

[GST14] Daniel Genkin, Adi Shamir, and Eran Tromer. RSA Key Extraction via Low-Bandwidth Acoustic Cryptanalysis. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology – CRYPTO 2014*, Lecture Notes in Computer Science, pages 444–461. Springer Berlin Heidelberg, 2014.

[KDK+14] Yoongu Kim, Ross Daly, Jeremie Kim, Chris Fallin, Ji Hye Lee, Donghyuk Lee, Chris Wilkerson, Konrad Lai, and Onur Mutlu. Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance

Errors. In *Proceeding of the 41st Annual International Symposium on Computer Architecuture*, ISCA '14, pages 361–372, Piscataway, NJ, USA, 2014. IEEE Press.

[KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential Power Analysis. In *Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '99, pages 388–397, Berlin, Heidelberg, 1999. Springer-Verlag.

[LGS+16] Moritz Lipp, Daniel Gruss, Raphael Spreitzer, Clémentine Maurice, and Stefan Mangard. ARMageddon: Cache Attacks on Mobile Devices. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 549–564, Austin, TX, 2016. USENIX Association.

[Lim17] ARM Limited. TrustZone technology for the ARMv8-M architecture, March 2017. https://static.docs.arm.com/100690/0200/armv8m_trustzone_technology_100690_0200.pdf.

[LW18] Ben Lapid and Avishai Wool. Cache-Attacks on the ARM TrustZone implementations of AES-256 and AES-256-GCM via GPU-based analysis. Technical Report 621, 2018.

[Mas94] J.L. Massey. Guessing and entropy. In *Proceedings of 1994 IEEE International Symposium on Information Theory*, page 204, Trondheim, Norway, 1994. IEEE.

[MM12] Amir Moradi and Oliver Mischke. On the Simplicity of Converting Leakages from Multivariate to Univariate - Case Study of a Glitch-Resistant Masking Scheme -. In *IACR Cryptology ePrint Archive*, 2012.

[NCC18] NCCG roup. CacheGrab, September 2018. https://github.com/nccgroup/cachegrab.

[NXP19] NXP. Leakage Resilient Primitive (LRP) Specification, 2019.

[OC13] Colin O'Flynn and Zhizhang Chen. Synchronous sampling and clock recovery of internal oscillators for side channel analysis and fault injection. *Journal of Cryptographic Engineering*, 5:53–69, 2013.

[OC14] Colin O'Flynn and Zhizhang Chen. ChipWhisperer: An Open-Source Platform for Hardware Embedded Security Research. *IACR Cryptology ePrint Archive*, 2014:204, 2014.

[RPD+18] Chethan Ramesh, Shivukumar B. Patil, Siva Nishok Dhanuskodi, George Provelengios, Sébastien Pillement, Daniel Holcomb, and Russell Tessier. FPGA Side Channel Attacks without Physical Access. *2018 IEEE 26th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, pages 45–52, 2018.

[SGMT18a] Falk Schellenberg, Dennis R. E. Gnad, Amir Moradi, and Mehdi B. Tahoori. Remote Inter-chip Power Analysis Side-channel Attacks at Board-level. In *Proceedings of the International Conference on Computer-Aided Design*, ICCAD '18, pages 114:1–114:7, New York, NY, USA, 2018. ACM.

[SGMT18b] Falk Schellenberg, Dennis R. E. Gnad, Amir Moradi, and Mehdi Baradaran Tahoori. An inside job: Remote power analysis attacks on FPGAs. *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1111–1116, 2018.

[SLT16] S. Saab, A. Leiserson, and M. Tunstall. Key extraction from the primary side of a switched-mode power supply. In *2016 IEEE Asian Hardware-Oriented Security and Trust (AsianHOST)*, pages 1–7, December 2016.

[SPK+10] Jörn-Marc Schmidt, Thomas Plos, Mario Kirschbaum, Michael Hutter, Marcel Medwed, and Christoph Herbst. Side-Channel Leakage across Borders. In Dieter Gollmann, Jean-Louis Lanet, and Julien Iguchi-Cartigny, editors, *Smart Card Research and Advanced Application*, Lecture Notes in Computer Science, pages 36–48. Springer Berlin Heidelberg, 2010.

[SPSG18] Manuel San Pedro, Victor Servant, and Charles Guillemet. LASCAR, December 2018. https://github.com/LedgerHQ/lascar.

[TSS17] Adrian Tang, Simha Sethumadhavan, and Salvatore Stolfo. CLKSCREW: Exposing the Perils of Security-Oblivious Energy Management. pages 1057–1074, 2017.

[ZS18] Mark Zhao and G. Edward Suh. FPGA-Based Remote Power Side-Channel Attacks. *2018 IEEE Symposium on Security and Privacy (SP)*, pages 229–244, 2018.

[ZSS+16] Ning Zhang, Kun Sun, Deborah Shands, Wenjing Lou, and Y. Thomas Hou. TruSpy: Cache Side-Channel Information Leakage from the Secure World on ARM Devices. Technical Report 980, 2016.

[ZSS+18] Ning Zhang, Kun Sun, Deborah Shands, Wenjing Lou, and Yiwei Thomas Hou. TruSense: Information Leakage from TrustZone. *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, pages 1097–1105, 2018.

# Appendix A: Hardware Details

## 6.1 Custom Boards

The schematic for the base of the custom boards is given in Figure 22. These boards plug into the ChipWhisperer CW308T, which has a schematic posted at http://wiki.newae.com. A higher resolution photo of the boards is provided at Figure 23. Of these boards, boards C & D both have a solder blob at R1 instead of a shunt resistor. In addition it can be seen that board D has an additional decoupling capacitor mounted slightly nearer to the chip.

Board A contains a "flywire" op-amp. This op-amp takes an AC-coupled version of the signal from the shunt resistor, and centers it approximately in the middle of the ADC input range. This is fed into the AD[1] pin of the SAML11, which is used for sampling the power consumption that has been externally conditioned.

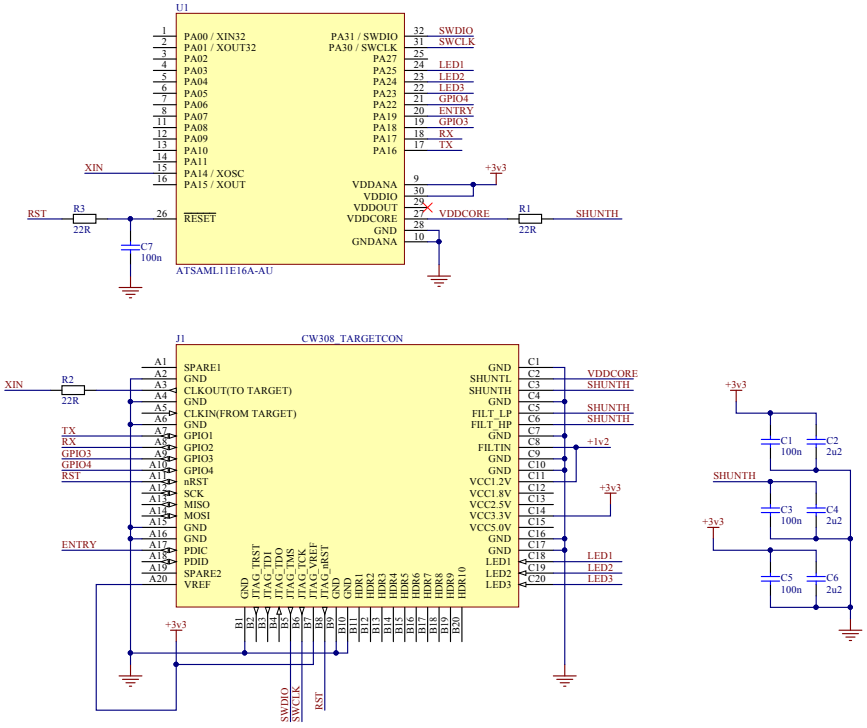


**Figure 22:** SAML11 Schematic

## 6.2 Capture Setup

As mentioned earlier, attacks on Boards A through D, as well as the attack on the SAML11 development board utilized Segger's RTT technology to enable high capture speeds by transferring data over a debug interface. A J-Link Pro communicating over SWD at 15MHz was used as the debugger. Segger's RTT library was used on the SAML11 for these transfers.

Before each capture campaign, the target device was reflashed using Ozone. With Ozone still running, pylink was used to connect the capture script to the debugger and transfer data using RTT.

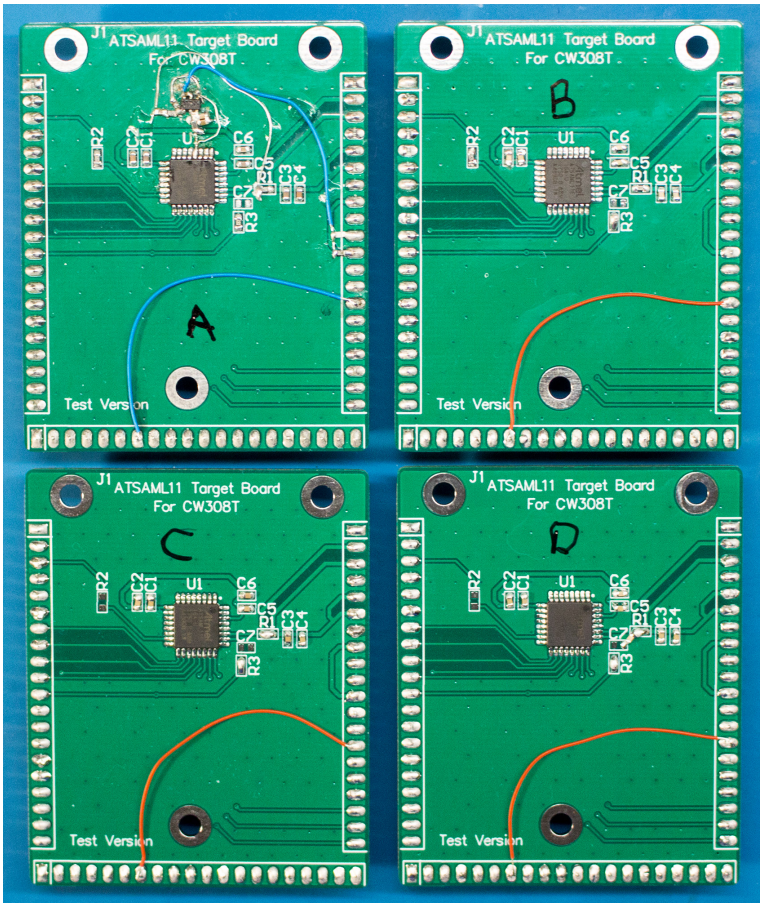


**Figure 23:** Higher-Resolution Photos of Test Boards

The SAML11's 16MHz internal oscillator was used as the source for all clocks on the device. No divider or other clock modifications were used.

The SAML11's ADC was setup to use the same generic clock controller (GCLK) as the main clock. As stated in the device's datasheet, GCLKs are asynchronous, with syncing performed when certain peripheral registers are written. If a separate GCLK is used, these registers may have to be used to maintain a constant phase offset. The reference used for the ADC was the internal variable reference voltage set at 1.0V, with the attack being repeated on Board D with $\frac{VDD_{ANA}}{2}$ as a reference with similar results. The ADC was used in freerunning mode in conjunction with the SAML11's DMAC to capture asynchronously to the main core. Freerunning mode was enabled immediately before the offset delays. Each trace was composed of 64 12-bit ADC readings, stored as 64 16-bit unsigned integers.

Encryptions and trace captures were performed in sets of 32, with plaintext being generated on the SAML11 using a 32 bit xorshift. The seed for this xorshift was updated after each capture set.

# Appendix B: Industry Considerations

In an effort to help industry usage of this paper, the following common Q&A is provided for those readers who may be less familiar with side-channel power analysis.

### Q. Is an attack needing 160 000 000 traces really a practical threat?

**A.** We have chosen to use a basic CPA attack in this work. Moving from $160 \times 10^6$ to $1 \times 10^6$ traces is a reasonable for an attacker by moving to more advanced attacks, as can be seen by the improvements occurring within the DPA Contest. Attackers which are quasi-local (local but not able to modify the hardware) may easily be able to record a considerable number of traces. Using the debug interface method in this paper for example, $1 \times 10^6$ traces could be recorded in about 15 minutes.

This paper demonstrates the fundamental ability of power analysis attacks to be successful, even across security domains of TrustZone-M. Users of these features must be aware of these attacks to make the correct decisions about system configuration.

### Q. Why wasn't this attack against a real product?

**A.** Currently, no commercial products based on released M23/M33 cores are known to be in the market. We have chosen to attempt a generic attack against a specific implementation of a TrustZone-M cryptographic primitives, rather than waiting for a specific commercial example. We hope this allows commercial products under development to consider the threat model described in this paper, and be designed to reduce the value of this attack.

### Q. Is there a bullet-proof countermeasure?

**A.** Side-channel analysis is most easily stopped by eliminating the value in leaking the secret. Re-use of keys across many devices for example is a prime target for side-channel analysis, since an attack against a single device can be replicated across all other devices with that shared key. This may mean a better method of performing key distribution, or using features such as physically unclonable functions (PUFs) which provide a unique per-device key.

### Q. Would this apply to a HSM?

**A.** The general idea of this paper is to apply power analysis across security domains. We have concentrated on the SAML11 / TrustZone-M to provide concrete results, but the general idea would apply across other device security domains.

### Q. Is this attack against only the SAML11, what about other M23/M33 devices?

**A.** The SAML11 was selected for this work due to being the first commercially available microcontroller with a M23 core. In addition, it has a reasonably fast ADC along with an ADC input that connects to $\text{VDD}_{\text{Core}}$ . The AES core present in the SAML11 is especially vulnerable to power analysis due to the implementation appearing not to be a full hardware implementation. Other devices may have a weaker side-channel power analysis leakage and more noise between the ADC and $\text{VDD}_{\text{Core}}$ , but the general conclusions are expected to hold across other devices. Evaluation of a potential device would be required to form a more concrete conclusion.

**Q. Is this specific to AES? What about other cores or algorithms?**

**A.** We have used a standard side-channel power analysis attack against the AES core in this device. This paper demonstrates such an attack could be performed using on-board circuitry to perform the measurement, but the evaluation of the core is most easily done using classic (external measurement) based equipment. Many other algorithms have relevant power analysis attacks published against them including RSA, ECC, DES, SHA-2, etc. Such vulnerabilities are well-known, but many designs assume that physical access was required to exploit them.

**Q. What exactly is the PGE result you are using?**

**A.**The 'guessing entropy' is defined as the "average number of successive guesses required with an optimum strategy to determine the true value of a random variable $X$"[Mas94]. The 'optimum strategy' here is to rank the possible values of the subkey from most to least likely based on the value of the correlation attack (higher correlation output is more likely).

The 'partial' refers to the fact that we are finding the guessing entropy on each subkey. This gives us a PGE for each of the 16 subkeys. A PGE of 0 indicates the subkey is perfectly known, a PGE of 10 indicates that 10 guesses were incorrectly ranked higher than the correct guess.

The attack algorithm is given access to $1, 2, \cdots, N$ traces, and the PGE for each subkey is calculated. When possible, to improve consistency the PGE for each subkey is averaged over several attacks (trials). Finally, we can average the PGE over all 16 subkeys to generate a single 'average PGE' for the attack.

A PGE of '0' for a byte means that that key was completely recovered. Often entropy reduction is sufficient as goal – a result where a single byte is unknown for example is indistinguishable in practice from a "full break" since the remaining entropy is so low.