





I, for One, Welcome Our New Power Analysis Overlords

aka: ChipWhisperer-Lint

by Colin O'Flynn

Something Something About Me.

- Assistant Professor → Dalhousie University
 - Very recent change – if you are interested in doing a MAsc/PhD in this work and living in Halifax, NS, Canada while so please talk to me!
- C.T.O. → NewAE Technology Inc.
 - Startup on embedded security, we make test equipment, run trainings, do open-source stuff!

Select	Image	Mouser Part #	Mfr. Part #	Mfr.	Description	Datasheet	Availability	Pricing (CAD)	Quantity	RoHS
<input type="checkbox"/>	 Enlarge	343-SCAPACKL2	NAE-SCAPACK-L2	NewAE	Development Boards & Kits - AVR ChipWhisperer Level 2 Starter Kit Learn More	Datasheet	17 In Stock More Info Available	1: \$1,258.56	<input type="text"/> Buy Min.: 1 Mult.: 1	 Details
<input type="checkbox"/>	 Enlarge	343-CW1200KIT	NAE-CW1200-KIT	NewAE	Development Boards & Kits - AVR ChipWhisperer Pro (Level 3 Starter Kit) Learn More	Datasheet	3 In Stock More Info Available	1: \$5,244.00	<input type="text"/> Buy Min.: 1 Mult.: 1	 Details
<input type="checkbox"/>	 Enlarge	343-CW11732PART	NAE-CW1173-2PART	NewAE	Development Boards & Kits - AVR ChipWhisperer-Lite 2-Part with XMEGA Learn More	Datasheet	8 In Stock	1: \$448.50	<input type="text"/> Buy Min.: 1 Mult.: 1	 Details

AES Hardware Acceleration?

SAM L10 and L11 Microcontroller Family

Industry's Lowest Power 32-bit MCUs, First to Offer Chip-Level : Arm® TrustZone® Technology

With the increasing growth of IoT end points and, consequently, the increased frequency of security power consumption while adding robust security. The SAM L10 and L11 MCU family takes an innovative variety of peripherals, including security features, into the industry's lowest power MCU in its class. 1 battery constraints of less power-efficient MCUs. These MCUs run at 32 MHz with memory configuration options: SAM L10 and SAM L11 and boast ultra-low power consumption as well as an enhanced SAM L11 variant adds integrated hardware security. They both come in 24- and 32-pin package capacitive touch and general purpose embedded control applications.

Robust Security

SAM L11 MCUs integrates chip and ARM® TrustZone® technology protect from both physical and They are supported by a comprehensive security solution framework implementation of security. In a SAM L11 provide strong resistance to software attacks there by increasing reliability and avoiding any data

STM32F405/415

The STM32F405/415 lines are designed for medical, industrial performance, embedded memories and rich peripheral The STM32F405/415 offers the full performance of the

Performance: At 168 MHz, the STM32F405/415 delivers states using ST's ART Accelerator. The DSP instruction

Power efficiency: ST's 90 nm process, ART Accelerator and the dynamic power executing from Flash memory to be as low as 238 µA/MHz at 168 MHz.

Rich connectivity: Superior and innovative peripherals

- 2x USB OTG (one with HS support)
- Audio: dedicated audio PLL and 2 full duplex I²S
- Up to 15 communication interfaces (including 6x USARTs running at up to 10.5 M 3x I²C, 2x CAN, SDIO)
- Analog: two 12-bit DACs, three 12-bit ADCs reaching 2.4 MSPS or 7.2 MSPS in
- Up to 17 timers: 16- and 32-bit running at up to 168 MHz
- Easily extendable memory range using the flexible static memory controller support
- Analog true random number generator
- The STM32F415 also integrates a crypto/hash processor providing hardware acceleration

Summary

Atmel's SAM4L series is a member of a family of Flash microcontrollers based on the high performance 32-bit ARM Cortex-M4 RISC processor running at frequencies up to 48MHz.

The SAM4L series embeds state-of-the-art picoPower technology for ultra-low power consumption. Combined power control techniques are used to bring active current consumption down to 90µA/MHz. The device allows a wide range of options between functionality and power consumption, giving the user the ability to reach the lowest possible power consumption with the feature set required for the application. The WAIT and RETENTION modes provide full logic and RAM retention, associated with fast wake-up capability (<1.5µs) and a very low consumption of, respectively, 3 µA and 1.5 µA. In addition, WAIT mode supports SleepWalking features. In BACKUP mode, CPU, peripherals and RAM are powered off and, while consuming less than 0.9µA with external interrupt wake-up supported.

The SAM4L series offers a wide range of peripherals such as segment LCD controller, embedded hardware capacitive touch (QTouch), USB device & embedded host, 128-bit AES and audio interfaces in addition to high speed serial peripherals such as USART, SPI and I²C. Additionally the Peripheral Event System and SleepWalking each other and make a qualified events or threshold.

Features

AVR1318: Using the XMEGA built accelerator

Features

- Full compliance with AES (FIPS Publication 197, 2002)
 - Both encryption and decryption procedures
- 128-bit Key and State memory
- XOR load option to State memory useful for cipher block coding
- Sequential access to State and Key memories
- Optional Interrupt- and DMA request on AES complete

1 Introduction

The XMEGA™ AES Crypto Module supports the Advanced Encryption Standard (AES), and can perform encryption and decryption. The module supports a key length of 128 bits. The 128-bit key block and 128-bit data block (plaintext or ciphertext) must be loaded into the Key and State memory in the AES Crypto Module. The AES uses 375 clock cycles to execute one encryption/decryption operation.

Kinetis K24F Sub-Family Data Sheet

120 MHz ARM® Cortex®-M4-based Microcontroller with FPU

The K24 product family members are optimized for cost-sensitive applications requiring low-power, USB connectivity, and up to 256 KB of embedded SRAM. These devices share the comprehensive enablement and scalability of the Kinetis family.

This product offers:

- Run power consumption down to 250 µA/MHz. Static power consumption down to 5.8 µA with full state retention and 5 µs wakeup. Lowest Static mode down to 339 nA
- USB LS/FS OTG 2.0 with embedded 3.3 V, 120 mA LDO Vreg, with USB device crystal-less operation

Performance

- Up to 120 MHz ARM® Cortex®-M4 core with DSP instructions and floating point unit

Memories and memory interfaces

- Up to 1 MB program flash memory and 256 KB RAM
- FlexBus external bus interface

System peripherals

- Multiple low-power modes, low-leakage wake-up unit
- Memory protection unit with multi-master protection
- 16-channel DMA controller
- External watchdog monitor and software watchdog

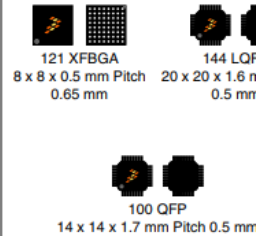
Security and integrity modules

- Hardware CRC module
- Hardware random-number generator
- Hardware encryption supporting DES, 3DES, AES, MD5, SHA-1, and SHA-256 algorithms
- 128-bit unique identification (ID) number per chip

Analog modules

Applic

MK24FN1M0VLQ12
MK24FN1M0VLL12
MK24FN1M0VDC12



Communication interfaces

- USB full-/low-speed On-the-Go controller
- Controller Area Network (CAN) module
- Three SPI modules
- Three I2C modules. Support for up to 1 Mbit/s
- Six UART modules
- Secure Digital Host Controller (SDHC)
- I2S module

Timers

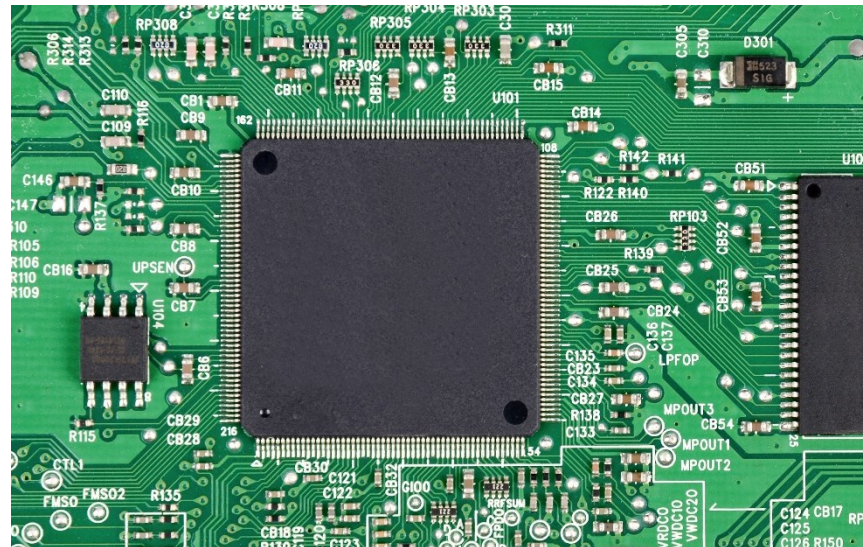
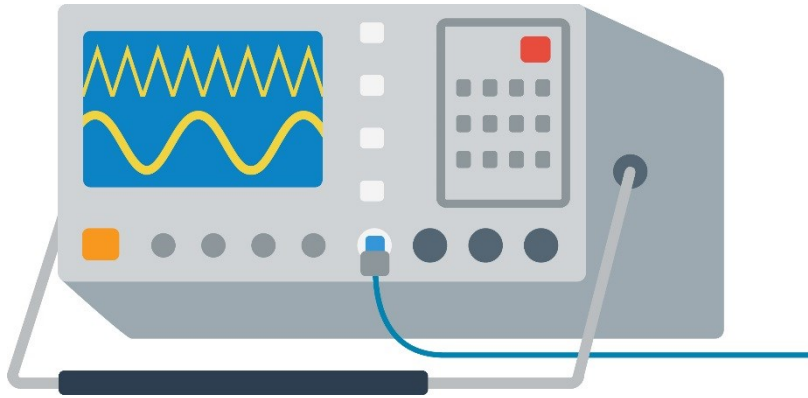
- Two 8-channel Flex-Timers (PWM/Motor control)
- Two 2-channel FlexTimers (PWM/Quad deco)
- 32-bit PITs and 16-bit low-power timers
- Real-time clock
- Programmable delay block

Clocks

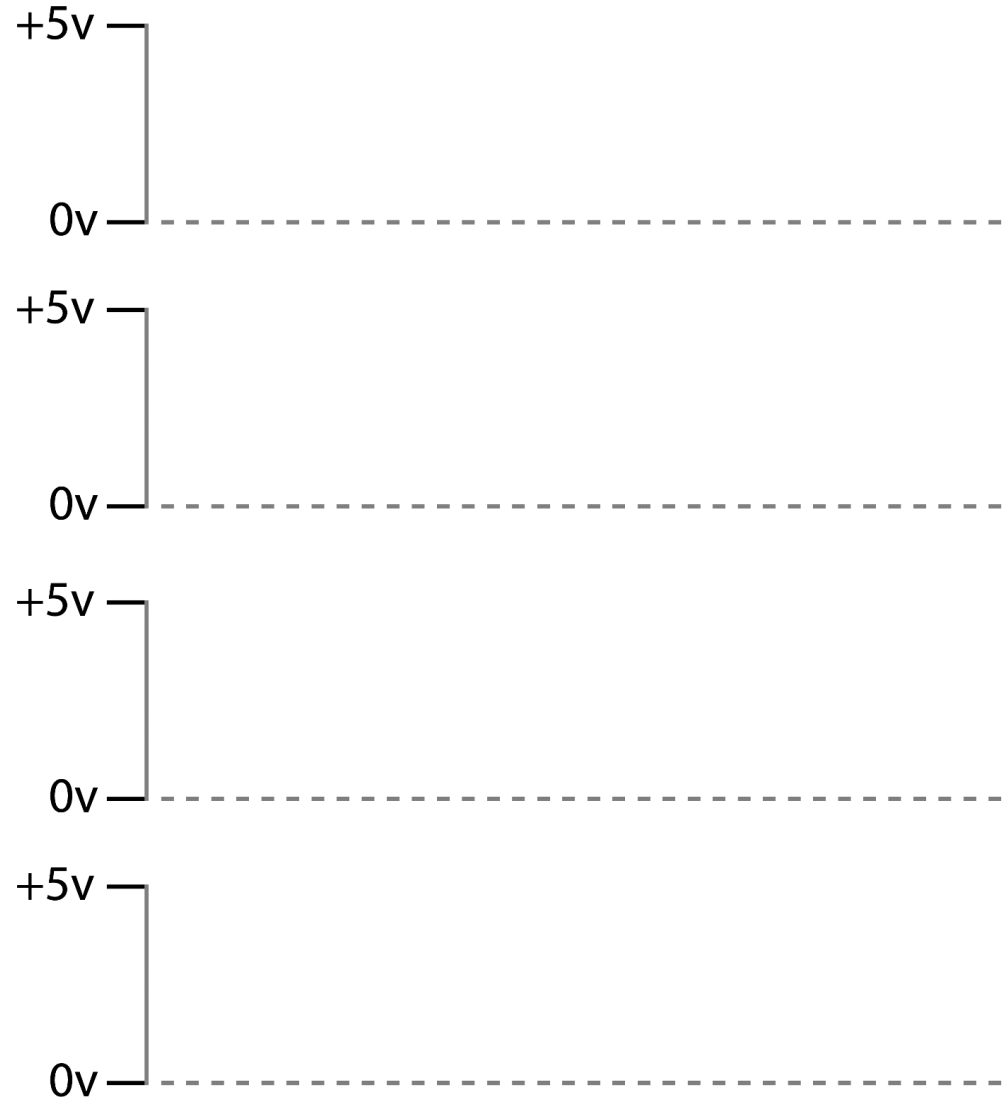
- 3 to 32 MHz and 32 kHz crystal oscillator
- PLL, FLL, and multiple internal oscillators
- 48 MHz Internal Reference Clock (IRC48M)

Operating Characteristics

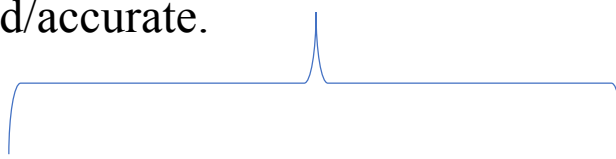
Side Channel Power Analysis



Data Busses...



Model of power consumption used with guess of secret information. Only ONE guess should match “real” measurement assuming our model is good/accurate.



Input Data	Power Measurement	Output of Leakage Model Byte Guess = 0x00	Output of Leakage Model Byte Guess = 0x01
0xC7			
0x1F			
0x2C			
0x89			
0x01			
0xD2			

Using in Real Life

Why Light Bulbs May Be the Next Hacker Target

By JOHN MARKOFF NOV. 3, 2016

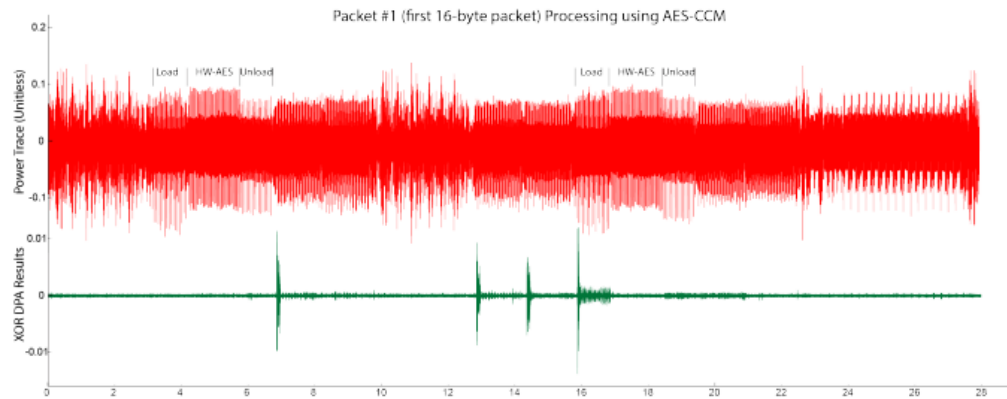
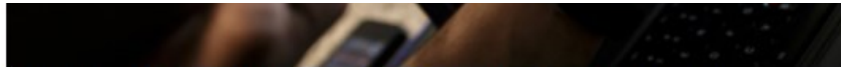


Figure 8. Power analysis of processing a single 16-byte block by the cryptographic bootloader.



The Internet of Things, activated through apps, promises tremendous convenience to homeowners. But it may also prove irresistible to hackers. Carlos Gonzalez for The New York Times

SAN FRANCISCO — The so-called Internet of Things, its proponents argue, offers many benefits: energy efficiency, technology so convenient it can anticipate what you want, even reduced congestion on the roads.

Now here's the bad news: Putting a bunch of wirelessly connected devices in one area could prove irresistible to hackers. And it could allow them to spread malicious code through the air, like a flu virus on an airplane.

Researchers report in a [paper](#) to be made public on Thursday that they have uncovered a flaw in a wireless technology that is often included in smart

On the Power of Power Analysis in the Real World: A Complete Break of the KEELoQ Code Hopping Scheme

Thomas Eisenbarth¹, Timo Kasper¹, Amir Moradi^{2,*}, Christof Paar¹, Mahmoud Salmasizadeh², and Mohammad T. Manzuri Shalmani²

¹ Horst Görtz Institute for IT Security
Ruhr University Bochum, Germany

² Department of Computer Engineering and Electronic Research Center
Sharif University of Technology, Tehran, Iran
{eisenbarth,tkasper,moradi,cpaar}@crypto.rub.de
{salmasi,manzuri}@sharif.edu

Abstract. KEELoQ remote keyless entry systems are widely used for access control purposes such as garage openers or car door systems. We present the first successful differential power analysis attacks on nume-

Schneier on Security



[Blog](#)

[Newsletter](#)

[Books](#)

[Essays](#)

[News](#)

[Talks](#)

[Academic](#)

[About Me](#)

[Blog](#) >

Breaking the Xilinx Virtex-II FPGA Bitstream Encryption

It's a [power-analysis attack](#), which makes it much harder to defend against. And since the attack model is an engineer trying to reverse-engineer the chip, it's a valid attack.

Abstract: Over the last two decades FPGAs have become central components for many advanced digital systems, e.g., video signal processing, network routers, data acquisition and military systems. In order to protect the intellectual property and to prevent fraud, e.g., by cloning an FPGA or manipulating its content, many current FPGAs employ a bitstream encryption feature. We develop a successful attack on the bitstream encryption engine integrated in the widespread Virtex-II Pro FPGAs from Xilinx, using side-channel analysis. After measuring the power consumption of a single power-up of the device and a modest amount of off-line computation, we are able to recover all three different keys used by its triple DES module. Our method allows extracting secret keys from any real-world device where the bitstream encryption feature of Virtex-II Pro is enabled. As a consequence, the target product can be cloned and manipulated at will of the attacker. Also, more advanced attacks such as reverse

Problem: What is Leakage Model?

Improved Side-Channel Analysis Attacks on Xilinx Bitstream Encryption of 5, 6, and 7 Series

Amir Moradi and Tobias Schneider

Horst Görtz Institute for IT Security, Ruhr-Universität Bochum, Germany
{amir.moradi, tobias.schneider-a7a}@rub.de

We have tried many different hypotheses for the design architecture, and finally the highest correlation has been observed considering the same architecture as shown in Figure 2 but with $HD(R_1, R_{i+1})$, $0 < i < 14$ model. Although no design architecture can justify why the SCA leakage depends on the state register at round $i + 1$ and that of the first round, such a model leads to considerably high correlations² as shown in Figure 6.

² As a side note, we found this leakage model by coincidence, and it is valid for all considered FPGAs and for both power and EM leakages.

Don't be "too smart", try all leakage models if you want CPA attack maybe?

But very time consuming to test lots of models... how to improve on this?

What about Machine Learning?

I tried it... once.

(Circa 2003).



What about Machine Learning?

OK twice. But I mean it was running in BASIC*, so it doesn't count.

(Circa 2002).

*Artificial Neural Network implemented in BASCOM-AVR, which is compiled BASIC running on embedded microcontroller.



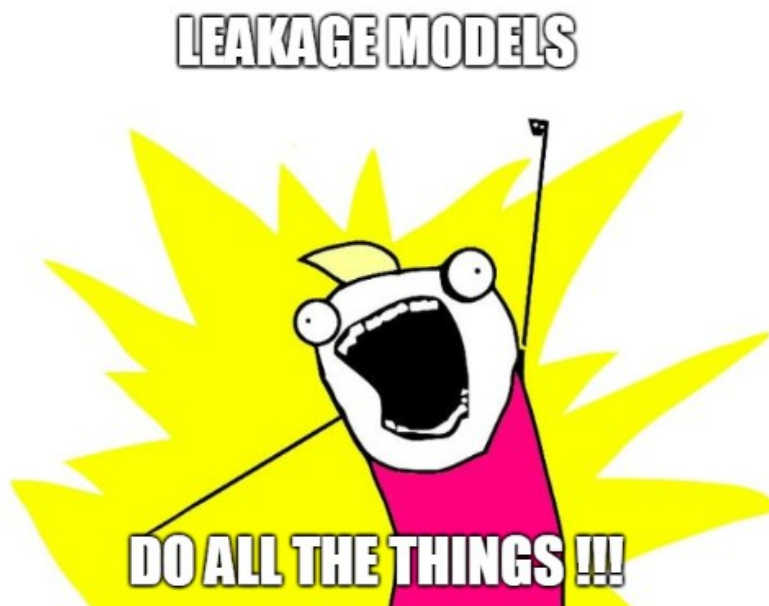
What about Machine Learning?

- Problem is very well formed for machine learning:
 - Huge datasets (we can create as many as we want).
 - SNR is something we can control/optimize very easily.
 - Relatively easy to define outcome.
- Problems with machine learning:
 - I haven't looked at it since 2003 with my sweet ANN robot.
 - We have a pretty good idea of underlying models, and we'd like to validate specific ones.

Dumb Method: Step #1

```
class AES128_Leakage(Leakage_Base):
    # Implements AES-128, saving internal states in a dictionary for analysis
    name = 'AES128'
    ks = None

    leakage_points = [
        "Plaintext",
        "Key",
        "Round 0: AddRoundKey Output",
        "Round 1: SubBytes Output",
        "Round 1: ShiftRows Output",
        "Round 1: MixColumns Output",
        "Round 1: RoundKey",
        "Round 1: AddRoundKey Output",
        "Round 2: SubBytes Output",
        "Round 2: ShiftRows Output",
        "Round 2: MixColumns Output",
        "Round 2: RoundKey",
        "Round 2: AddRoundKey Output",
        "Round 3: SubBytes Output",
        "Round 3: ShiftRows Output",
        "Round 3: MixColumns Output",
        "Round 3: RoundKey",
        "Round 3: AddRoundKey Output",
        "Round 4: SubBytes Output",
        "Round 4: ShiftRows Output",
        "Round 4: MixColumns Output",
        "Round 4: RoundKey",
        "Round 4: AddRoundKey Output",
        "Round 5: SubBytes Output",
        "Round 5: ShiftRows Output",
        "Round 5: RoundKey",
        "Round 5: MixColumns Output",
        "Round 5: AddRoundKey Output",
        "Round 6: SubBytes Output",
        "Round 6: ShiftRows Output",
```



```
Nr = 10
state = pt
ret['Plaintext'] = self.flatten(state[:])

ret['Key'] = self.flatten(self.ks[0])

state = [state[i] ^ self.ks[0][i] for i in range(16)]
ret['Round 0: AddRoundKey Output'] = self.flatten(state[:])

for r in range(1, Nr):
    state = subbytes(state)
    ret['Round ' + str(r) + ': SubBytes Output'] = self.flatten(state[:])

    state = shiftrows(state)
    ret['Round ' + str(r) + ': ShiftRows Output'] = self.flatten(state[:])

    state = mixcolumns(state)
    ret['Round ' + str(r) + ': MixColumns Output'] = self.flatten(state[:])

    ret['Round ' + str(r) + ': RoundKey'] = self.flatten(self.ks[r])

    state = [state[i] ^ self.ks[r][i] for i in range(16)]
    ret['Round ' + str(r) + ': AddRoundKey Output'] = self.flatten(state[:])

state = subbytes(state)
ret['Round 10: SubBytes Output'] = self.flatten(state[:])
```

Dumb Method: Step #2

Amazon EC2 Pricing

With On-Demand instances you only pay for EC2 instances you use. The use of On-Demand instances frees you from the costs and complexities of planning, purchasing, and maintaining hardware and transforms what are commonly large fixed costs into much smaller variable costs.

Manage Your AWS Resources

Enter EC2 Console

Compute Optimized - Current Generation

c5.large	2	9	4 GiB	EBS Only	\$0.093 per Hour
c5.xlarge	4	17	8 GiB	EBS Only	\$0.186 per Hour
c5.2xlarge	8	34	16 GiB	EBS Only	\$0.372 per Hour
c5.4xlarge	16	68	32 GiB	EBS Only	\$0.744 per Hour
c5.9xlarge	36	141	72 GiB	EBS Only	\$1.674 per Hour
c5.18xlarge	72	281	144 GiB	EBS Only	\$3.348 per Hour
c5d.large	2	9	4 GiB	1 x 50 NVMe SSD	\$0.106 per Hour
c5d.xlarge	4	17	8 GiB	1 x 100 NVMe SSD	\$0.212 per Hour
c5d.2xlarge	8	34	16 GiB	1 x 200 NVMe SSD	\$0.424 per Hour
c5d.4xlarge	16	68	32 GiB	1 x 400 NVMe SSD	\$0.848 per Hour
c5d.9xlarge	36	141	72 GiB	1 x 900 NVMe SSD	\$1.908 per Hour
c5d.18xlarge	72	281	144 GiB	1 x 1800 NVMe SSD	\$3.816 per Hour
c4.large	2	8	3.75 GiB	EBS Only	\$0.11 per Hour
c4.xlarge	4	16	7.5 GiB	EBS Only	\$0.218 per Hour
c4.2xlarge	8	31	15 GiB	EBS Only	\$0.438 per Hour
c4.4xlarge	16	62	30 GiB	EBS Only	\$0.876 per Hour
c4.8xlarge	36	132	60 GiB	EBS Only	\$1.75 per Hour

How do you know she is a witch?

Remember these?

Most micros not specifically designed as secure element don't talk about power analysis.

But *maybe* you actually care about side-channel resistance?

Not *so much* you are going to spend \$\$\$ getting a validated crypto core.

What to do???

SAM L10 and L11 Microcontroller Family

Industry's Lowest Power 32-bit MCUs, First to Offer Chip-Level! Arm® TrustZone® Technology

With the increasing growth of IoT end points and, consequently, the increased frequency of security power consumption while adding robust security. The SAM L10 and L11 MCU family takes an innovative variety of peripherals, including security features, into the industry's lowest power MCU in its class. 1 battery constraints of less power-efficient MCUs. These MCUs run at 32 MHz with memory configurable variant options: SAM L10 and SAM L11 and boast ultra-low power consumption as well as an enhanced SAM L11 variant adds integrated hardware security. They both come in 24- and 32-pin packages for capacitive touch and general purpose embedded control applications.

Robust Security

SAM L11 MCUs integrate Arm® TrustZone® security and ARM® TrustZone® security protect from both physical and logical attacks. They are supported by a security isolation framework implementation of security. A SAM L11 provides strong resistance to software attacks through its reliability and avoidance of vulnerabilities.

STM32F405/415

The STM32F405/415 lines are designed for medical, industrial, performance, embedded memories and rich peripheral performance. The STM32F405/415 offers the full performance of the STM32F405/415.

Performance: At 168 MHz, the STM32F405/415 delivers states using ST's ART Accelerator. The DSP instruction set.

Power efficiency: ST's 90 nm process, ART Accelerator and the dynamic power executing from Flash memory to be as low as 238 µA/MHz at 168 MHz.

Rich connectivity: Superior and innovative peripherals

- 2x USB OTG (one with HS support)
- Audio: dedicated audio PLL and 2 full duplex I2S
- Up to 15 communication interfaces (including 6x USARTs running at up to 10.5 Mbit/s, 3x I2C, 2x CAN, SDIO)
- Analog: two 12-bit DACs, three 12-bit ADCs reaching 2.4 MSPS or 7.2 MSPS in
- Up to 17 timers: 16- and 32-bit running at up to 168 MHz
- Easily extendable memory range using the flexible static memory controller support
- Analog true random number generator
- The STM32F415 also integrates a cryptohash processor providing hardware acceleration

Summary

Atmel's SAM4L series is a member of a family of Flash microcontrollers based on the high performance 32-bit ARM Cortex-M4 RISC processor running at frequencies up to 48 MHz.

The SAM4L series embeds state-of-the-art picoPower technology for ultra-low power consumption. Combined power control techniques are used to bring active current consumption down to 90 µA/MHz. The device allows a wide range of options between functionality and power consumption, giving the user the ability to reach the lowest possible power consumption with the feature set required for the application. The WAIT and RETENTION modes provide full logic and RAM retention, associated with fast wake-up capability (<1.5 µs) and a very low consumption of, respectively, 3 µA and 1.5 µA. In addition, WAIT mode supports SleepWalking features. In BACKUP mode, CPU, peripherals and RAM are powered off and, while consuming less than 8.5 µA with external interrupt wake-up supported.

The SAM4L series offers a wide range of peripherals such as segment LCD controller, embedded hardware capacitive touch (QTouch), USB device & embedded host, 128-bit AES and audio interfaces in addition to high speed serial peripherals such as USART, SPI and I2C. Additionally the Peripheral Event System and SleepWakeup each other and make a qualified events or threshold.

Features

AVR1318: Using the XMEGA built accelerator

Features

- Full compliance with AES (FIPS Publication 197, 2002)
 - Both encryption and decryption procedures
- 128-bit Key and State memory
- XOR load option to State memory useful for cipher block coding
- Sequential access to State and Key memories
- Optional Interrupt- and DMA request on AES complete

1 Introduction

The XMEGA™ AES Crypto Module supports the Advanced Encryption Standard (AES), and can perform encryption and decryption. The module supports a key length of 128 bits. The 128-bit key block and 128-bit data block (plaintext or ciphertext) must be loaded into the Key and State memory in the AES Crypto

Kinetis K24F Sub-Family Data Sheet

120 MHz ARM® Cortex®-M4-based Microcontroller with FPU

The K24 product family members are optimized for cost-sensitive applications requiring low-power, USB connectivity, and up to 256 KB of embedded SRAM. These devices share the comprehensive enablement and scalability of the Kinetis family.

This product offers:

- Run power consumption down to 250 µA/MHz. Static power consumption down to 5.8 µA with full state retention and 5 µs wakeup. Lowest Static mode down to 339 nA
- USB LS/FS OTG 2.0 with embedded 3.3 V, 120 mA LDO Vreg, with USB device crystal-less operation

Performance

- Up to 120 MHz ARM® Cortex®-M4 core with DSP instructions and floating point unit

Memories and memory interfaces

- Up to 1 MB program flash memory and 256 KB RAM
- FlexBus external bus interface

System peripherals

- Multiple low-power modes, low-leakage wake-up unit
- Memory protection unit with multi-master protection
- 16-channel DMA controller
- External watchdog monitor and software watchdog

Security and integrity modules

- Hardware CRC module
- Hardware random-number generator
- Hardware encryption supporting DES, 3DES, AES, MD5, SHA-1, and SHA-256 algorithms
- 128-bit unique identification (ID) number per chip

Analog modules

Appl

MK24FN1M0VLO12
MK24FN1M0VLL12
MK24FN1M0VDC12

121 XFBGA
8 x 8 x 0.5 mm Pitch 20 x 20 x 1.6 mm
0.65 mm

144 LQFP
20 x 20 x 1.6 mm
0.65 mm

100 QFP
14 x 14 x 1.7 mm Pitch 0.5 mm

Communication interfaces

- USB full-speed On-the-Go controller
- Controller Area Network (CAN) module
- Three SPI modules
- Three I2C modules. Support for up to 1 Mbit/s
- Six UART modules
- Secure Digital Host Controller (SDHC)
- I2S module

Timers

- Two 8-channel FlexTimers (PWM/Motor con)
- Two 2-channel FlexTimers (PWM/Quad deco)
- 32-bit PITs and 16-bit low-power timers
- Real-time clock
- Programmable delay block

Clocks

- 3 to 32 MHz and 32 kHz crystal oscillator
- PLL, FLL, and multiple internal oscillators
- 48 MHz Internal Reference Clock (IRC48M)

Operating Characteristics

Test Vector Leakage Assessment (TVLA)

- Based on proposal “A testing methodology for sidechannel resistance validation” by *Gilbert Goodwill, Benjamin Jun, Josh Jaffe, and Pankaj Rohatgi of Cryptography Research Inc.*



Welch's t-test

You have *observations* of two random variables (the test situations).
How confident are you they have equal means?

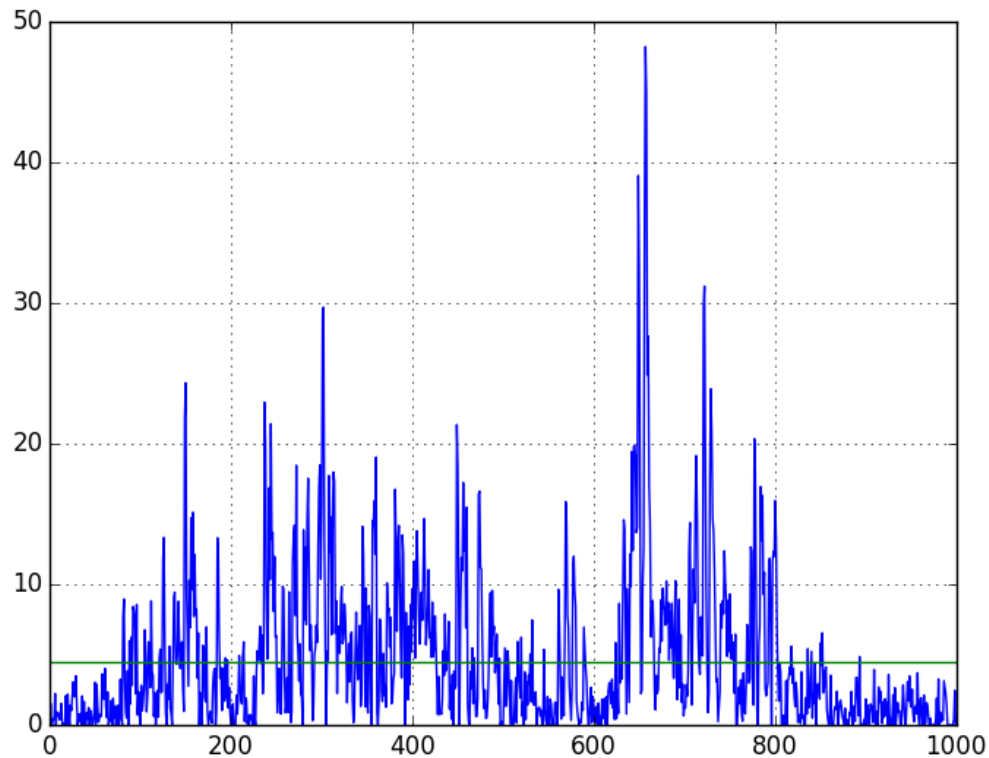
If they had un-equal means, you could form a probability that a given measurement (observation of encryption) came from one set vs. the other.

$$t = \frac{\overline{X}_1 - \overline{X}_2}{\sqrt{\frac{s_1^2}{N_1} + \frac{s_2^2}{N_2}}}$$

T-Test Results & Expansions

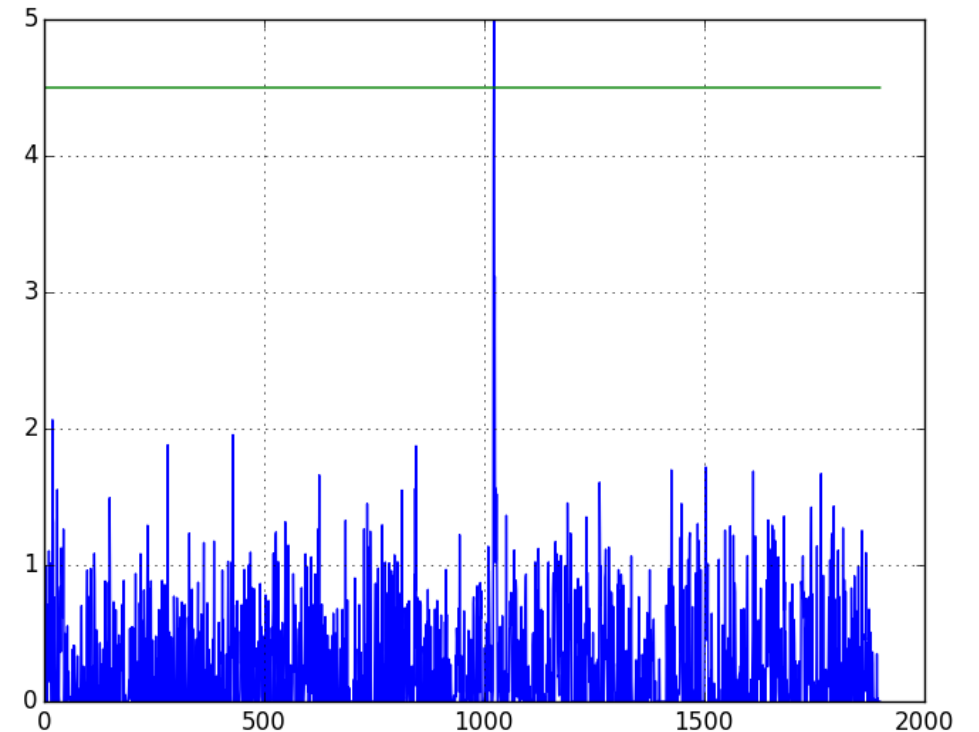
Test 1: Test 0: Fixed/Random Plaintext

Maximum t: 48.228980 @ 657 [FAIL]



Test 65: HW: Round 5: SubBytes Output byte 0

Maximum t: 4.998180 @ 1022 [FAIL]





ChipWhisperer-Lint Objectives

- Provide method of running pass/fail security tests.
- Provide information regarding *specific leakage model in use*.
- Make simple method of automatically running these tests.

Examples of Hardware Crypto

Examples: Running on some Micros

Chosen five “representative” samples of micros you might use for various tasks. Specifically:

- Mix of application-specific, general-purpose.
- NONE of these are “secure” devices marked for credit cards, content delivery, etc.

Disclosure:

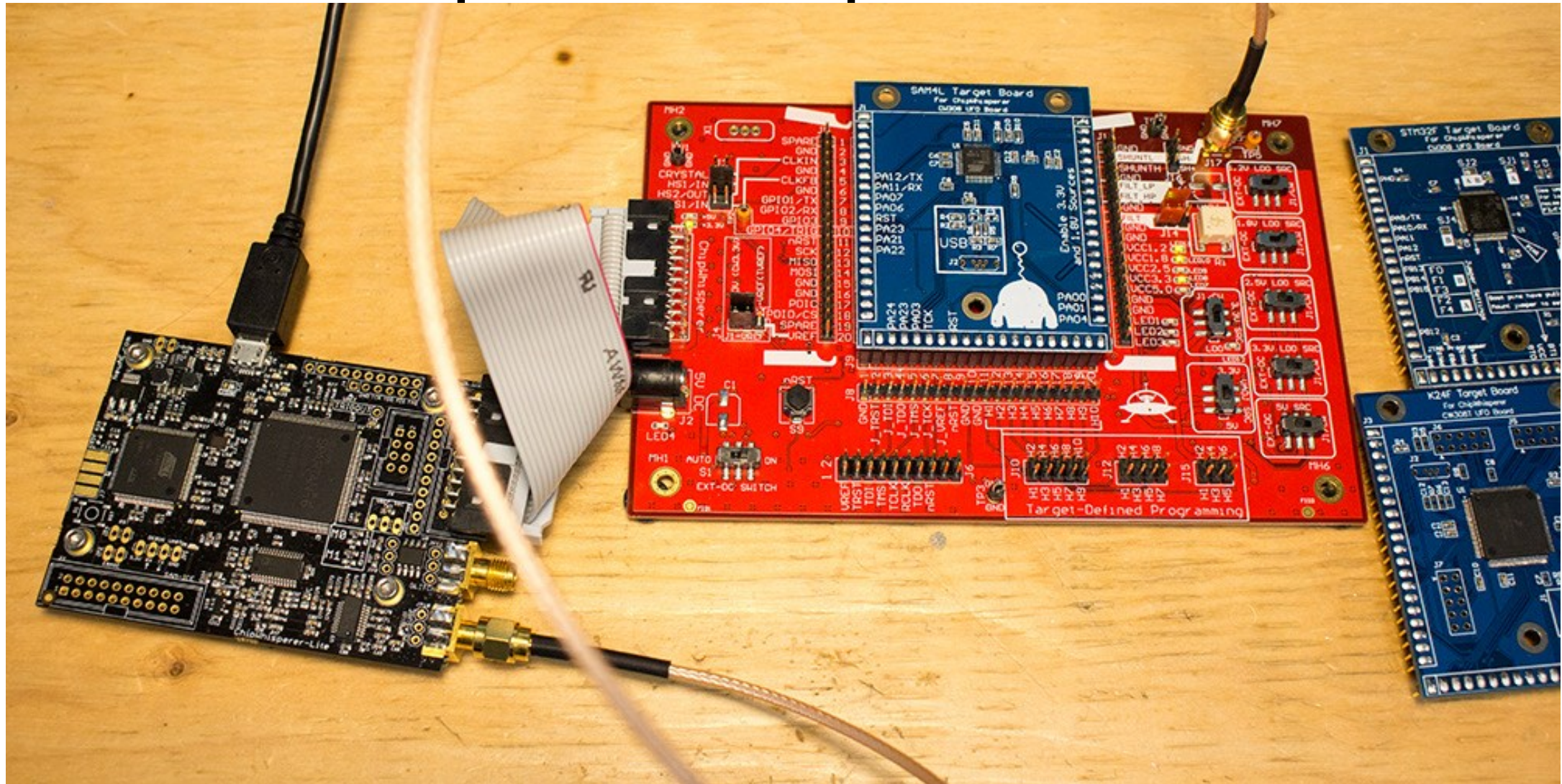
- This is NOT an “attack” specific to the micros – so nothing is more broken now than it was a week ago. Rather this is a generic weakness across ALL similar devices. The “attack” only exists when people mis-use the crypto.
- Vendors have been contacted or I’ve attempted to contact them (only last week as I was running late, sorry about that vendors).




Target Devices

- ST STM32F415 (Arm Cortex-M4 Core)
- NXP Kinetis K24 (Arm Cortex-M4 Core)
- Espressif ESP32 (Tensilica Xtensa LX6 Core)
- Atmel SAM4L (Arm Cortex-M4 Core)

General Capture Setup



STM32F415 – Hardware Crypto

 **life.augmented**

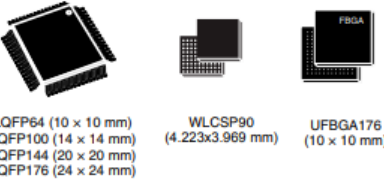
STM32F415xx
STM32F417xx

ARM Cortex-M4 32b MCU+FPU, 210DMIPS, up to 1MB Flash/192+4KB RAM, crypto, USB OTG HS/FS, Ethernet, 17 TIMs, 3 ADCs, 15 comm. interfaces & camera

Datasheet - production data

Features

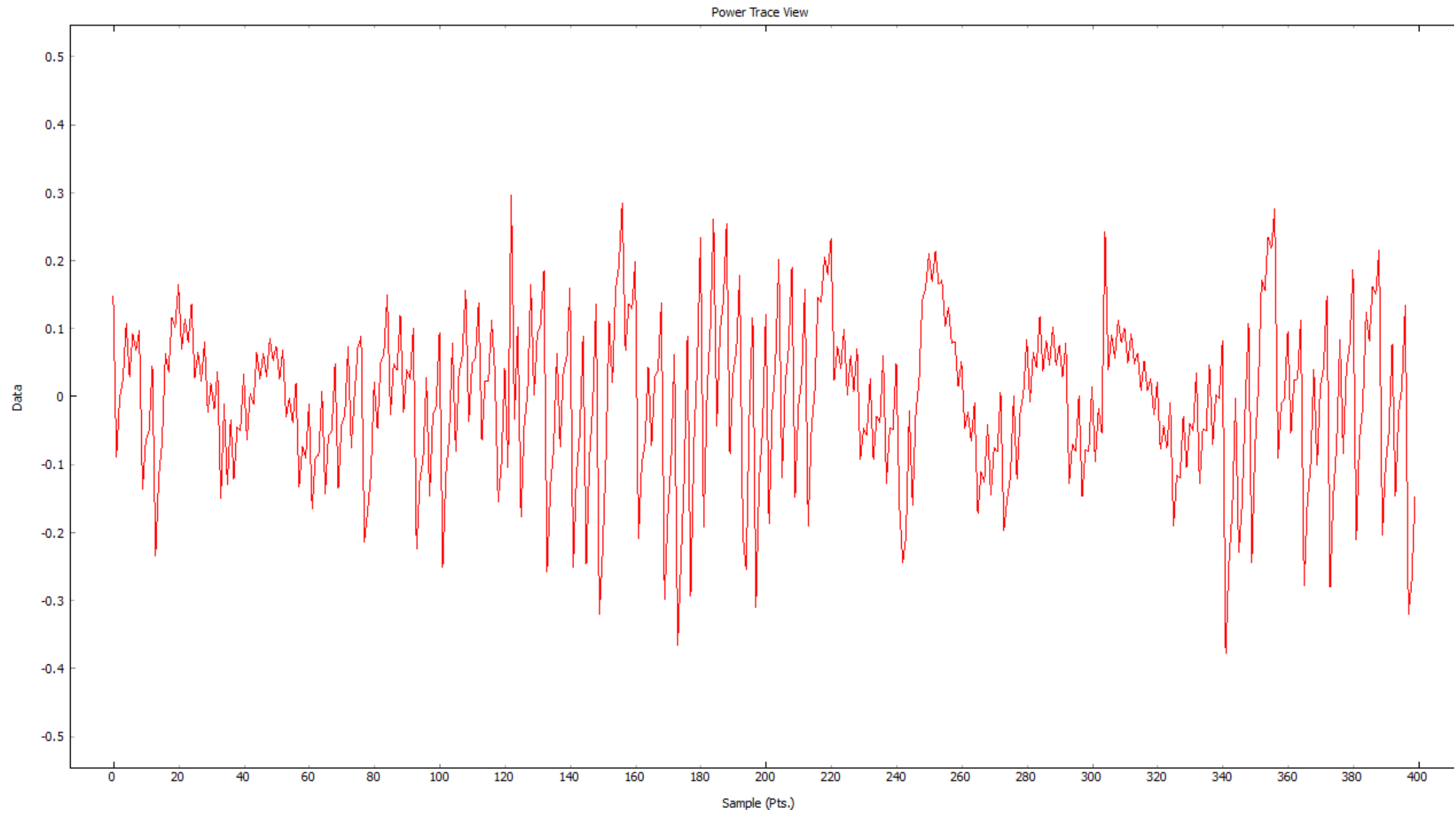
- Core: ARM® 32-bit Cortex®-M4 CPU with FPU, Adaptive real-time accelerator (ART Accelerator™) allowing 0-wait state execution from Flash memory, frequency up to 168 MHz, memory protection unit, 210 DMIPS/1.25 DMIPS/MHz (Dhrystone 2.1), and DSP instructions
- Memories
 - Up to 1 Mbyte of Flash memory
 - Up to 192+4 Kbytes of SRAM including 64-Kbyte of CCM (core coupled memory) data RAM
- Flexible static memory controller supporting Compact Flash, SRAM, PSRAM, NOR and NAND memories
- LCD parallel interface, 8080/6800 modes
- Clock, reset and supply management
 - 1.8 V to 3.6 V application supply and I/Os
 - POR, PDR, PVD and BOR
 - 4-to-26 MHz crystal oscillator
 - Internal 16 MHz factory-trimmed RC (1% accuracy)
 - 32 kHz oscillator for RTC with calibration
 - Internal 32 kHz RC with calibration
- Low-power operation
 - Sleep, Stop and Standby modes



LQFP64 (10 × 10 mm)
LQFP100 (14 × 14 mm)
LQFP144 (20 × 20 mm)
LQFP176 (24 × 24 mm)
WLCSP90 (4.223x3.969 mm)
UFBGA176 (10 × 10 mm)

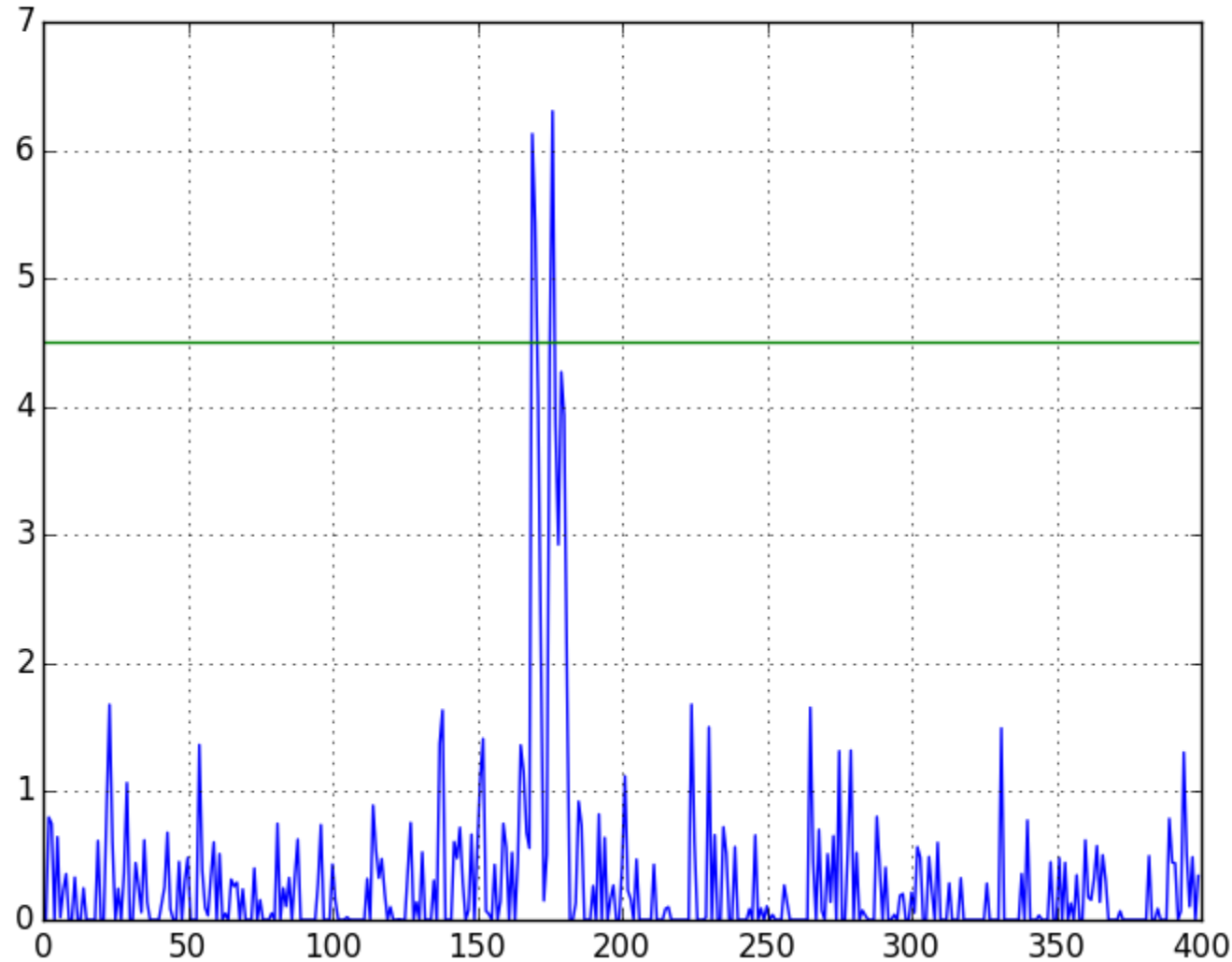
- Up to 17 timers: up to twelve 16-bit and two 32-bit timers up to 168 MHz, each with up to 4 IC/OC/PWM or pulse counter and quadrature (incremental) encoder input
- Debug mode
 - Serial wire debug (SWD) & JTAG interfaces
 - Cortex-M4 Embedded Trace Macrocell™
- Up to 140 I/O ports with interrupt capability
 - Up to 136 fast I/Os up to 84 MHz
 - Up to 138 5 V-tolerant I/Os
- Up to 15 communication interfaces
 - Up to 3 × I²C interfaces (SMBus/PMBus)
 - Up to 4 USARTs/2 UARTs (10.5 Mbit/s, ISO 7816 interface, LIN, IrDA, modem control)
 - Up to 3 SPIs (42 Mbits/s), 2 with muxed full-duplex I²S to achieve audio class accuracy via internal audio PLL or external

- Hardware AES engine.
- Relatively powerful Arm Cortex M4 core at 168 MHz.
- Lots of I/O (USB, CAN, etc).



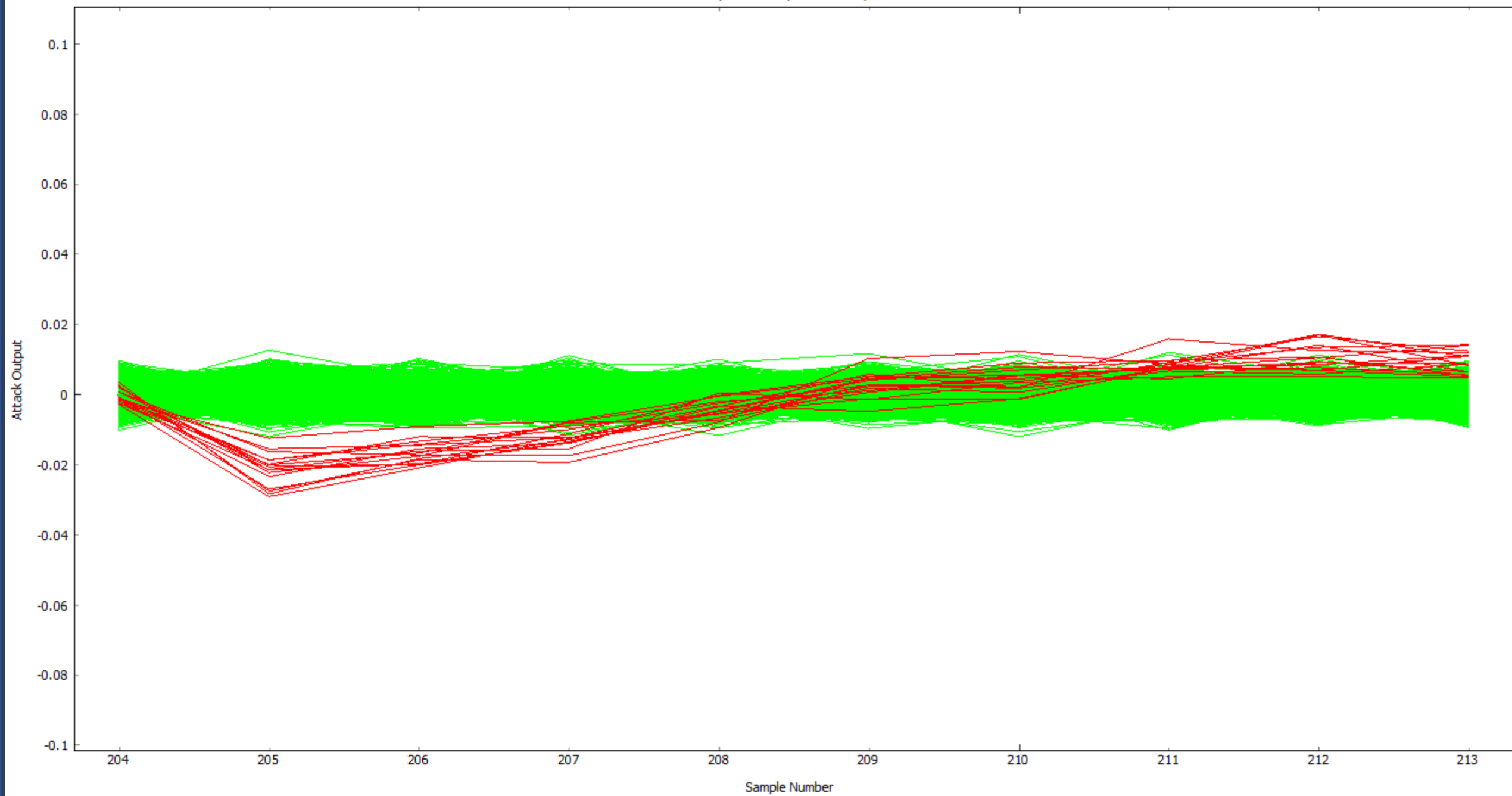
Test 88: HD: Round 0: AddRoundKey Output to Round 1: AddRoundKey Output

Maximum t: 6.304197 @ 176 [FAIL]



**AES State to State
Hamming Distance
Leakage**

Attack Output vs. Sample for Subkey Guesses



STM32F415 Summary

- Standard AES State to State Register Leakage (Hamming distance)
- Narrow window needed around leaking point (here around point 205 – this is last round attack)
- Break is ~6K traces

Kinetis K24

NXP Semiconductors
Data Sheet: Technical Data

K24P144M120SF5
Rev. 7, 11/2016



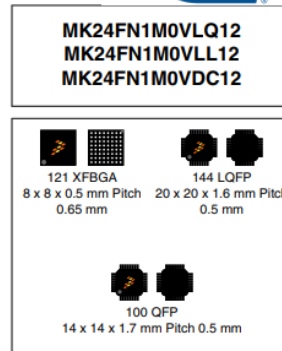
Kinetis K24F Sub-Family Data Sheet

120 MHz ARM® Cortex®-M4-based Microcontroller with FPU

The K24 product family members are optimized for cost-sensitive applications requiring low-power, USB connectivity, and up to 256 KB of embedded SRAM. These devices share the comprehensive enablement and scalability of the Kinetis family.

This product offers:

- Run power consumption down to 250 µA/MHz. Static power consumption down to 5.8 µA with full state retention and 5 µs wakeup. Lowest Static mode down to 339 nA
- USB LS/FS OTG 2.0 with embedded 3.3 V, 120 mA LDO Vreg, with USB device crystal-less operation



Performance

- Up to 120 MHz ARM® Cortex®-M4 core with DSP instructions and floating point unit

Memories and memory interfaces

- Up to 1 MB program flash memory and 256 KB RAM
- FlexBus external bus interface

System peripherals

- Multiple low-power modes, low-leakage wake-up unit
- Memory protection unit with multi-master protection
- 16-channel DMA controller
- External watchdog monitor and software watchdog

Security and integrity modules

- Hardware CRC module
- Hardware random-number generator
- Hardware encryption supporting DES, 3DES, AES, MD5, SHA-1, and SHA-256 algorithms
- 128-bit unique identification (ID) number per chip

Analog modules

- Two 16-bit SAR ADCs
- Two 12-bit DACs
- Three analog comparators (CMP)
- Voltage reference

Communication interfaces

- USB full-/low-speed On-the-Go controller
- Controller Area Network (CAN) module
- Three SPI modules
- Three I2C modules. Support for up to 1 Mbit/s
- Six UART modules
- Secure Digital Host Controller (SDHC)
- I2S module

Timers

- Two 8-channel Flex-Timers (PWM/Motor control)
- Two 2-channel FlexTimers (PWM/Quad decoder)
- 32-bit PITs and 16-bit low-power timers
- Real-time clock
- Programmable delay block

Clocks

- 3 to 32 MHz and 32 kHz crystal oscillator
- PLL, FLL, and multiple internal oscillators
- 48 MHz Internal Reference Clock (IRC48M)

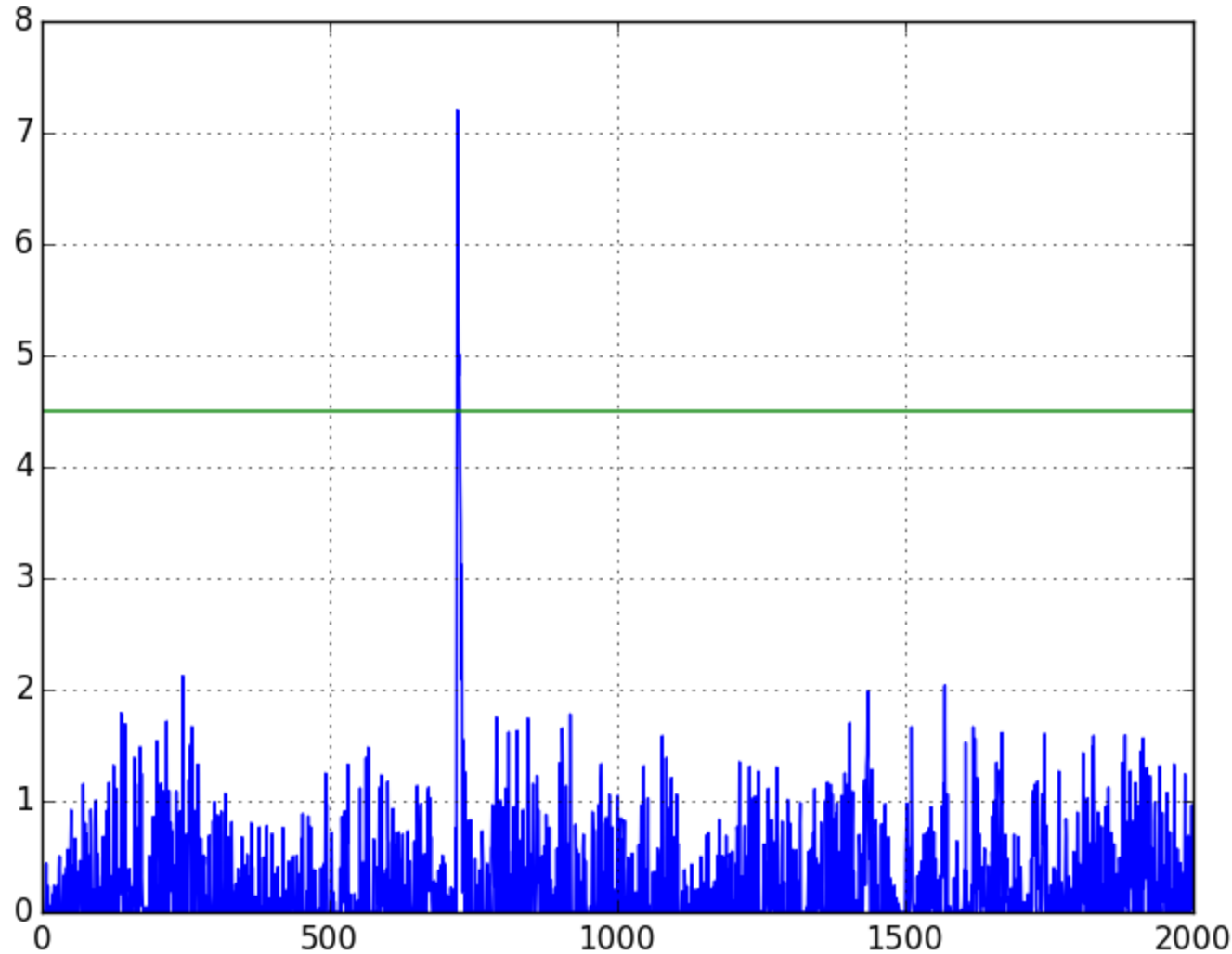
Operating Characteristics

- Voltage range: 1.71 to 3.6 V
- Flash write voltage range: 1.71 to 3.6 V
- Temperature range (ambient): -40 to 105°C

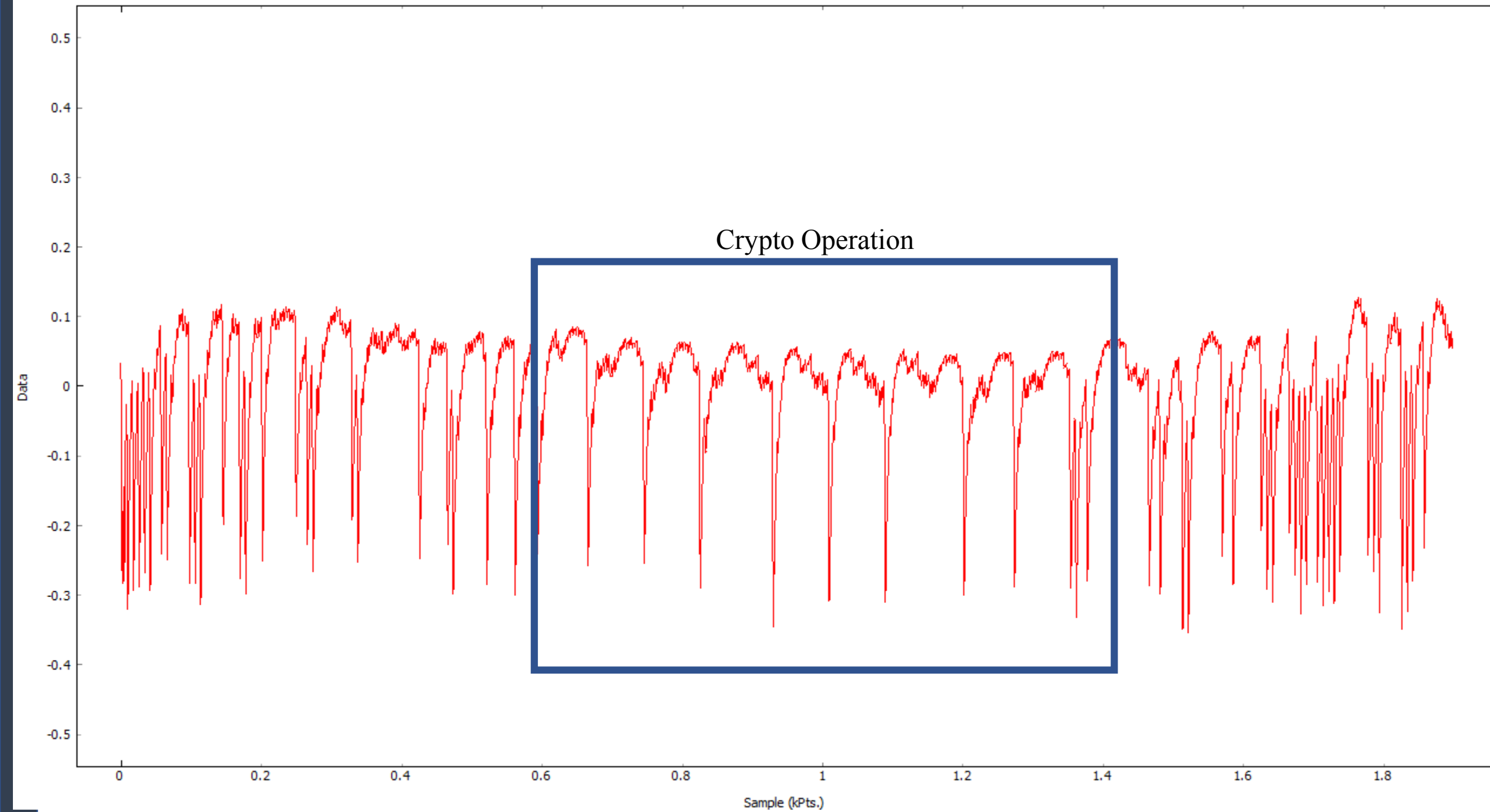
- 120 MHz Cortex-M4
- Hardware AES peripheral (also DES, SHA).

Test 127: HD: Round 1: SubBytes Output to Round 1: AddRoundKey Output

Maximum t: 7.205168 @ 722 [FAIL]



**Hamming Distance
leakage on S-Box
input to output**





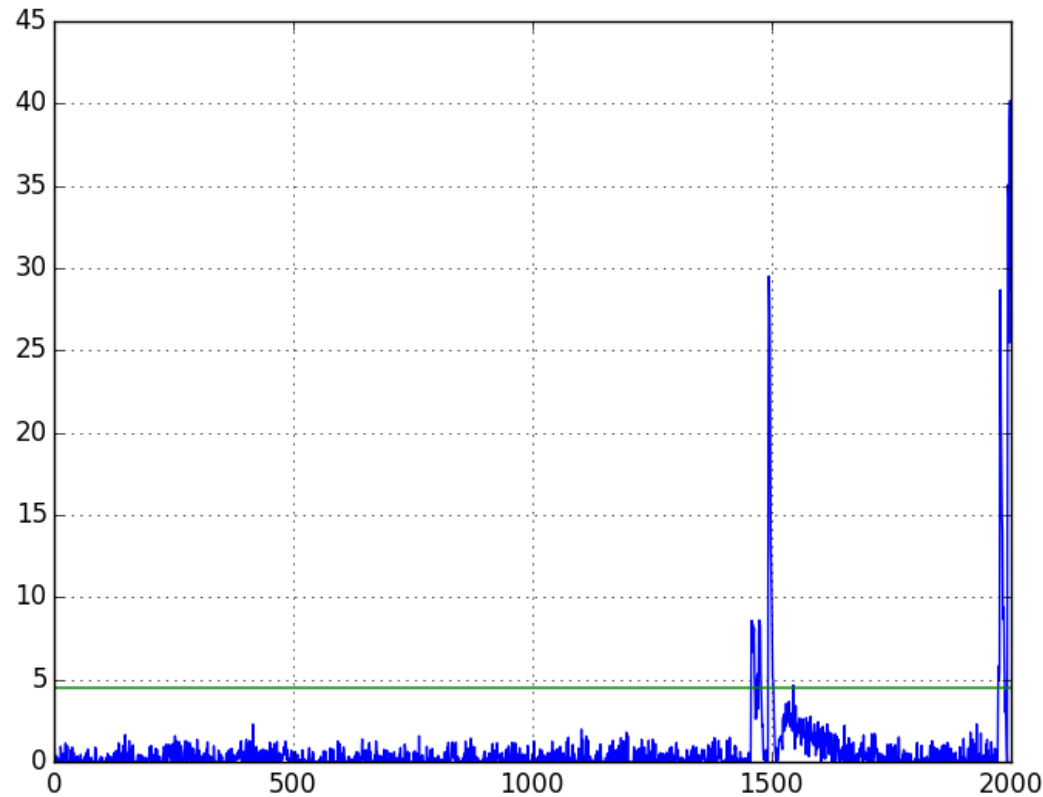
K24F Summary

- Interesting Sbox Input to Output Leakage
- Appears to process in 32-bit sections, possibly T-Table or other implementation.
- Break is $\sim 14\text{K}$ traces.

BONUS: Catching library leakages (either in examples or even ROM code)

Test 81: HD: Key to Round 10: SubBytes Output

Maximum t: 40.174129 @ 1998 [FAIL]



Espressif ESP32

1. OVERVIEW

1.6 Block Diagram

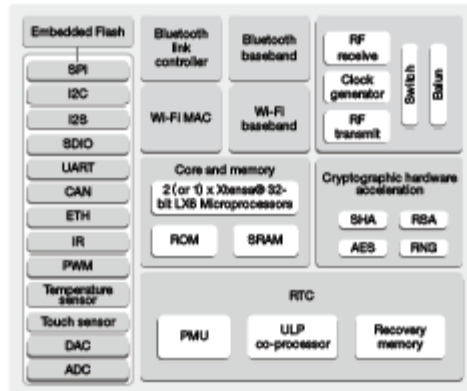
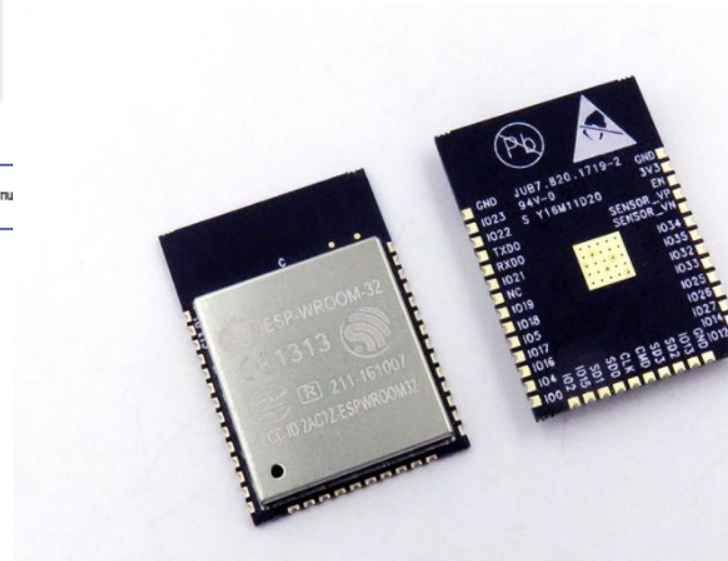


Figure 1: Function Block Diagram

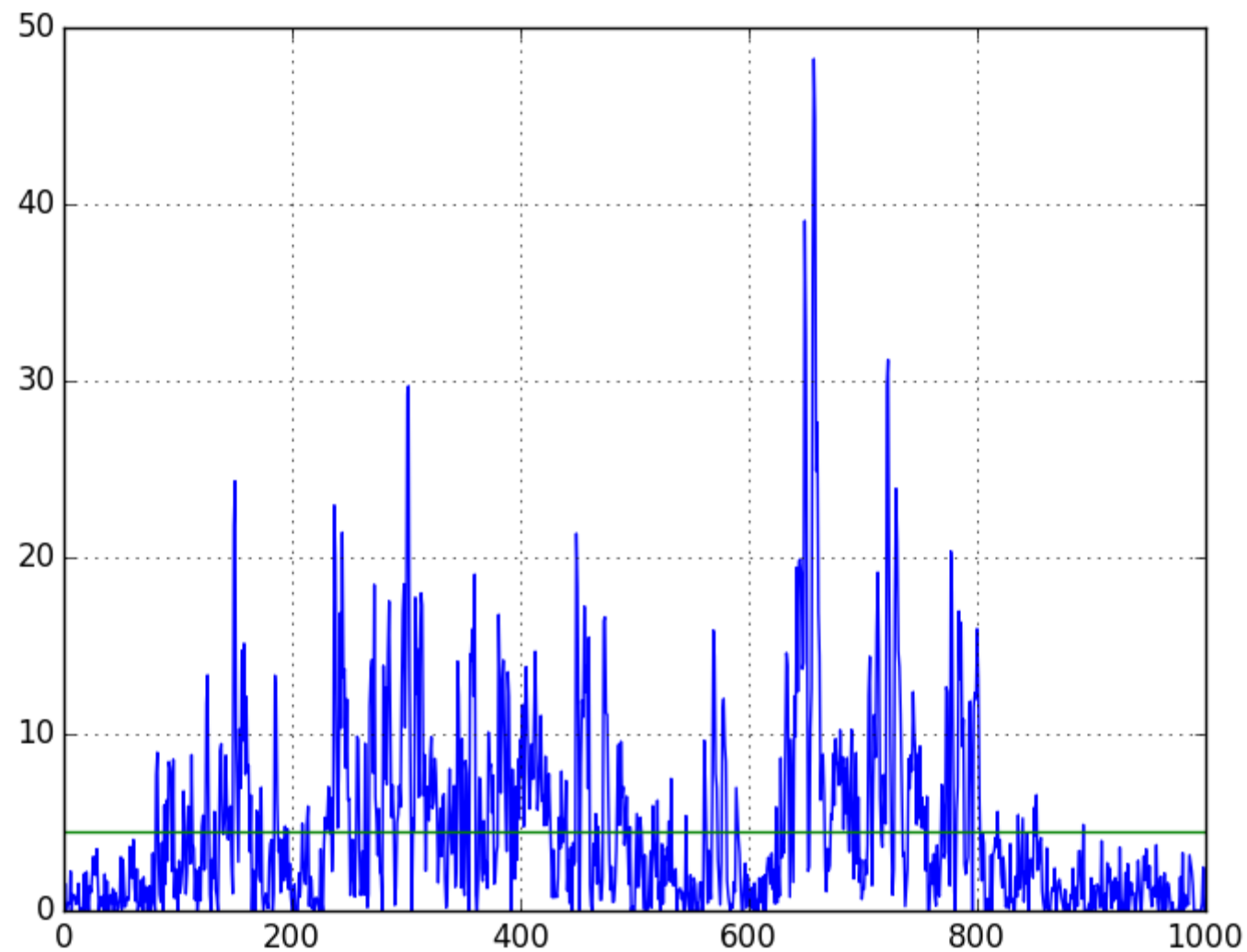
Note:
Products in the ESP32 series differ from each other in terms of their support for embedded flash and the number they have. For details, please refer to [Part Number and Ordering Information](#).



- Single or dual core device.
- Super-low cost device targets IoT.
- AES accelerator, also used to encrypt external SPI flash.

Test 1: Test 0: Fixed/Random Plaintext

Maximum t: 48.228980 @ 657 [FAIL]



ESP32 Summary

- AES-128 hardware accelerator is leaky, breakable with CPA.
- Leakage Model: HW S-Box Output, approx. 18K traces needed.
- Briefly looked at usage during boot – not immediately successful. More research needed to understand when decryption core is used.

Atmel Microchip SAM4L

Summary

Atmel's SAM4L series is a member of a family of Flash microcontrollers based on the high performance 32-bit ARM Cortex-M4 RISC processor running at frequencies up to 48MHz.

The SAM4L series embeds state-of-the-art picoPower technology for ultra-low power consumption. Combined power control techniques are used to bring active current consumption down to 90µA/MHz. The device allows a wide range of options between functionality and power consumption, giving the user the ability to reach the lowest possible power consumption with the feature set required for the application. The WAIT and RETENTION modes provide full logic and RAM retention, associated with fast wake-up capability (<1.5µs) and a very low consumption of, respectively, 3 µA and 1.5 µA. In addition, WAIT mode supports SleepWalking features. In BACKUP mode, CPU, peripherals and RAM are powered off and, while consuming less than 0.9µA with external interrupt wake-up supported.

The SAM4L series offers a wide range of peripherals such as segment LCD controller, embedded hardware capacitive touch (QTouch), USB device & embedded host, 128-bit AES and audio interfaces in addition to high speed serial peripherals such as USART, SPI and I²C. Additionally the Peripheral Event System and SleepWalking allows the peripherals to communicate directly with each other and make intelligent decisions and decide to wake-up the system on a qualified events on a peripheral level; such as I²C address match or and ADC threshold.

Features

- Core
 - ARM® Cortex™-M4 running at up to 48MHz
 - Memory Protection Unit (MPU)
 - Thumb®.2 instruction set
- picoPower® Technology for Ultra-low Power Consumption
 - Active mode down to 90µA/MHz with configurable voltage scaling
 - High performance and efficiency: 28 coremark/mA
 - Wait mode down to 3µA with fast wake-up time (<1.5µs) supporting SleepWalking



ATSAM
ARM-based
Flash MCU

SAM4L Series

- Hardware AES Engine....
With “countermeasures”
- Very low-power, Arm Cortex M4 @ 48 MHz.
- USB support.

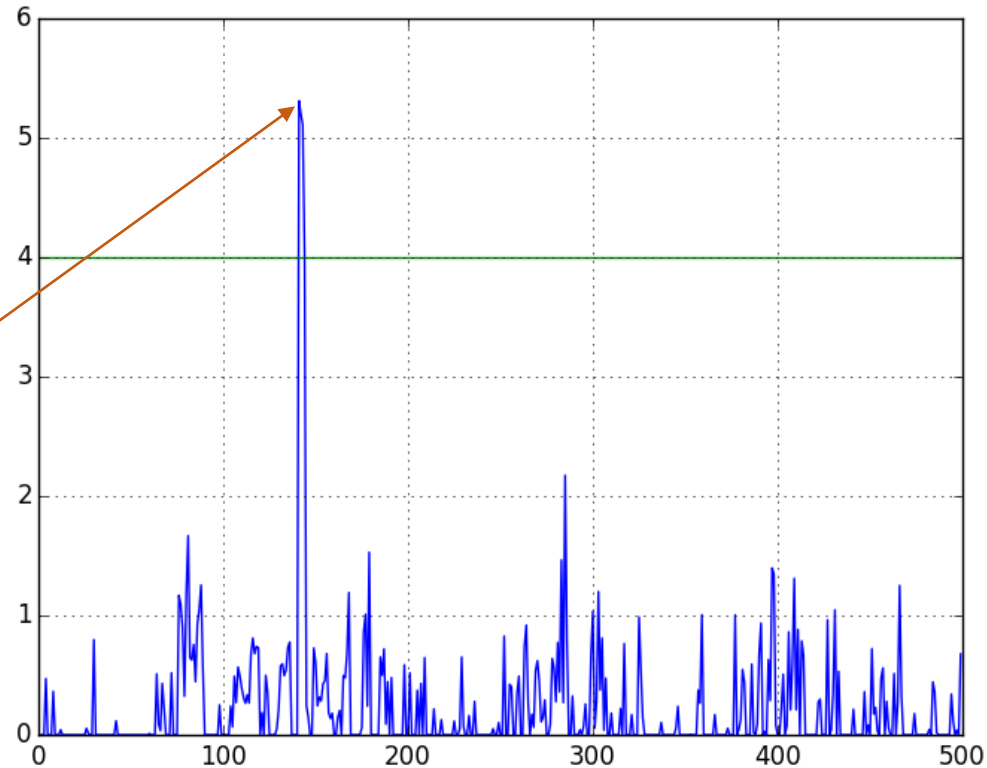
Countermeasures On (incorrectly configured)

Test 55: HD: Round 2: AddRoundKey Output to Round 3: AddRoundKey Output

Maximum t: 5.310029 @ 141 [FAIL]

Round to round leakage

T-Test maximum



Bonus: Detecting Datasheet Bugs

- Datasheet is missing critical configuration information that results in countermeasures not being effective.
- Based on feedback from Atmel these changes were made to software and tests re-run.

18.4.5 Security Features

18.4.5.1 Hardware Countermeasures Against Differential Power Analysis Attacks

AESA features four types of hardware countermeasures that are useful for protecting data against differential power analysis attacks:

- Type 1: Randomly add one cycle to data processing
- Type 2: Randomly add one cycle to data processing (other version)
- Type 3: Add a random number of clock cycles to data processing, subject to a maximum of 11 clock cycles for key size of 128 bits
- Type 4: Add random spurious power consumption during data processing

By default, all countermeasures are enabled. One or more of the countermeasures can be disabled by programming the Countermeasure Type (CTYPE) field in the MODE register.

The countermeasures use random numbers generated by a deterministic random number generator embedded in AESA. The seed for the random number generator is written to the DRNGSEED register. Note that access to the DRNGSEED register is by 32-bit words only (i.e., no halfword or byte access). Note also that a new seed must be written after a change in the key size.

Note that enabling countermeasures reduces AESA's throughput. In short, the throughput is highest with all the countermeasures disabled. On the other hand, with all of the countermeasures enabled, the best protection is achieved but the throughput is worst.

Specific error: DRNGSEED MUST be written as part of initialization. Datasheet only references setting this after changing key size (default is 128-bit so by default no change occurs), and otherwise claims countermeasures are always on unless explicitly disabled.

~~Atmel~~ Microchip SAM4L Summary

- Some of the countermeasures ineffective ('add noise' does not really do anything for example).
- CPA attack effective in ~ 3000 traces.
- Jitter stops CPA from working.
 - Simple resync insufficient to fix \rightarrow either random state included or more work needed.

Summary

Device	AES-128 Cycles*	Target Leakage	CPA Traces	Notes
STM32F415	493	Round to Round HD	~6 000	
Kinetis K24F	475	S-Box Input to Output HD	~14 000	
ESP32	252	S-Box Output HW	~18 000	Calling ROM function directly.
SAM4L	81	Round to Round HD	~3 000	Jitter countermeasures DISABLED.

*Includes overhead of calling crypto library function + setup for single block. NOT speed-optimized.

All devices: Measurement with ChipWhisperer-Lite Capture, 7.37 MHz, 29.48 MS/s.

How to Care

- Use SECURE device if you need side-channel power analysis protection. Ensure it includes ratings based on relevant industry standard.
- Example: Common Criteria rated secure element or microcontroller.

Examples of Checking Software Libraries

Getting Started

- You'll need to configure an example target to provide encryption.
 - Lot of examples in ChipWhisperer repo.
 - Additional examples on ChipWhisperer-Lint repo showing complete attack

ChipWhisperer Lint Detection on mbed TLS Library

This demo is designed to run AES using the mbed TLS library against a number of ARM targets. This demonstrates the power of automated analysis in determining where leakage exists, and why testing against ALL possible variants is useful for detecting leakage that might go unnoticed on specific builds/variants.

The setup involves (a) building possible variants of the library with different compilers, library options, and compiler flags, (b) running the target code on different physical architectures, and (c) performing automated side-channel analysis of the resulting power traces.

These are provided by:

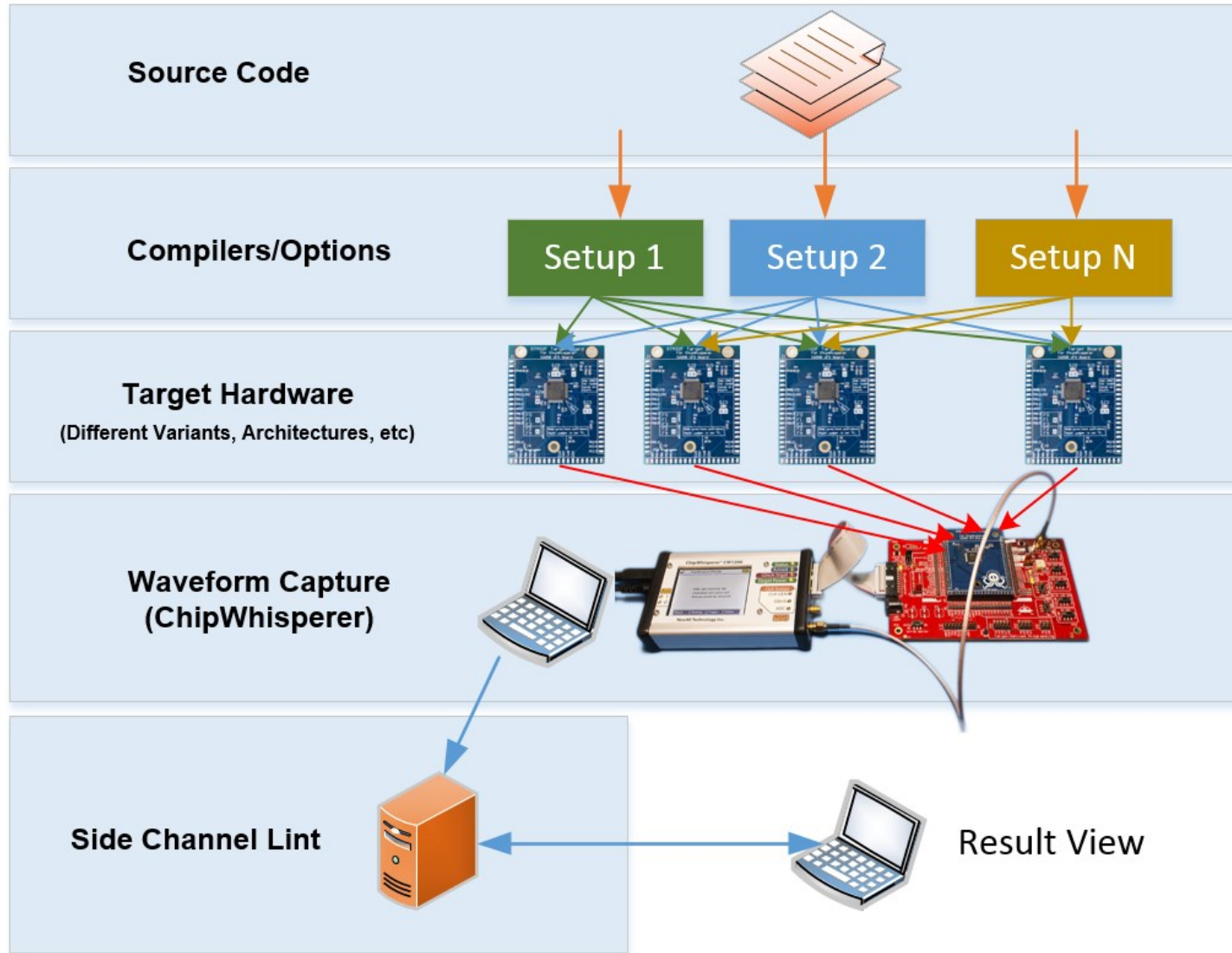
(a) A special build script (would be specific to your build environment). (b) The ChipWhisperer-Capture software connected to a ChipWhisperer-Lite + UFO Board, run in a basic script. (c) The ChipWhisperer-Lint software.

(a) Build Process

The target is the mbed TLS library. The automated test script performs the following actions:

1. Perform "git pull" to get latest code from public repository.
2. For each variant:
 - Autogenerate makefile to build binary (based on settings below).

Further Example - C.I. Test Suite



Source code is crypto library under test (for SW), but can be IP core for FPGA tests.

Various binaries generated – library is compiled for supported platforms and with various options a user might enable.

Binaries loaded onto test platform (for example, based on UFO board), but can also use existing development kits that have been instrumented to take power measurements.

Capture can be done with regular oscilloscope. Here ChipWhisperer hardware (open-source versions available) shown, which simplifies setup considerably.

Captured power traces analyzed by ChipWhisperer-Lint. Can run a local server or use more powerful cloud style server.

Side-Channel Lint Test Report

Report generated at 2017-07-07 16:29:04.026000

Test Results

1. ROM vs. RAM Tables

2. Optimization Off vs. On

3. IAR vs GCC Compiler

4. STM32F0 vs STM32F1 vs STM32F2

STM32F0: GCC, OPT=0, RAM Tables

STM32F0: GCC, OPT=0, ROM Tables

STM32F0: GCC, OPT=1, RAM Tables

STM32F0: GCC, OPT=1, ROM Tables

STM32F0: IAR, OPT=0, RAM Tables

STM32F0: IAR, OPT=0, ROM Tables

STM32F0: IAR, OPT=1, RAM Tables

STM32F0: IAR, OPT=1, ROM Tables

STM32F1: GCC, OPT=0, RAM Tables

STM32F1: GCC, OPT=0, ROM Tables

STM32F1: IAR, OPT=0, RAM Tables

STM32F1: IAR, OPT=0, ROM Tables

STM32F2: GCC, OPT=0, RAM Tables

STM32F2: GCC, OPT=0, ROM Tables

STM32F2: GCC, OPT=1, RAM Tables

STM32F2: GCC, OPT=1, ROM Tables

STM32F2: IAR, OPT=0, RAM Tables

STM32F2: IAR, OPT=0, ROM Tables

STM32F2: IAR, OPT=1, RAM Tables

STM32F2: IAR, OPT=1, ROM Tables

Test Number	Test Name	STM32F0: GCC, OPT=0, RAM Tables	STM32F0: GCC, OPT=0, ROM Tables	STM32F0: GCC, OPT=1, RAM Tables	STM32F0: GCC, OPT=1, ROM Tables	STM32F0: IAR, OPT=0, RAM Tables	STM32F0: IAR, OPT=0, ROM Tables	STM32F0: IAR, OPT=1, RAM Tables	STM32F0: IAR, OPT=1, ROM Tables	STM32F1: GCC, OPT=0, RAM Tables	STM32F1: GCC, OPT=0, ROM Tables	STM32F1: IAR, OPT=0, RAM Tables	STM32F1: IAR, OPT=0, ROM Tables	STM32F2: GCC, OPT=0, RAM Tables	STM32F2: GCC, OPT=0, ROM Tables	STM32F2: GCC, OPT=1, RAM Tables	STM32F2: GCC, OPT=1, ROM Tables	STM32F2: IAR, OPT=0, RAM Tables	STM32F2: IAR, OPT=0, ROM Tables	STM32F2: IAR, OPT=1, RAM Tables	STM32F2: IAR, OPT=1, ROM Tables
-	Minimum Time	10147	10147	7147	7143	9963	9963	6699	6691	8795	4475	7055	4071	11659	17259	7283	10747	9967	6363	9559	
-	Maximum Time	10147	10147	7147	7143	9963	9963	6699	6691	8795	4475	7055	4071	11659	17259	7283	10847	9967	6363	9655	
1	HW: Plaintext	20.058	23.162	46.461	48.719	15.408	14.293	15.215	17.111	19.737	6.457	14.728	10.381	14.733	15.471	2.675	7.373	3.710	2.918	5.449	
2	HD: Plaintext to Key	15.928	16.230	14.478	13.311	18.310	18.810	18.035	18.881	16.802	19.151	15.109	13.497	9.993	14.722	2.500	5.302	4.647	5.076	3.483	
3	HD: Plaintext to Round 0: AddRoundKey Output	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	
4	HD: Plaintext to Round 1: SubBytes Output	2.681	2.623	4.164	2.105	2.363	2.606	2.056	2.538	2.258	2.094	2.667	2.584	2.589	2.605	2.734	2.243	2.532	2.098	3.767	
5	HD: Plaintext to Round 1: ShiftRows Output	2.681	2.623	4.164	2.105	2.363	2.606	2.056	2.538	2.258	2.094	2.667	2.584	2.589	2.605	2.734	2.243	2.532	2.098	3.767	
6	HW: Key	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	
7	HD: Key to Round 0: AddRoundKey Output	20.058	23.162	46.461	48.719	15.408	14.293	15.215	17.111	19.737	6.457	14.728	10.381	14.733	15.471	2.675	7.373	3.710	2.918	5.449	
8	HD: Key to Round 1: SubBytes Output	13.812	7.490	3.816	5.199	6.081	5.818	7.122	5.745	5.026	6.078	11.391	7.386	2.997	2.703	2.278	2.573	3.506	2.698	2.412	
9	HD: Key to Round 1: ShiftRows Output	13.812	7.490	3.816	5.199	6.081	5.818	7.122	5.745	5.026	6.078	11.391	7.386	2.997	2.703	2.278	2.573	3.506	2.698	2.412	

000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
457	14.728	10.381	14.733	15.471	2.675	7.373	3.710	2.918	5.449
078	11.391	7.386	2.997	2.703	2.278	2.573	3.506	2.698	2.412
078	11.391	7.386	2.997	2.703	2.278	2.573	3.506	2.698	2.412
714	3.090	2.824	2.337	7.004	2.322	4.856	2.269	2.452	6.957
151	15.109	13.497	9.993	14.722	2.500	5.302	4.647	5.076	3.483

Leakage based on certain tests may not be present on specific configurations/hardware.

Testing on a single device is insufficient, even when devices are of similar family (in this case all ARM, STM32F).

RAM Tables	RAM Tables	RAM Tables	RAM Tables	RAM Tables
STM32F2: GCC, OPT=0, ROM Tables	STM32F2: GCC, OPT=s, RAM Tables	STM32F2: GCC, OPT=s, RAM Tables	STM32F2: IAR, OPT=0, RAM Tables	STM32F2: IAR, OPT=s, RAM Tables
7283	10747	9967	6363	9559
7283	10847	9967	6363	9655
2.675	7.373	3.710	2.918	5.449
2.500	5.302	4.647	5.076	3.483

Potential timing attack on STM32F2

ROM lookup tables have non-constant time due to 128-bit wide FLASH bus (ART accelerator) on F2, but NOT present on F0/F1.

Almost impossible to catch this error without hardware validation (cycle-counting would NOT catch).

The End.

Conclusions

- Side channel power analysis – you should care. It's real. It's here.
- It can break most generic AES (or other crypto) implementations. ChipWhisperer-Lint can help you with leakage model if you need.
- We do automated software testing, why not also add hardware-in-the-loop automated testing to detect power side-channel flaws?

www.ChipWhisperer.com

github.com/pythohon/newaetech/chipwhisperer-lint

coflynn@newae.com