FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Assignment of bachelor's thesis

Title:	Power Analysis of Cryptographic Processor CEC 1702
Student:	Tereza Horníčková
Supervisor:	Dr.-Ing. Martin Novotný
Study program:	Informatics
Branch / specialization:	Computer engineering
Department:	Department of Digital Design
Validity:	until the end of summer semester 2022/2023

Instructions

Make the firmware development flow of Microchip CEC 1702 cryptographic processor functional. Use either the Clicker2 development kit by Mikroelektronika or the ChipWhisperer kit by NewAE, compare the two systems, and choose one for further work. Implement the AES cipher on the CEC 1702. Examine the resistance of the CEC 1702 to power analysis attacks. Compare the resistance of the software version of the AES cipher and the hardware version of the AES cipher using the built-in hardware accelerator. Also, compare the resistance of the Microchip CEC 1702 cryptographic processor and the STM32 processor.

Electronically approved by prof. Ing. Hana Kubátová, CSc. on 12 December 2021 in Prague.

FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Bachelor's thesis

Power Analysis of Cryptographic Processor CEC 1702

Tereza Horníčková

Department of Digital Design
Supervisor: Dr.-Ing. Martin Novotný

May 12, 2022

Acknowledgements

I would like to extend my heartfelt gratitude to both my incredibly helpful and patient supervisor M. Novotný as well as a fantastic partner in the adventure that was CEC1702, L. Daněk. Without them I would've struggled unspeakably more.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended. In accordance with Article 46 (6) of the Act, I hereby grant a nonexclusive authorization (license) to utilize this thesis, including any and all computer programs incorporated therein or attached thereto and all corresponding documentation (hereinafter collectively referred to as the "Work"), to any and all persons that wish to utilize the Work. Such persons are entitled to use the Work in any way (including for-profit purposes) that does not detract from its value. This authorization is not limited in terms of time, location and quantity. However, all persons that makes use of the above license shall be obliged to grant a license at least in the same scope as defined above with respect to each and every work that is created (wholly or in part) based on the Work, by modifying the Work, by combining the Work with another work, by including the Work in a collection of works or by adapting the Work (including translation), and at the same time make available the source code of such work at least in a way and scope that are comparable to the way and scope in which the source code of the Work is made available.

In Prague on May 12, 2022 . . . . . .

Czech Technical University in Prague
Faculty of Information Technology

Citation of this thesis

Horníčková, Tereza. *Power Analysis of Cryptographic Processor CEC 1702.* Bachelor's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2022.

Abstrakt

Cílem práce je poskytnout development flow pro kryptografický mikrokontroler CEC1702 a prozkoumat jeho odolnost proti útokům postranními kanály. Provedené útoky byla korelační odběrová analýza a korelační odběrová analýza vyššího řádu, útočící na první rundu šifrování. Po úspěšném napadení firmwarové implementace AES na jiném ARM procesoru - STM32F3 - bylo provedeno několik dalších - rovněž úspěšných - útoků na odpovídající implementaci na CEC1702. Poslední útoky byly vedeny na implementaci AES pomocí hardwarového akcelerátoru, kde všechny selhaly. Platformami pro útok byly ChipWhisperer toolchain a SICAK toolkit.

Klíčová slova AES, CEC1702, PicoScope, CPA, HOCPA, SICAK, Útoky Postranními Kanály, ChipWhisperer

Abstract

The goal of this work is provide a development flow for CEC1702 Cryptographic Microcontroller and to assess its robustness against Power Analysis side-channel attacks. Attacks mounted were Correlational Power Analysis and Higher-Order Power Analysis targeting first round of encryption. After successfully attacking firmware implementation of AES on another ARM controller — STM32F3, several other — also successful — attacks were mounted on corresponding implementation on CEC1702. Final attacks were mounted on the AES implementation using hardware accelerator, where all failed. Platforms for attack used were ChipWhisperer toolchain and SICAK toolkit.

Keywords AES, CEC1702, PicoScope, CPA, HOCPA, SICAK, Side-channel attacks, ChipWhisperer

Contents

Introduction **1**

1 State of the art **3**
1.1 AES Algorithm . . . 3
1.1.1 SubBytes . . . 3
1.1.2 ShiftRows . . . 3
1.1.3 MixColumns . . . 5
1.1.4 AddRoundKey . . . 5
1.2 Side-Channel attacks . . . 5
1.2.1 Differential Power Analysis . . . 5
1.2.2 Correlation Power Analysis . . . 5
1.2.3 Higher Order Differential Power Analysis . . . 6
1.2.4 Differential Power Analysis on AES . . . 6

2 Analysis and setup **9**
2.1 Microcontroller CEC1702 . . . 9
2.2 Clicker2 for CEC1702 . . . 9
2.3 ChipWhisperer toolchain . . . 11
2.3.1 CW308 CEC1702 Target board [1] . . . 13
2.3.2 Programming using the HW debugger . . . 13
2.3.3 Programming the SPI Flash . . . 13
2.4 mikroC pro for ARM IDE . . . 15
2.4.1 Points of caution about mikroC pro for ARM IDE . . . 15
2.5 Trace Capture Tools . . . 15
2.5.1 ChipWhisperer Lite . . . 15
2.5.2 SICAK toolchain . . . 17

3 Implementation **19**
3.1 AES Implementation . . . 19
3.1.1 Hardware AES . . . 20

3.1.2 Serial Communication ChipWhisperer 20
3.1.3 Serial Communication SICAK 20
3.2 CPA notebook 21

4 Testing **23**
4.1 AES 23
4.2 Serial communication testing against known values 23
4.3 Implementation of CPA in Mathematica 23

5 CPA Attack **25**
5.1 CPA on example data 25
5.2 CPA using ChipWhisperer Toolchain 26
5.2.1 STM32F3 26
5.2.2 CEC1702 26
5.3 CPA on CEC1702 using SICAK Toolkit 27
5.3.1 Firmware AES 28
5.3.2 Hardware AES 30
5.3.3 Higher order CPA 34

6 Future work **37**
6.1 Attacks on other platforms 37
6.2 Other attacks 37
6.2.1 Last round CPA 37
6.3 Analysis scripts 37

Conclusion **39**

Bibliography **41**

A Acronyms **45**

B Contents of enclosed SD card **47**

List of Figures

1.1 AES diagram adapted from [2] 4
1.2 Spot to attack during the first round 7
1.3 Spot to attack during the last round 7

2.1 Clicker2 for CEC1702 10
2.2 Faulty UART communication as seen using Advanced Serial Port Terminal.[3] 10
2.3 Measured UART communication with CEC1702; sending 'A'; used oscilloscope: *Agilent DSOX3012A* 11
2.4 Measured UART communication with CEC1702; sending "AAA"; used oscilloscope: *Agilent DSOX3012A* 12
2.5 OTP Programming for CEC1702 as written in Help section of mikroC PRO 12
2.6 Project Settings to use CEC1702 with hardware programmer . . . 14
2.7 Wire connection UFO board with CEC1702 target while using mikroProg for CEC 14
2.8 Example of errors upon compilation in case the `Non-case sensitive` toggle is kept on. Code screenshot taken from code related to [4] . 16
2.9 Top solution will compile but act incorrectly, bottom is the correct solution with this compiler. Code screenshot taken from code related to [4] 16

5.1 Setup for trace capture on STM32F3 using ChipWhisperer Lite. 1 — STM32F3 connected to UFO board, 2 — ChipWhisperer Lite, 3 — ChipWhisperer Lite connection to PC via microUSB, 4 — measuring probe connected to J17, 5 — 20pin cable connection between UFO board and ChipWhisperer Lite 27
5.2 Depicted is one AES encryption done using its firmware implementation in its entirety as captured using ChipWhisperer Lite with STM32F3. 28

5.3 Setup for trace capture on CEC1702 using ChipWhisperer Lite. 1 — Chipwhisperer Lite, 2 — mikroProg For CEC, 3 — CEC1702 connected to UFO board, 4 — ChipWhisperer Lite connection to PC via microUSB, 5 — mikroProg For CEC connection to PC via miniUSB, 6 — measuring probe connected to J17, 7 — 20pin cable connection between UFO board and ChipWhisperer Lite 29
5.4 Depicted is one AES encryption done using its firmware implementation in its entirety as captured using ChipWhisperer Lite with CEC1702. 30
5.5 Setup while using PicoScope. 1 — PicoScope A channel probe (power consumption), 2 — PicoScope C channel probe (measuring trigger), 3 — mikroProg for CEC, 4 — 6pin UART to USB converter: CP2102, 5 — USB connection to PicoScope, 6 — CEC1702 target mounted to UFO target board, 7 — Advanced Serial Port Terminal 31
5.6 Setup when using PicoScope 32
5.7 Depicted is one AES encryption done using its firmware implementation in its entirety as captured using Picoscope6404D. 32
5.8 Graph of correlations traces for first correlation matrix. Plotted in black is correlation trace of the first key candidate (correct). 33
5.9 Depicted is one AES encryption done using hardware accelerator in its entirety as captured using PicoScope6404D. 33
5.10 Graph of correlations traces for first correlation matrix. Plotted in black is correlation trace of the first key candidate (correct but visibly not the one with the greatest correlation). 34
5.11 Graph of correlations traces for first keybyte using Second Order Analysis. Plotted in black is correlation graph of first key candidate (correct but visibly not the one with greatest correlation). 35

List of Tables

5.1 Measured platforms 25
5.2 Time and space used 29
5.3 Time and space used 31
5.4 Time and space used 34
5.5 Mounted attacks configurations 35

Introduction

In the modern world of global networking, the topic of protection of private data enjoys more attention than ever. This includes the data gathered by low-power embedded devices found all over the industry as well as in many home appliances. Their nature more often than not requires the collected data to be processed by a third-party. This interaction calls for heightened security in the form of fast and hard to successfully attack encryption. Many microcontrollers already have the hardware to fulfill these needs. One such microcontroller within easily affordable price range was chosen to assess its robustness against attacks targetting the physical projection of the encryption process.

The goal of this work is to provide an operational firmware development flow of Microchip CEC 1702 cryptographic processor and make an assessment of its resistance against side channel attacks – specifically power analysis type of attacks. The assessment will be made by comparing basic firmware and hardware implementation of AES cipher using different development kits for CEC 1702 and another ARM based processor — STM32F.

The first chapter provides brief introduction to theoretical foundations of AES cipher and power analysis attacks. Following is the analysis chapter discussing different platforms and tools pertaining to CEC1702 that were considered or used during the course of this work. The third chapter introduces the development flow for CEC1702 and implemented firmware. Chapter four introduces testing methods used during development of used implementations. The fifth chapter discusses results of attacks on each configuration. Final chapter summarizes the entirety of our work.

Chapter 1

State of the art

This chapter will introduces basic theory behind this work. It describes the workings of the AES cipher and correlation power analysis together with its application on AES cipher.

1.1 AES Algorithm

AES is a widespread encryption standard established by the U.S. National Institute of Standards and Technology in 2001 [5] as the replacement for the at the time already outdated DES. Input and output consist of sequences of 128 bits. The 16 bytes of input sequence are organized into a 4x4 matrix called the State. AES algorithm is capable of performing encryption using 128, 192 or 256 bits long key. This work is focused only on the 128 bit variant which is comprised of 10 rounds of four data transforming operations – byte substitution according to a predefined table, byte permutation, byte multiplication and key addition to the data. The key addition also occurs prior to the first round and is applied to the initial plaintext.

1.1.1 SubBytes

The only non-linear part of the cipher. This transformation is applied to each byte of the State independently by substituting it with a specific value from substitution table (S-Box).

1.1.2 ShiftRows

This operation cyclically shifts last three rows of the State matrix to the left by different offsets. The second row moves by one, third two and the fourth row by three. The first row remains unchanged.

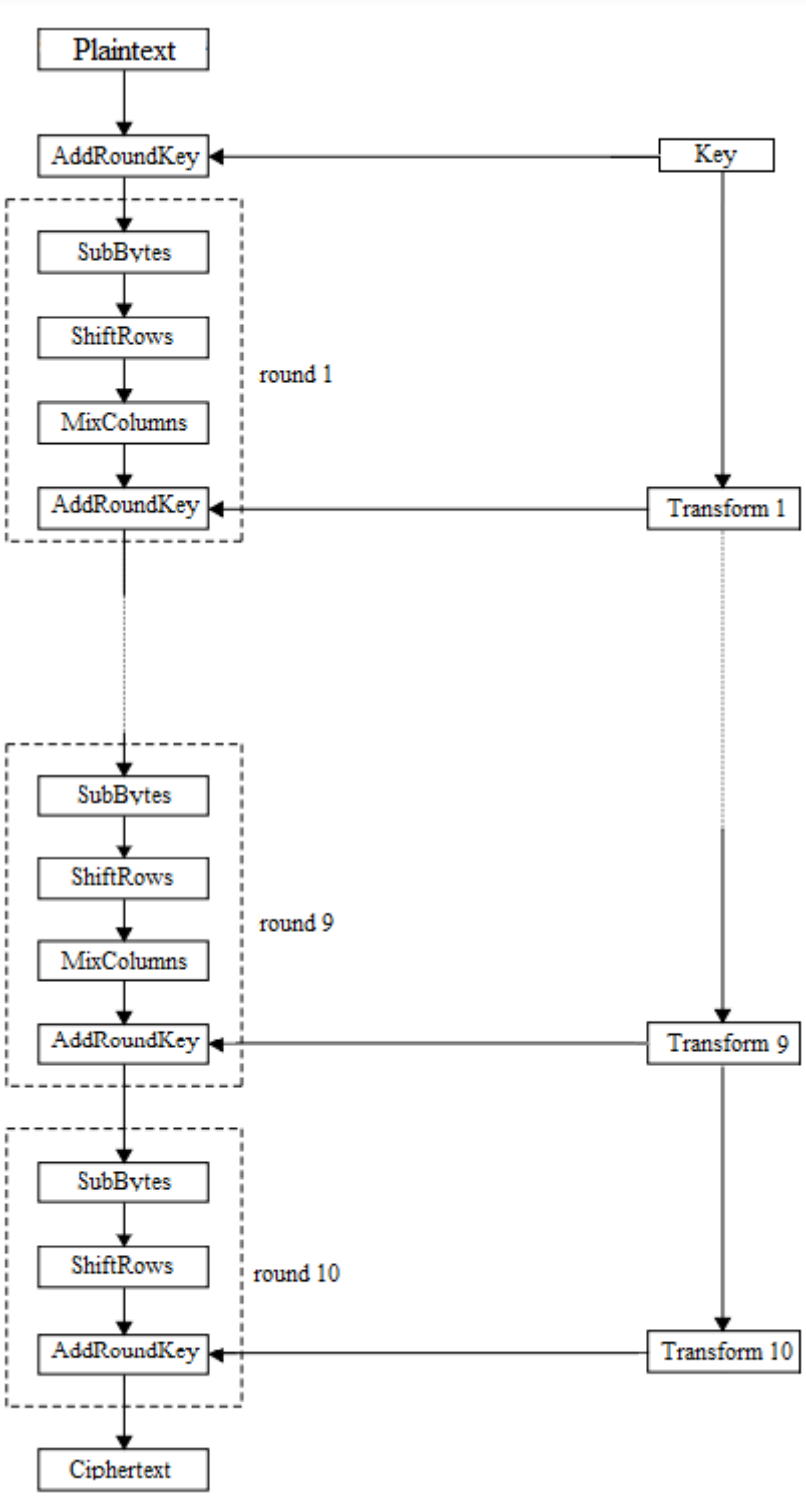

Figure 1.1: AES diagram adapted from [2]

1.1.3 MixColumns

Columns of the State matrix are multiplied by a fixed 4x4 matrix. All operations are done within Galois Field. This operation does not take place during the last round.

1.1.4 AddRoundKey

Round key is added to the State by a simple bitwise XOR operation. The original key is applied in this way prior to the first round and for each following the key undergoes an expansion defined by the standard's specification.

1.2 Side-Channel attacks

Side-channel attacks are types of attacks based on exploiting physical manifestation of the handled data. Vital information may leak based on the device's power consumption, electromagnetic radiation, etc as all of these are closely tied to the values of handled data. As such, even a mathematically secure cipher's implementation can be successfully attacked if it doesn't incorporate appropriate countermeasures against these attacks. Examples of some of the proposed countermeasures would be dual-rail logic [6], masking [7] or threshold cipher implementations [8].

1.2.1 Differential Power Analysis

Differential Power Analysis (DPA) is a type of side-channel attack using relationship of power consumption and handled data. The method was first introduced by P. Kocher et al. in[9]

This method stands on measurements of power consumed by the device while performing encryption and having access to either the original plaintext or resulting ciphertext. Following the measurements, a power consumption prediction model needs to be computed. This model predicts approximate power consumption for given sample using Hamming's weight for each possible key candidate.

1.2.2 Correlation Power Analysis

For the variant used in this thesis, after the model is computed the search for correlation between the hypothetical model and actual measurements begins. Correlation between measured values and correct hypothetical key candidate is significantly higher compared to the rest. This version of the attack is known as Correlation Power Analysis (CPA), which was first introduced in [10] in 2004.

1.2.3 Higher Order Differential Power Analysis

Various countermeasures such as dual-rail logic [6], masking [7] or threshold cipher implementations [8] were proposed to prevent DPA types of side-channel attacks. While they do prevent first-order analysis, vulnerabilities remain against higher-order attacks.

A higher-order DPA attack is — as defined by Kocher et al. [9] — an attack that combines one or more samples within a single power trace. This is especially useful against AES S-box masking.

1.2.4 Differential Power Analysis on AES

Mounting an attack using DPA requires a value with nonlinear dependency on used key and known data (ie. plaintext or ciphertext). The only point in AES algorithm satisfying this condition is the operation SubBytes. Hence the value used is the one produced after this operation.

Another condition for the DPA attack is the lack of relation between bytes to enable formulation of power hypothesis for each keybyte separately. This condition gets broken by MixColumns operation, which limits the rounds appropriate for the attack to the first and the last round only.

Thick black line in Figure 1.2 depicts an appropriate point to target within the first round. The round is preceded by an AddRoundKey operation which means the value that's being processed is dependant on the key and this dependency is nonlinear thanks to the following SubBytes operation. Plaintext used during the encryption is required when attacking at this point.

In case of attacking the last round, the attacker is reliant on having access to the resulting ciphertext. The attack targets the point just before the last SubBytes operation (see Figure 1.3). SubBytes and AddRoundKey once again fullfill the nonlinear key dependency condition with the additional ShiftRows operation not causing any interference as it only performs a byte permutation.

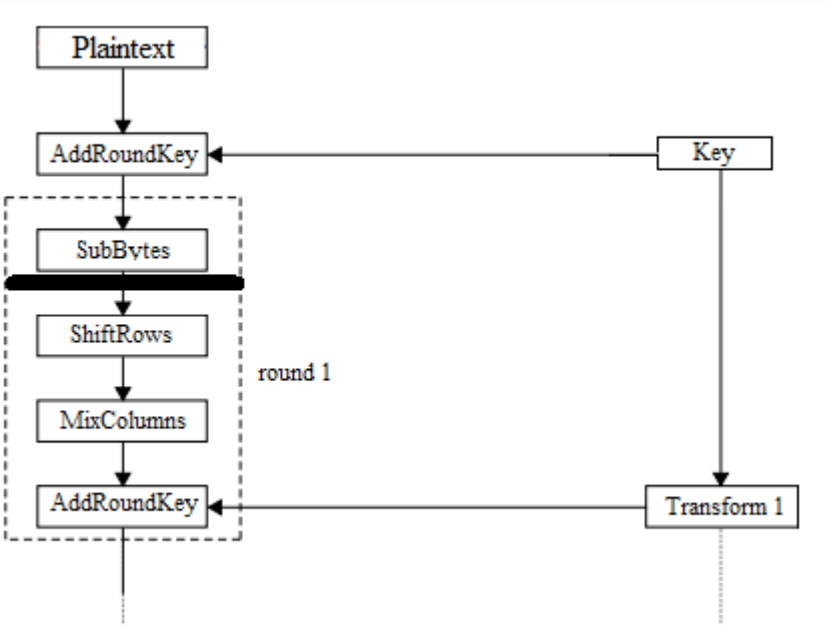

Figure 1.2: Spot to attack during the first round

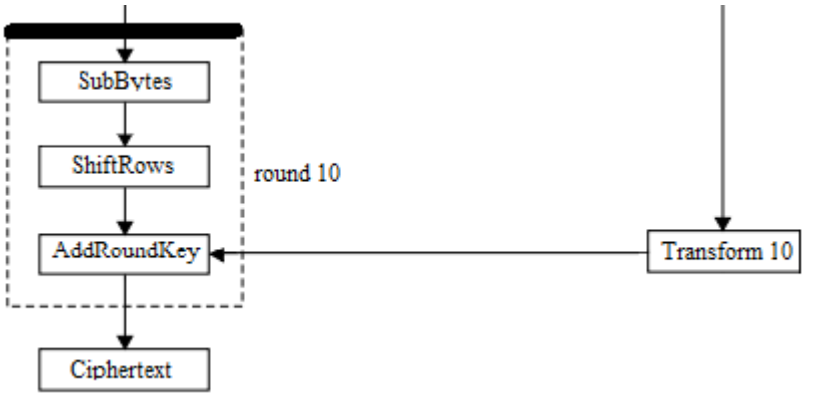

Figure 1.3: Spot to attack during the last round

Chapter 2

Analysis and setup

This chapter's focal topics are cryptographic processor CEC1702 (CEC1702) and setup of related development tools. The setup process is a co-project with L. Daněk that we took over after initial steps were made by Ing. J. Klemsa.

2.1 Microcontroller CEC1702

CEC1702 is a cryptographic microcontroller containing RSA, AES and SHA hardware accelerators, a true random number generator and a 32-bit ARM Cortex-M4cs core. The microcontroller was chosen prior to the assignment of this thesis based on commercial availability and lack of NDA gate to documentation of its cryptographic functions which is — to our knowledge — a combination of traits unique to it among microcontrollers with similar advertised security level. Development boards supporting this microcontroller that we considered and worked with are Clicker2, which is manufactured and sold by MikroElektronika[11], and CW308T-CEC1702 — a target board for NewAE's [12] side-channel attack specialized kit. The boards are further discussed later in this chapter. Accessible MikroC for ARM [13] integrated development environment offered by MikroElektronika with API for all of its peripherals seemed to be yet another attractive trait.

2.2 Clicker2 for CEC1702

We were initially introduced to the project at the point where the LED blinking example project had already been made functional using the Clicker2 development board (see Figure 2.1), hardware programmer mikroProg for CEC and Mikroelektronika's technical support's advice — "try uploading the LED blinking through the mikroPROG's options → Program EFuse". However the example project for UART communication that we programmed the MCU with after was producing erroneous results. The error manifested in the form

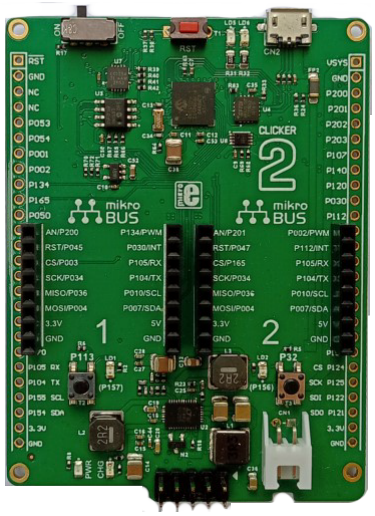

Figure 2.1: Clicker2 for CEC1702

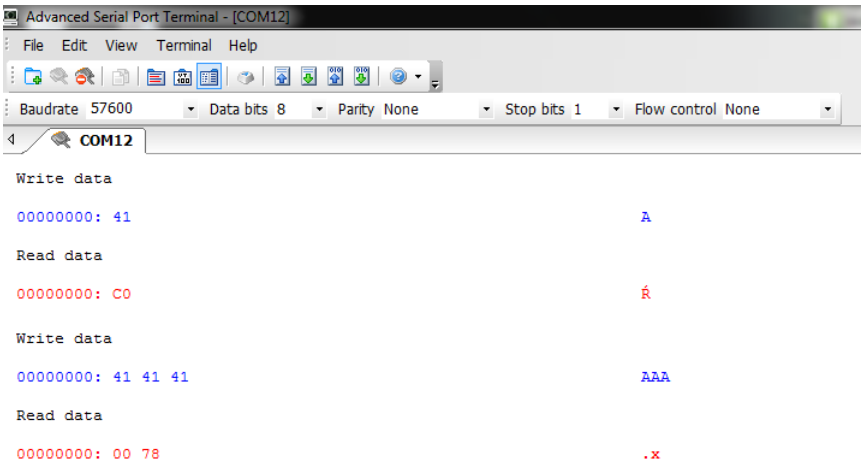

Figure 2.2: Faulty UART communication as seen using Advanced Serial Port Terminal.[3]

of wrong data being received (see Figure 2.2), in case of sending more than two bytes even seemingly randomly lost.

Further testing using an oscilloscope (Figures 2.3 and 2.4) brought about a hypothesis that there are problems with either the clock speed or baud rate setting (57600 baud corresponded to approximately 32000 baud, 9600 to 4800).

After editing the UART example code to manually program the baud rate using the system clock and divisor setting, with system clock set to 24 MHz,

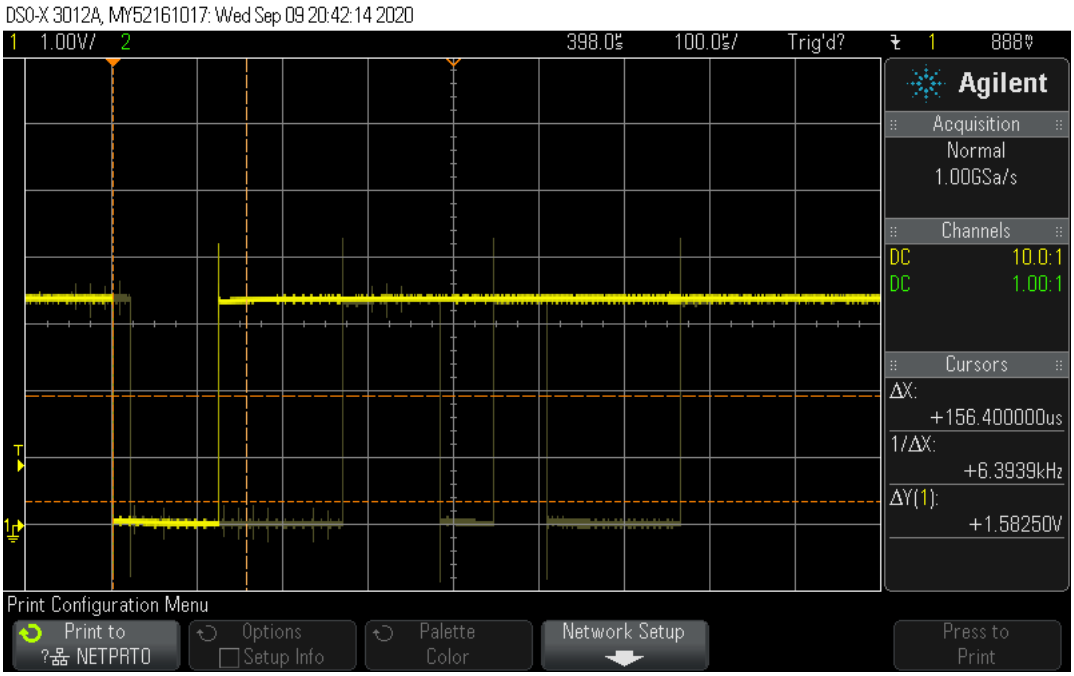

Figure 2.3: Measured UART communication with CEC1702; sending 'A'; used oscilloscope: *Agilent DSOX3012A*

desired baud rate of 9600 and divisor set to expected 156, the errors were exactly the same as before. Their disappearance after setting the divisor to 172 (experimentally reached value) confirmed the actual clock speed to be above the set threshold (24 MHz → 26.4 MHz).

By consulting NewAE documentation[12], UART problems were confirmed to be a common occurrence caused by faultily set/unset EFUSE bits. We attempted to set the EFUSE bits according to a tutorial provided in the Help section of mikroC PRO for ARM IDE (see Figure 2.5) without any semblance of success.

Multiple attempts to contact the tech support for specific, in-depth guide to setting up the board and environment resulted in two boards' EFUSE bits being wrongly set and therefore rendered worthless. After several more query-heavy emails the technical support stopped responding altogether.

In response to these happenings we abandoned Clicker2 board in favor of CW308-CEC1702 target board for ChipWhisperer toolchain.

2.3 ChipWhisperer toolchain

"ChipWhisperer is a complete open source toolchain for learning about side channel attacks on embedded devices and validating the side channel resis-

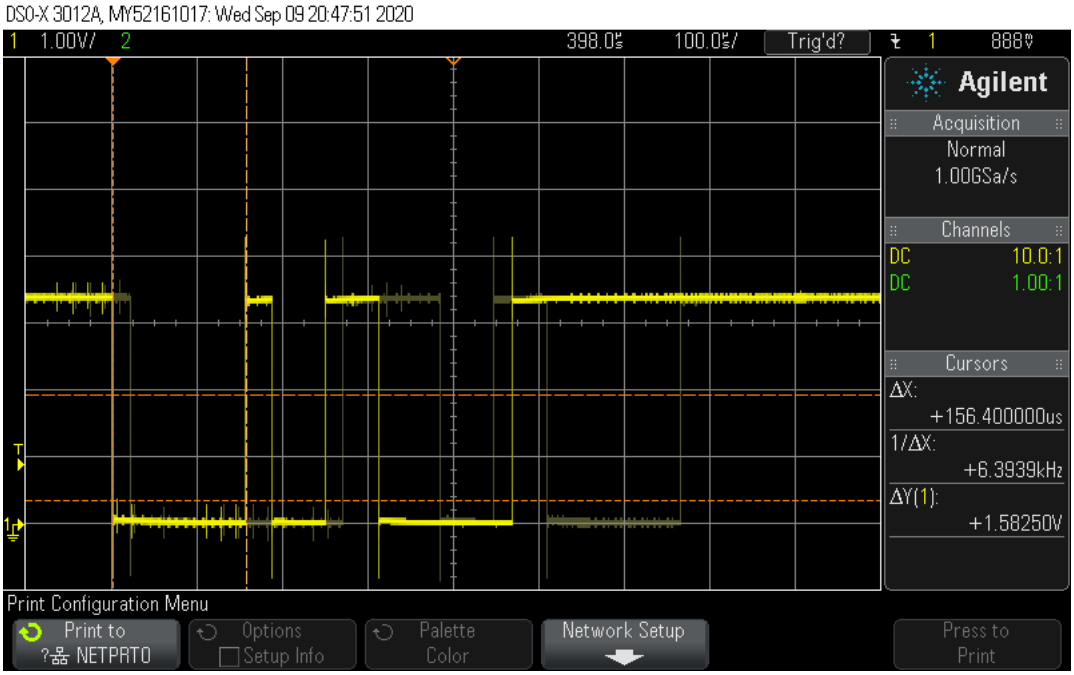

Figure 2.4: Measured UART communication with CEC1702; sending "AAA"; used oscilloscope: *Agilent DSOX3012A*

OTP Programming the CEC1702 device

Option one:

1. Generate eFUSE data with Microchip eFUSE generator tool,
2. Copy generated header file `efuse_data.h` into MikroE_otp_prog project folder,
3. Build MikroE_otp_prog project for hardware debug,
4. Start debugger session (*Run->Start Debugger*),
5. Run debugger (*Run->Run/Pause Debugger*),
6. Turning on LED on Clicker/Clicker2 board indicates that OTP memory is written.

Option two:

1. Generate eFUSE data with Microchip eFUSE generator tool,
2. Run mikroProg Suite for ARM tool,
3. Open Options, select Hardware Interface to CEC,
4. Click Program eFUSE button and choose generated HEX or binary image created in Step 1,
5. Confirm that you are sure that you want to program OTP memory,
6. Turning on LED on Clicker/Clicker2 board indicates that OTP memory is written.

Documentation and tool for the eFUSE data generation is located in: `Tools\CEC1702 Efuse Generator`.
MikroE_otp_prog project is located in: `Tools\CEC1702 Efuse Generator\MikroE_otp_prog`.

Figure 2.5: OTP Programming for CEC1702 as written in Help section of mikroC PRO

tance of these devices." [14] As an entry to this toolchain we used starter kit SCAPACK-L1 [15]. This kit includes a ChipWhisperer compatible capture board (ChipWhisperer-Lite), versatile target board (CW308-UFO) and three exchangeable target boards with 8-bit XMEGA and 32-bit ARM STM32F0/3 microcontrollers. Multitude of target boards with other microcontrollers is available for individual purchase — CEC1702 board being one of them.

Attractive features are NewAE's active support, well-documented API and the amount of tutorials for side-channel attacks using ChipWhisperer. We used these tutorials and the STM32F3 target board to better familiarize ourselves with the toolchain as well as our chosen attack methods. Insights gained from this venture were later applied to work with the board using CEC1702.

2.3.1 CW308 CEC1702 Target board [1]

Target board available as an individually purchasable target for CW308-UFO target board. Immediately convenient properties of this board when compared to Clicker2 are the accessibility of pins for side-channel attacks and that its shipped with EFUSE bits already properly set.

2.3.2 Programming using the HW debugger

This approach requires the user to run **mikroProg suite for ARM v161** [16].

The target board itself only has pins to interact with the UFO board and therefore any connections to additional peripheral devices such as programmers have to be wired from there. UFO board has a universal 2x5 pin connector, which was used to connect JTAG debugger **mikroProg for CEC** to temporarily program the MCU. Wire connection is depicted in Figure 2.7

To program the device in debugging mode using **mikroProg for CEC**, there's a need to toggle the **Build Type** to **Debug** and **Debugger** to **Hardware** in Project Settings (one of toggle-able windows accessible from View tab). To be seen in Figure 2.6.

2.3.3 Programming the SPI Flash

Another way of programming the CEC1702 target is from external flash via SPI programmer. In such a case there's a need to either hold the nRST button or connect the J8 jumper on the UFO board. The board uses SST26VF016B flash chip which requires command code 98 to disable global block protection prior to each write operation. This command code can be added to the communication protocol within the user's firmware.[1]

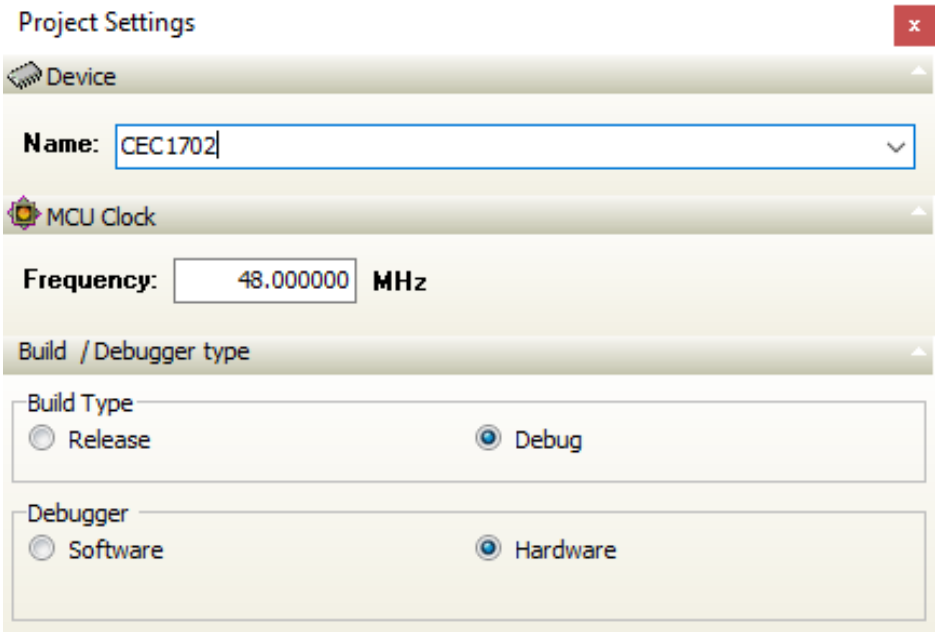

Figure 2.6: Project Settings to use CEC1702 with hardware programmer

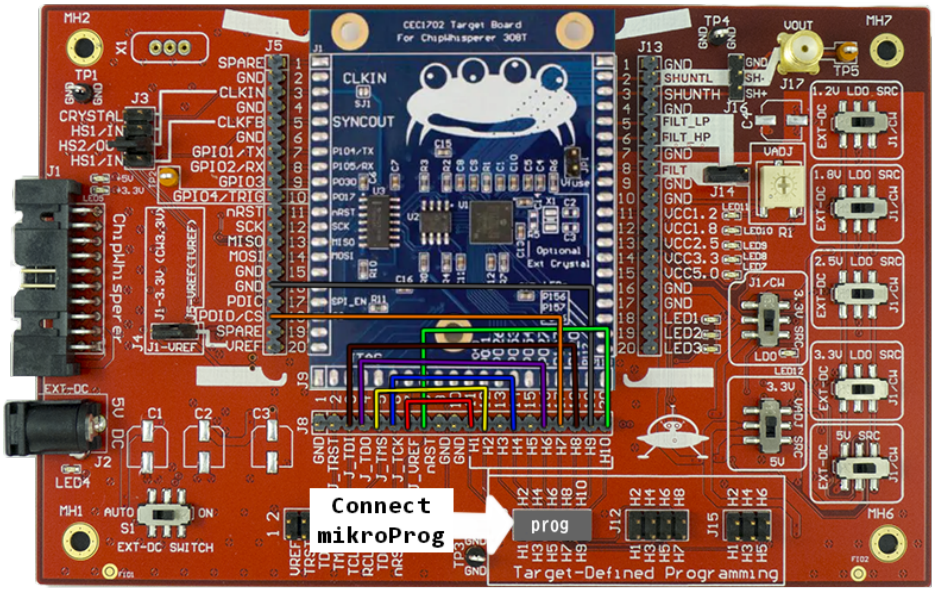

Figure 2.7: Wire connection UFO board with CEC1702 target while using mikroProg for CEC

2.4 mikroC pro for ARM IDE

CEC1702's firmware is designed to work with this specific IDE. While a certain level of support can be found in other IDE's such as MPLAB [17], the lack of libraries for use of peripheries makes the choice of IDE still very streamlined. Frankly there's no way anybody would actually make a conscious choice to work with this unless held at a gunpoint which is pretty much the case here.

2.4.1 Points of caution about mikroC pro for ARM IDE

Demo version of the IDE is available with all the features and example codes. The only limitation is the size of code that can be compiled and programmed into the target device. Therefore it is recommended to secure a licensed version in case of working on a larger project.

While the compiler is mostly for ANSI C, we encountered several errors stemming from the compiler going against the said ANSI C standard. Notable and most baffling case of this being the existence of a case sensitivity toggle which is furthermore set to non-case sensitive by default. This toggle is accessed through `Options` → `Output` → `Output` → `Settings` → `Compiler`. Compilation errors hinting towards this problem are depicted in Figure 2.8.

Another noteworthy trait would be a lack of operation priority with typecasting. This trait manifests in a way where performing an arithmetic operation and then casting the result in one line acts unexpectedly — the operands first undergo the cast and the arithmetic operation follows after. Hence the result can easily end up warped especially when the user is used to standard behavior of most C compilers. Specific situation and its solution is to be seen in Figure 2.9.

2.5 Trace Capture Tools

Two different platforms for trace capture and analysis were used during the attacks. First being the ChipWhisperer toolchain [14] where ChipWhisperer Lite was used as the measuring device and Wolfram Mathematica[18] script implementing CPA was created for the analysis. Second platform was SICAK toolkit developed by P. Socha[19] using PicoScope 6404D[20] as the measuring device and the further analysis was done using SICAK's utilities.

2.5.1 ChipWhisperer Lite

Main part of the kit offering both trace capture and glitch generation capabilities.[21] Board connects to the PC via micro USB cable and to the target board via a 20-pin connection. All communication is done through ChipWhisperer's Python API. The board's maximum sampling rate is 105 MSa/s but the use of this is speed is limited by maximum sample size, which is only 24400.

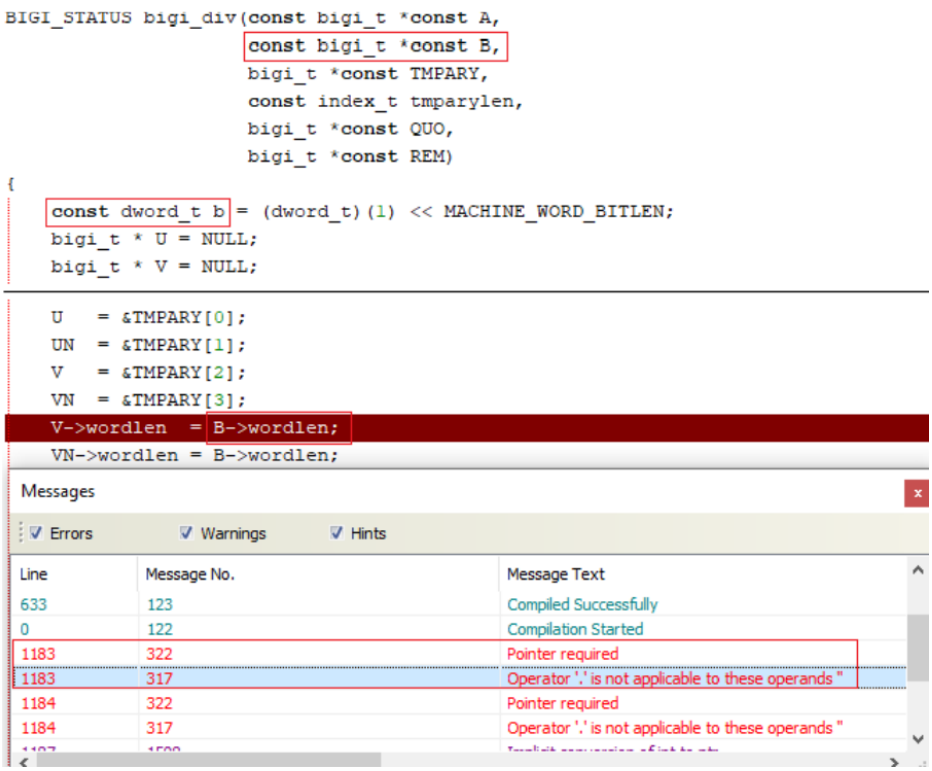

Figure 2.8: Example of errors upon compilation in case the `Non-case sensitive` toggle is kept on. Code screenshot taken from code related to [4]

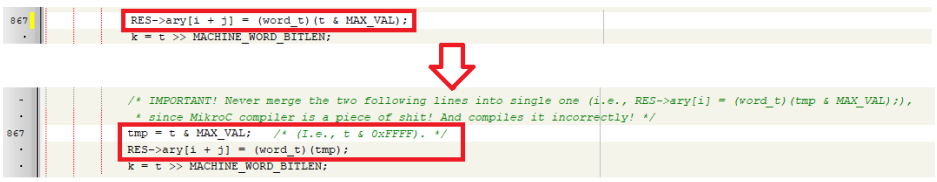

Figure 2.9: Top solution will compile but act incorrectly, bottom is the correct solution with this compiler. Code screenshot taken from code related to [4]

It offers synchronous capture improving performance in comparison to asynchronous setup.

2.5.2 SICAK toolchain

SIde-Channel Analysis toolKit (SICAK)[19] is an open-source toolkit. Users may utilize text utilities such as `meas` or `stan` to interact with various plugins relevant to mounting a side-channel attack. Current release version (1.2.1) doesn't support more than one capture at time while using `ps6000` plugin (PicoScope 6000 series plugin). As such, working with the version on GitHub repository where this issue has already been resolved is recommended.

PicoScope 6404D was used due to it's 5 GSa/s sample rate paired with 2 Gpts maximum memory depth being more suitable for the attack on hardware implementation of AES.

Chapter 3

Implementation

This chapter describes the implementations of AES algorithm used in this thesis, specifications of communications protocols and CPA script in Wolfram Mathematica [18]. All AES implementations are in ECB mode.

Source codes can be found on attached electronic media in the folder `/source`

3.1 AES Implementation

Implementations used are coded very naively with no added countermeasures to maximize the impact of used hardware. Firmware implementation of AES is built based on a template provided in BI-HWB [22] following the NIST specification[5].

ChipWhisperer based AES implementations receive one command byte, 32 bytes of data to work and a newline symbol with via UART. This is to accomodate ChipWhisperer's default communication protocol called `SimpleSerial`.[23] The command bytes expected by these implementations are:

- 'k' indicating the following data sequence to be the key
- 'p' indicating the following data sequence to be the plaintext

AES implementations interacting with SICAK toolchain receive one command byte and 16 bytes of data to work with via UART. The command bytes expected by these implementations are:

- 0x01 indicating the following data sequence to be the key
- 0x02 indicating the following data sequence to be the plaintext

After receiving the plaintext sequence, the implementation goes on to an encrypting loop which repeats according to a hardcoded amount of `CAPTURES`. Each iteration after the first one uses the preceding ciphertext as its new

plaintext and sends its own ciphertext out via UART communication. After the loop ends the program returns to a waiting loop until it receives another command. This is added purely for the sake of measuring time reduction as the initialization of UART communication takes up considerable amount of time compared to trace capture itself. For illustration, measuring 2000 traces with this loop took 3 seconds while the version without took 8 seconds.

3.1.1 Hardware AES

Implementation using CEC1702's hardware accelerator uses available API for cryptographic functions[24]. However, the source code of these functions is not publicly available and therefore specific countermeasures used against side-channel attacks are unknown.

While the processor's clock speed can be lowered, when measuring only the hardware AES encryption the time it took did not change proportionally to the clock speed (when decreasing the clock frequency 4×, namely from 48MHz to 12 MHz, the execution time incresed from 3,4 μs to 8.8 μs, i.e. just 2.6×.). This leads to a belief that either the hardware accelerator is running off full 48 MHz clock at all times, or the overhead for using cryptographic library functions forms a bottleneck.

3.1.2 Serial Communication ChipWhisperer

The version for communication with the ChipWhisperer adheres to specifications of SimpleSerial protocol and doesn't deviate from the implementation provided as part of ChipWhisperer's tutorials [23]. The implementation expects data in ASCII format followed by a newline or carriage return symbol. Therefore, after the command is determined, the received data related to AES such as key and plaintext first undergo a reformat from 32 ASCII symbols to 16 corresponding bytes. Reverse of this operation is later performed on data being sent out. ChipWhisperer's SimpleSerial protocol on firmware version 1.1 also expects a confirmation of successful communication in the form of an `ack` signal: "z(`ack_value`)\n". The `ack_value` is determined by return value of the function called through the received command with '0' signaling success and '1' failure.

3.1.3 Serial Communication SICAK

The version for communication with the SICAK toolkit — specifically the `meas` utility — works with binary data only. The protocol waits until it receives 17 bytes of data from UART, determines the command and passes the received data without command byte to the specified function.

3.2 CPA notebook

While not used in an attack on CEC1702's hardware accelerator, a Wolfram Mathematica notebook implementing CPA was made for the sake of familiarization with the attack. The notebook is available on enclosed medium in folder `/CPA data/CPA_EXAMPLE`. We tested the notebook with datasets provided by supervisor. Datasets are available in the same folder. We also used this notebook to analyze traces obtained by ChipWhisperer when attacking STM32F target and CEC1702 target running sole SW version of AES algorithm.

The notebook receives a plaintext text file and a traces file. Received plaintext is converted to a vector of hex values `inputs` while traces are saved in a `traceCount` X `sampleCount` matrix `traces`. Following process is repeated for each of the 16 keybytes.

First the power hypothesis is computed. Values of vector `inputs` are first bitwise `xored` with each key candidate and added to a `inputsXORkeys` matrix (`traceCount` X 256), which is then mapped to power model `HamBox`. This power hypothesis is stored as `HWinputsXORkeys` matrix (`traceCount` X 256). Following is computation of correlation matrix (256 X `sampleCount`). Correlation matrix is computed from absolute value of correlation of transposed `HWinputsXORkeys` and `traces`. Maximum value in this matrix is stored in `max`. The row index of mentioned maximum value can be assumed to be the correct value of the currently analysed keybyte. Hence the code finally loops through `CorrelationMatrix` rows to extract this index.

Chapter 4

Testing

This chapter discusses testing of used AES implementations, serial communication and script implementing CPA.

4.1 AES

Testing of all AES implementations was performed by running the specific implementation of its respective platform. All implementations produce files with used plaintext and resulting ciphertext. Produced plaintext was then encrypted with known key in online AES encryption tool [25] and the resulting ciphertext was then manually compared with the one produced by the implementation.

4.2 Serial communication testing against known values

Testing of the functions of implemented serial communication protocols was was done using Advanced Serial Port Terminal[3] by sending valid data and then comparing the validity of received response.

4.3 Implementation of CPA in Mathematica

The script with implementation of the attack used when attacking STM32F3 and firmware version of AES on CEC1702 was tested on example data provided by the supervisor. Example data is available at [26]. The attack on example data is closely described in Section 5.1.

Chapter 5

CPA Attack

This chapter describes and analyzes mounted attacks and their results (see Table 5.1). It starts with introductory attack on example data which is followed by attacks on firmware versions of AES cipher on STM32F3 and CEC1702. All of the mentioned attacks were successful and data gathered from them is used as a comparison for the attack on CEC1702's AES cipher using AES hardware accelerator. This attack failed.

5.1 CPA on example data

This attack was done to verify the correctness of CPA implementation in Mathematica notebook (see Section 3.2). The first version was applied to two sets of 100 example measurements provided by the supervisor. One set came with a known key while the other were only traces, plaintext and ciphertext. Keys of both sets were successfully uncovered using an attack on the first round of AES. Verification of success regarding the set with unknown key was done by comparing provided ciphertext with ciphertext resulting from an encryption using the uncovered key.

Table 5.1: Measured platforms

MCU	platform	HW	success
STM32	CW	✗	✓
CEC1702	CW	✗	✓
CEC1702	SICAK	✗	✓
CEC1702	SICAK	✓	✗

5.2 CPA using ChipWhisperer Toolchain

Attacks in this section were mounted using ChipWhisperer toolchain together with a Wolfram Mathematica notebook (see Section 3.2). ChipWhisperer-Lite board (CW-Lite) connected to CW-UFO via 20-pin connector was used as the trace capture device. CW-Lite's maximum sample rate is 105 MSa/s, however the rate used in these attacks was lower as to accomodate CW-Lite's maximum Sample Buffer Size — 24400 samples. This limit is higher on other ChipWhisperer capture boards — 96000 for CW-1200 and 131070 for CW-Husky. [27]

All commmunication was done through ChipWhisperer's Python API with data transfer rate being 34800 Baud..

5.2.1 STM32F3

Attack on naive implementation of AES running on 7.385 MHz clock on STM32F3. Capturing single trace on this configuration takes 939.29 μs. While capturing maximum sample count (24400 samples per trace) over the course of these measurements, the sampling rate was limited to only 29.5 MSa/s to cover the entire range. The setup is to be seen in Figure 5.1.

The attack was successful after measuring and analyzing 50 traces. A singular trace can be seen in Figure 5.2. The analysis was done using the Wolfram Mathematica notebook with CPA implementation. Measured data, Python Book file used for capture and the CPA notebook file are available on enclosed medium in folder `/CPA data/STM32`.

5.2.2 CEC1702

Attack on naive implementation of AES running on 48 MHz clock on CEC1702 using ChipWhisperer Lite and CPA script written in Wolfram Mathematica notebook. Although the CEC1702 has no option of running off on an external clock, attacker is provided a possibility to use a 12 MHz PWM signal that is generated on the CLKOUT pin. By setting a jumper on J3 pins to HS1/IN (set to HS2/OUT by default), the ChipWhisperer will run off of the ExtClk, which provides almost synchronous sampling. A single trace takes up 500 μs with sampling rate in this attack being 48.51 MHz. The setup is to be seen in Figure 5.3.

One successful attack was mounted using only 10 000 traces with the key being `[69b00b6969b00b6969b00b6969b00b69]`. Attempts to replicate this with other data as well as keys were unsuccessful. Using 13 000 traces didn't provide any approach to success. A singular trace can be seen in Figure 5.4. Data for both the successful 10 000 trace attack as well the failed 10 000 trace and 13 000 trace attacks are available on enclosed medium in folder `/CPA data/CEC1702_FW_CW`.

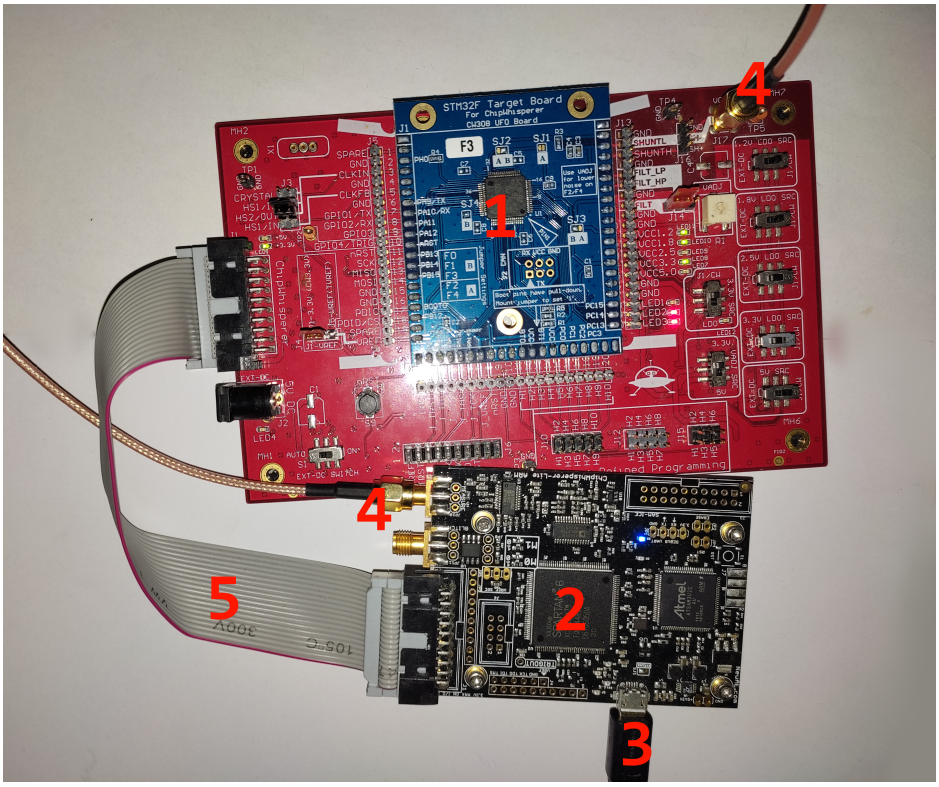

Figure 5.1: Setup for trace capture on STM32F3 using ChipWhisperer Lite. 1 — STM32F3 connected to UFO board, 2 — ChipWhisperer Lite, 3 — ChipWhisperer Lite connection to PC via microUSB, 4 — measuring probe connected to J17, 5 — 20pin cable connection between UFO board and ChipWhisperer Lite

Both of these attacks had key `[00112233445566778899aabbccddeeff]`.

5.3 CPA on CEC1702 using SICAK Toolkit

Attacks in this section were mounted using SICAK Toolkit [19] and PicoScope6404D [20]. ChipWhisperer UFO target board was used both as a charge supply for CEC1702 target board as well as provider of easily accessible pins for trace capture.

PicoScope's channel A was connected to J17 (SHUNTL pin) and channel C to TP2 (trigger pin). Full setup is seen in Figure 5.5 with closer look at connections to the UFO board provided in Figure 5.6. Sampling rate was set to 2.5 GSa/s which is the highest rate while using two channels. When used with `meas` utility from SICAK, unused channels have to be explicitly turned off to achieve this sampling rate. Combinations A/B or C/D would achieve

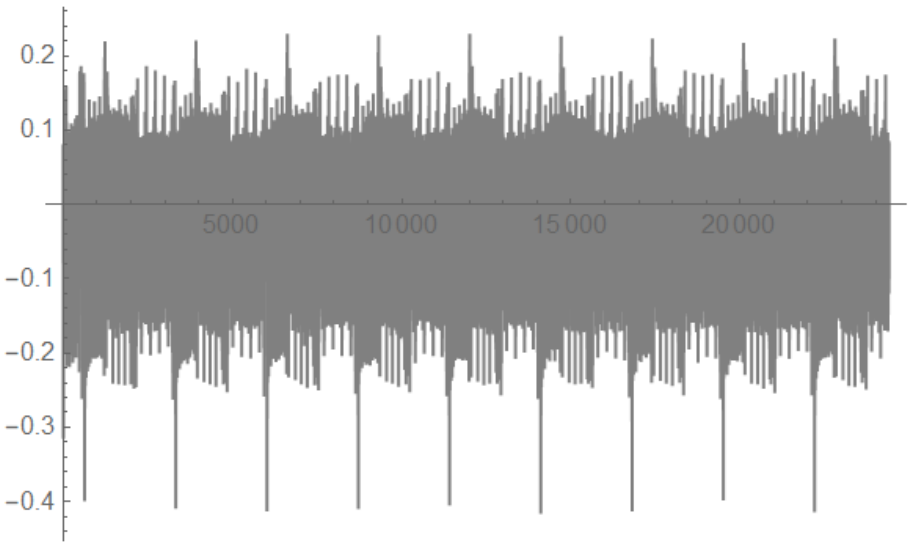

Figure 5.2: Depicted is one AES encryption done using its firmware implementation in its entirety as captured using ChipWhisperer Lite with STM32F3.

half the sampling rate, as Channels A and B, or C and D, respectively, share the same sample circuit. Other combinations of channels provide full sampling rate.[20] In this case, channels A and C were used for the sake of achieving the optimal sampling rate. Specific sample sizes are handled in SICAK using `picoscope6000` plugin. These adjustments are done in relation to trace length and set amount of *captures* (measurements to be performed and have related data held in picoscope's buffer each cycle).

All communication was done using a 6pin UART to USB converter CP2102 at 115200 baud rate using COM3 port.

All encryptions used key `[00112233445566778899AABBCCDDEEFF]`

5.3.1 Firmware AES

Attack on naive implementation of AES running on 48 MHz clock. While the entire encryption takes up over 500 μs (see Figure 5.7), the post-trigger range in configuration file for `meas` is set only to 60 μs for the sake of time and space conservation (see Table 5.2). Each trace is comprised of 150000 samples. Successful attack was mounted using 16000 traces. Using an identical configuration, measuring and analysing 15000 traces yielded an incorrect key guess: `[00112233445566778899aabbccc4eeff]`.

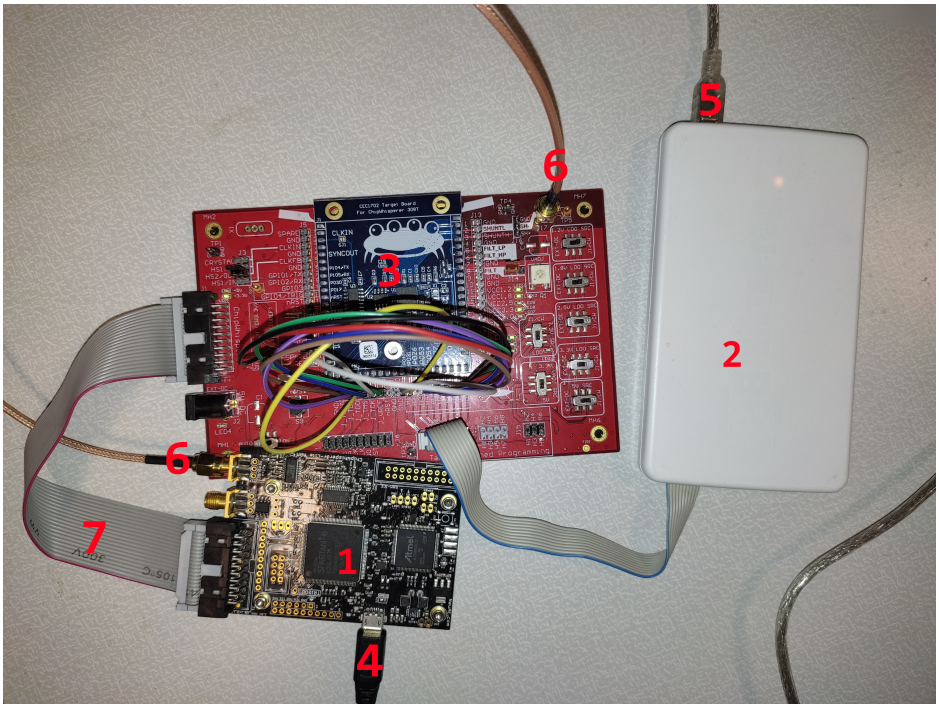

Figure 5.3: Setup for trace capture on CEC1702 using ChipWhisperer Lite. 1 — Chipwhisperer Lite, 2 — mikroProg For CEC, 3 — CEC1702 connected to UFO board, 4 — ChipWhisperer Lite connection to PC via microUSB, 5 — mikroProg For CEC connection to PC via miniUSB, 6 — measuring probe connected to J17, 7 — 20pin cable connection between UFO board and ChipWhisperer Lite

Table 5.2: Time and space used for each command in SICAK pipeline — Firmware AES 16000 traces

Command	`Time`	`File Size`
`meas`	3m 15s	4.69 GB
`prep`	1s	64 MB
`stan - create`	120m	4.84 GB
`stan - finalize`	43s	4.8 GB
`correv`	6s	✗

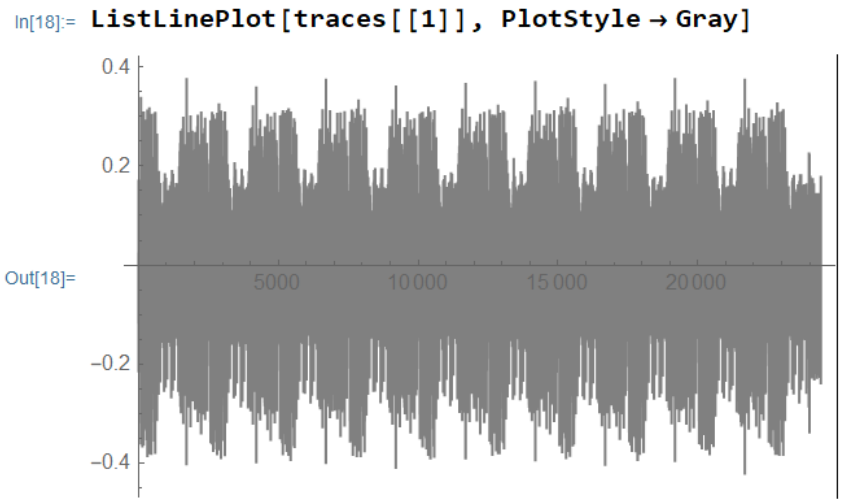

Figure 5.4: Depicted is one AES encryption done using its firmware implementation in its entirety as captured using ChipWhisperer Lite with CEC1702.

5.3.2 Hardware AES

The attack on AES using hardware accelerator is the main attack of this thesis. The clock speed was reduced from 48 MHz to 12 MHz. A single encryption as seen in Figure 5.9 takes 8.8 μs with this reduced speed. This time may be lower as the `trigger_low` command in the code happens only after the completion signal has been processed. However due to the uncertainty of how large the actual time lag is, the configuration file for `meas` command uses this time as a *post-trigger range* value. Each trace is comprised of 1375 samples. Multiple attacks were mounted with trace count rising each time, starting with 1 million and ending with 100 million traces. Due to the time and space demands (see Table 5.3), the measurements for this large attack were split into 10 separate sessions capturing 10 million traces each. Following the computation of separate *contexts*, SICAK's context merge utility was used to produce a merged context whose finalization resulted in correlation matrix based on data from all 100M traces. Plot of correlation graph for the first correlation matrix — first byte of the key — is to be seen in Figure 5.10.

Actual key used for encryption was `[00112233445566778899AABBCCDDEEFF]`. Guessed key that was the result of this attack was `[F00B0F0D0F32F0EBD2F40EDCAD800FF0]` with 68 out of 128 bits being incorrect.

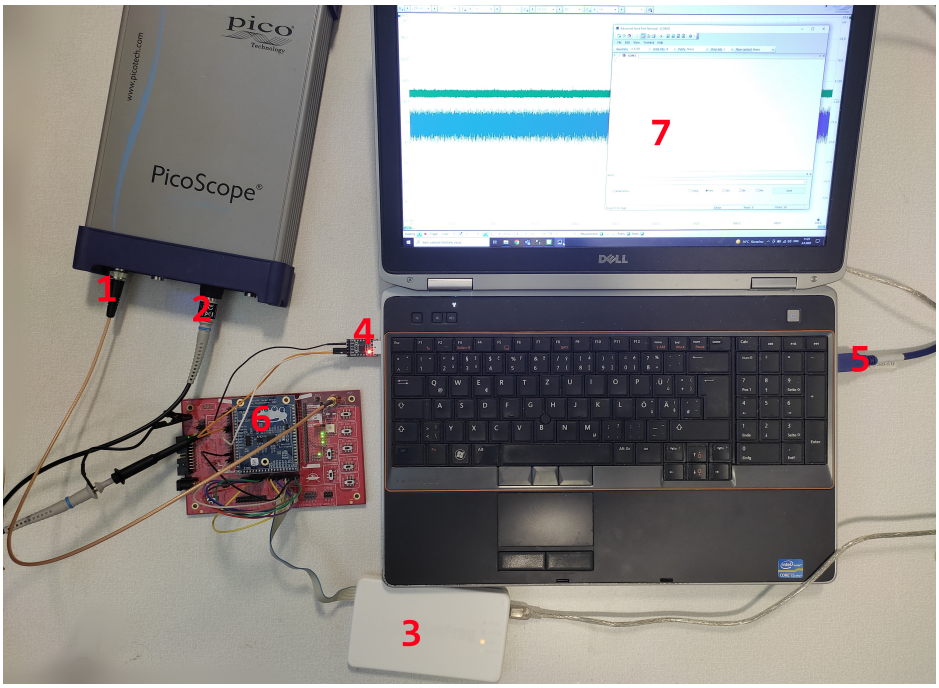

Figure 5.5: Setup while using PicoScope. 1 — PicoScope A channel probe (power consumption), 2 — PicoScope C channel probe (measuring trigger), 3 — mikroProg for CEC, 4 — 6pin UART to USB converter: CP2102, 5 — USB connection to PicoScope, 6 — CEC1702 target mounted to UFO target board, 7 — Advanced Serial Port Terminal

Table 5.3: Time and space used for each command in SICAK pipeline — HW AES 100M traces

Command		Time	File Size
`meas`	`(x10)`	5h 20m	26.86 GB
`prep`	`(x10)`	40m	40 GB
`stan - create`	`(x10)`	13h	44.4 MB
`stan - merge`	`(x9)`	1s	44 MB
`stan - finalize`	`(x1)`	1s	44 MB
`correv`	`(x1)`	1s	✗

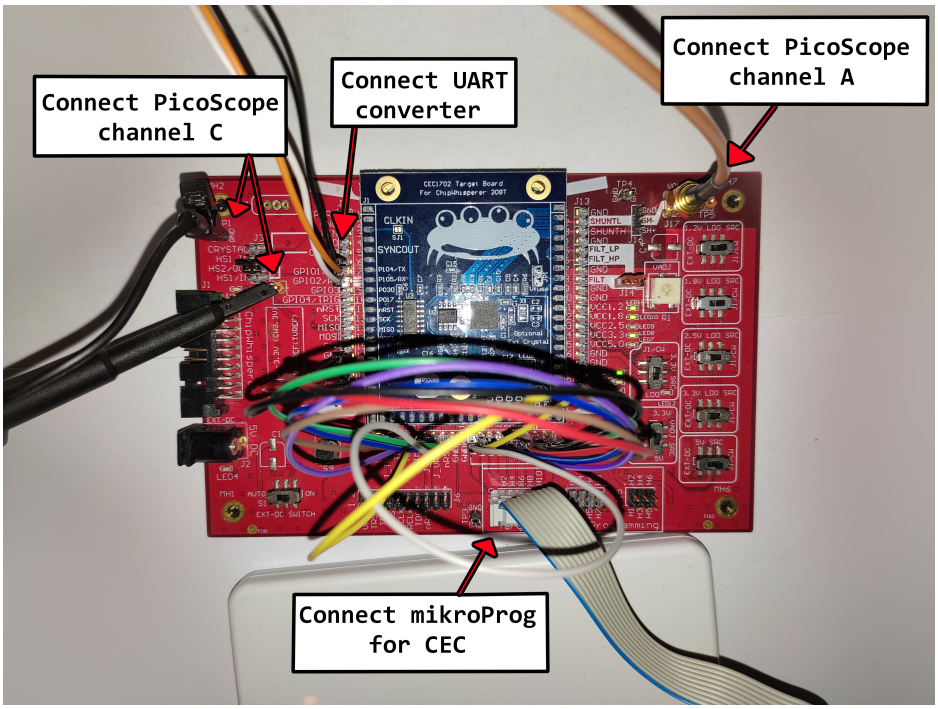

Figure 5.6: Setup when using PicoScope

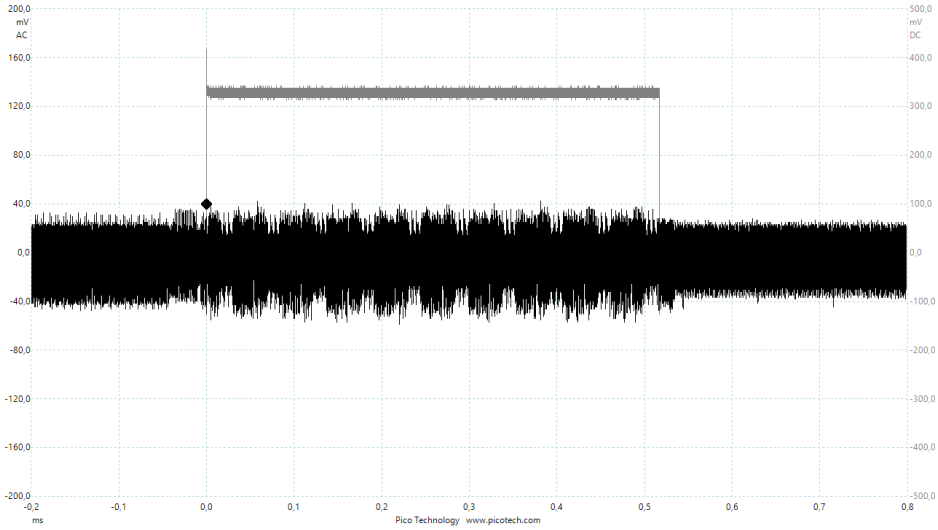

Figure 5.7: Depicted is one AES encryption done using its firmware implementation in its entirety as captured using Picoscope6404D.

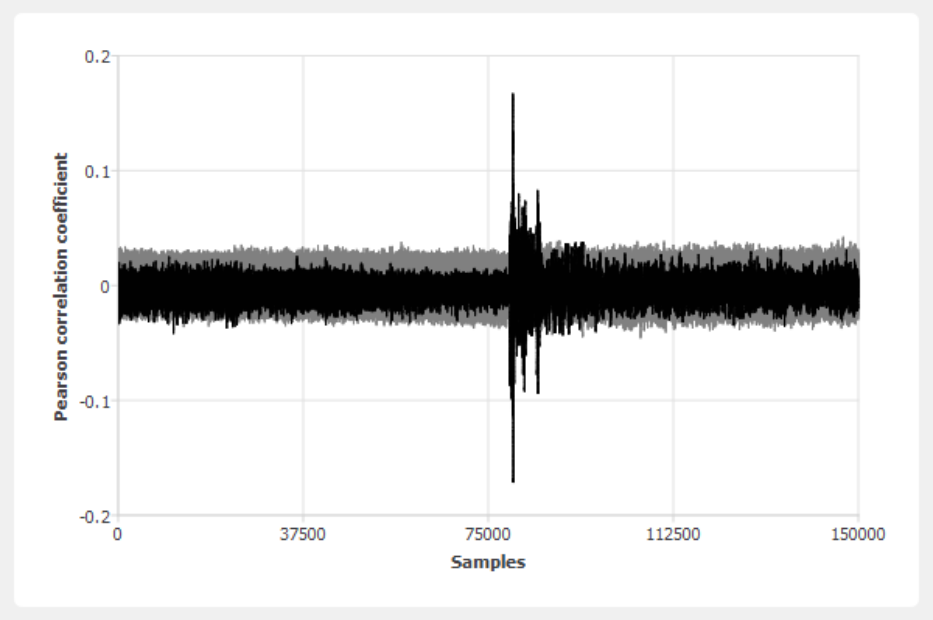

Figure 5.8: Graph of correlations traces for first correlation matrix. Plotted in black is correlation trace of the first key candidate (correct).

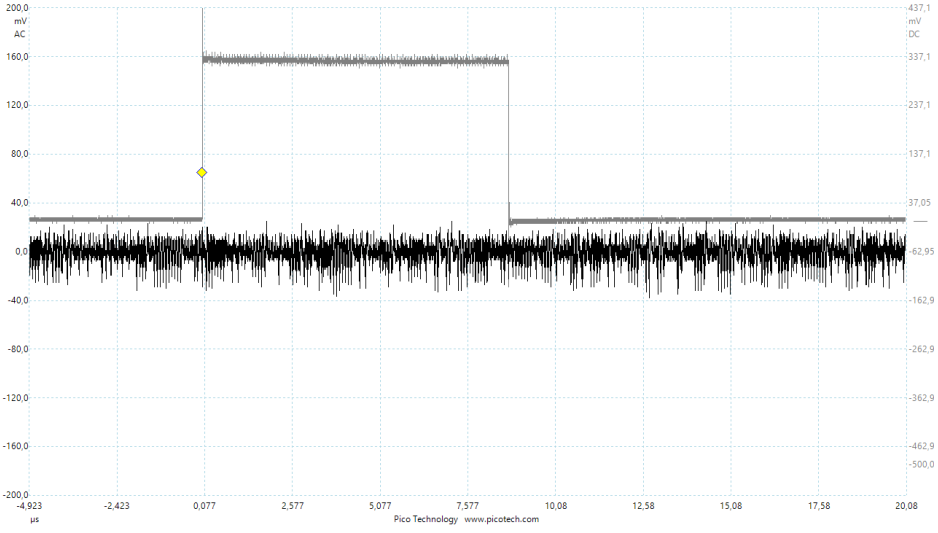

Figure 5.9: Depicted is one AES encryption done using hardware accelerator in its entirety as captured using PicoScope6404D.

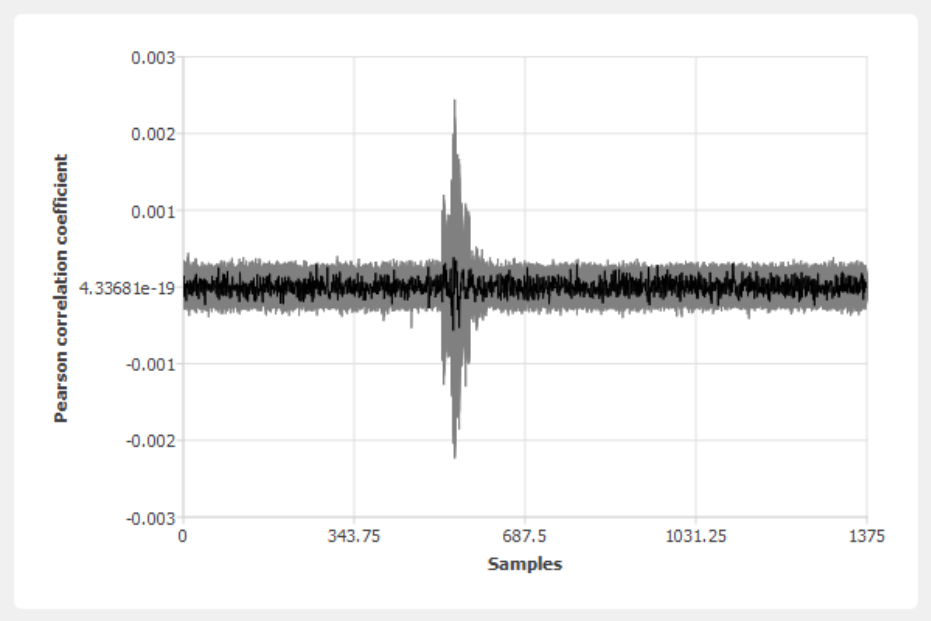

Figure 5.10: Graph of correlations traces for first correlation matrix. Plotted in black is correlation trace of the first key candidate (correct but visibly not the one with the greatest correlation).

Table 5.4: Time and space used for statistical analysis of second order — HW AES 10M traces

Command	`Time`	`File Size`
`stan - create`	8h 40m	88.8 MB
`stan - finalize`	1s	44 MB
`correv`	1s	✗

5.3.3 Higher order CPA

Higher order CPA was attempted on AES using hardware accelerator. The attack used 10 million traces that were measured as part of the final attack in 5.3.2. Due to time constraints, only a second-order analysis was completed. Plot of correlation graph for the first correlation matrix — first byte of the key — is to be seen in Figure 5.11.

This attack failed. Guessed key that was the result of this attack was `[F00B0FF0D613A6EBD40D4BB4507FDFF4]` with 55 out of 128 bits being incorrect.

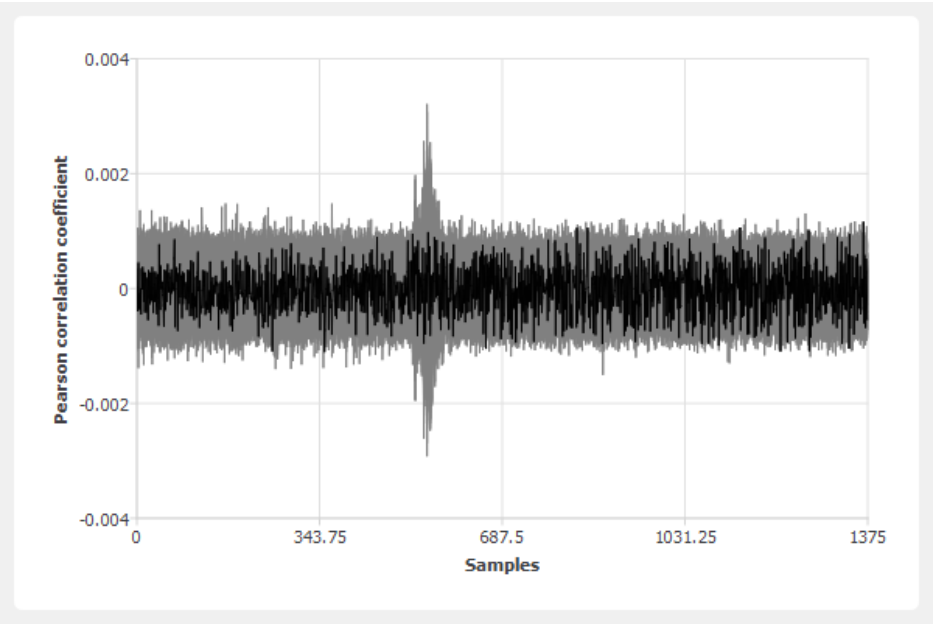

Figure 5.11: Graph of correlations traces for first keybyte using Second Order Analysis. Plotted in black is correlation graph of first key candidate (correct but visibly not the one with greatest correlation).

Table 5.5: Mounted attacks configurations

Type	Trace count	Sampling Rate	Samples	Success
STM32F3 - CW	50	29.538 MSa/s	24400	✓
CECFW - CW	10k	48.51 MSa/s	24400	✓
CECFW - PS6	16k	2.5 GSa/s	150k	✓
CECHW - CPA	100M	2.5 GSa/s	1375	✗
CECHW - HOCPA	10M	2.5 GSa/s	1375	✗

Chapter 6

Future work

This chapter outlines possible further ways to ascertain CEC1702's robustness against power analysis attacks.

6.1 Attacks on other platforms

While STM32F series devices with hardware acceleration for AES exist, none were used in these attacks. Procurement of one of these and its attack on its hardware implementation of AES could provide further insight in terms of CEC1702's robustness. Several boards with such devices are even available as target boards for ChipWhisperer toolchain (STM32F2, STM32F4 and STM32L4).

6.2 Other attacks

As the power analysis attack on first round of hardware implementation of AES was unsuccessful even with a considerable amount of data, use of other attacks could be considered appropriate.

6.2.1 Last round CPA

All power analysis attacks in this thesis were carried out on the first round after the SubBytes operation. Future attempts could include attacks on the last round as well. These attacks would be focused on acquiring a round key and then reverse computing the original key.

6.3 Analysis scripts

This thesis doesn't provide insight into changes in correlation coefficient depending on the number of traces. As such there is no conclusive way to

state minimal amount of traces needed to mount a successful attack for the firmware implementations. Implementation of a visualisation utility for this purpose seems to be an effective way to reach more conclusive results.

Another useful utility to implement would be a script for calculating guessing entropy.

Conclusion

Goal of this thesis was to provide a firmware development flow for Microchip CEC1702 cryptographic processor as well as asses it's resistance to power analysis attacks.

The first step of a development flow production was to choose a fitting development platform out of the two mentioned in the assignment. While the attempt at making the Clicker2 functional itself ended up in failure, work with the ChipWhisperer development kit turned out successful. Furthermore this platform ended up being deemed as considerably more fitting in the face of further work's needs. The platform considerably lessens the difficulty of mounting a side channel attack and the documentation is publicly available along with an active social network community.

Naive implementations of AES cipher were used for software version using both STM32F03 and CEC1702. The only differences were platform dependent. The hardware version for CEC1702 utilized AES hardware accelerator using the publicly available API.

The resistance related measurements were done using both the ChipWhisperer platform as well as using Picoscope and SICAK tool. Both platforms operate with slightly different data format, hence there was a need to implement appropriate communication protocols.

While the attacks on all software implementations were successful, the version utilizing integrated hardware accelerator resisted all attempts. The final and most computationally demanding were the attempts where 100M traces had first-order CPA applied to them and the attempt where higher order CPA, specifically second-order, was applied to 10M traces. These results imply that the hardware AES accelerator was designed to be resistant against both basic as well as higher-order power analysis attacks.

Bibliography

[1] CW308T-CEC1702, [Online], NewAE Technology Inc. [cit. 2022-11-05]. Available from: `https://rtfm.newae.com/Targets/UFO%20Targets/CW308T-CEC1702/`

[2] Paar, C.; Pelzl, J. A Textbook for Students and Practitioners. *Understanding Cryptography, Springer*, 2009.

[3] Advanced Serial Port Terminal, [Online]. [cit. 2022-11-05]. Available from: `https://www.eltima.com/products/serial-port-terminal/`

[4] Říha, J.; Klemsa, J.; et al. Multiprecision ANSI C Library for Implementation of Cryptographic Algorithms on Microcontrollers. In *2019 8th Mediterranean Conference on Embedded Computing (MECO)*, 2019, pp. 1–4, doi:10.1109/MECO.2019.8760285.

[5] National Institute of Standards and Technology: FIPS PUB 197 [online]. 2001, [Cited 2022-11-05]. Available from: `http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf`

[6] Popp, T.; Mangard, S. Masked dual-rail pre-charge logic: DPA-resistance without routing constraints. In *International Workshop on Cryptographic Hardware and Embedded Systems*, Springer, 2005, pp. 172–186.

[7] Blömer, J.; Guajardo, J.; et al. Provably secure masking of AES. In *International workshop on selected areas in cryptography*, Springer, 2004, pp. 69–83.

[8] Moradi, A.; Poschmann, A.; et al. Pushing the limits: A very compact and a threshold implementation of AES. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, 2011, pp. 69–88.

[9] Kocher, P.; Jaffe, J.; et al. Differential power analysis. In *Annual international cryptology conference*, Springer, 1999, ISBN 978-3-540-66347-8, pp. 388–397.

[10] Brier, E.; Clavier, C.; et al. Correlation power analysis with a leakage model. In *International workshop on cryptographic hardware and embedded systems*, Springer, 2004, ISBN 978-3-540-22666-6, pp. 16–29.

[11] Clicker 2 for CEC1702 Development Board. [cit. 2022-11-05]. Available from: https://www.mikroe.com/clicker-2-cec1702

[12] Inc., N. T. CW308T-Cec1702. [cit. 2022-11-05]. Available from: https://rtfm.newae.com/Targets/UFO%20Targets/CW308T-CEC1702/

[13] Mikroc Pro for ARM: Mikroe. [cit. 2022-11-05]. Available from: https://www.mikroe.com/mikroc-arm

[14] ChipWhisperer — Overview [online], NewAE Technology Inc. [cit. 2022-11-05]. Available from: https://chipwhisperer.readthedocs.io/en/latest/getting-started.html

[15] Level 1 Starter Kit (SCAPACK-L1), [Online], NewAE Technology Inc. [cit. 2022-11-05]. Available from: https://rtfm.newae.com/Starter%20Kits/SCAPACK-L1/

[16] mikroProg for CEC, [Online], MikroElektronika d.o.o. [cit. 2022-11-05]. Available from: https://www.mikroe.com/mikroprog-cec

[17] MPLAB X Integrated Development Environment (IDE), [Online], Microchip Technology Inc. [cit. 2022-11-05]. Available from: https://www.microchip.com/en-us/tools-resources/develop/mplab-x-ide

[18] Wolfram Mathematica, [Online], Wolfram. [cit. 2022-11-05]. Available from: https://www.wolfram.com/mathematica/

[19] Socha, P. Sicak: Side-channel analysis toolkit [Online]. [cit. 2022-11-05]. Available from: https://petrsocha.github.io/sicak/

[20] PicoScope 6000CD Series Datasheet [online], Pico Technology Ltd. [cit. 2022-11-05]. Available from: https://www.picotech.com/download/datasheets/PicoScope6000CDSeriesDataSheet.pdf

[21] CW1173 ChipWhisperer-Lite, [Online], NewAE Technology Inc. [cit. 2022-11-05]. Available from: https://rtfm.newae.com/Capture/ChipWhisperer-Lite/

[22] Buček, J. Lectures and Tutorials of course BI-HWB [Online]. [cit. 2022-11-05]. Available from: https://courses.fit.cvut.cz/BI-HWB/

[23] SimpleSerial Documentation [online], NewAE Technology Inc. [cit. 2022-11-05]. Available from: https://github.com/newaetech/chipwhisperer/tree/develop/hardware/victims/firmware/simpleserial

[24] CEC/MEC Family Devices TOM API User's Guide [online], Microchip Technology Inc. [cit. 2022-11-05]. Available from: https://www.microchip.com/content/dam/mchp/documents/OTH/ProductDocuments/UserGuides/50002157B.pdf

[25] OnlineDomainTools: AES encryption [online]. [cit. 2022-11-05]. Available from: http://aes.online-domain-tools.com/

[26] Novotný, M. Lectures and Tutorials of course Embedded Security (NI-BVS) [online], Pico Technology Ltd. [cit. 2022-11-05]. Available from: https://courses.fit.cvut.cz/NI-BVS/

[27] ChipWhisperer API — Scope [online]. [cit. 2022-11-05]. Available from: https://chipwhisperer.readthedocs.io/en/latest/api.html#chipwhisperer.scopes.OpenADC.adc

Appendix A

Acronyms

AES Advanced Encryption Standard

DPA Differential Power Analysis

CPA Correlation Power Analysis

HOCPA Higher-Order Correlation Power Analysis

UART Universal Asynchronous Receiver/Transmitter

EFUSE Electrically Programmable Fuse

CW ChipWhisperer

SICAK SIde-Channel Analysis toolKit

Appendix B

Contents of enclosed SD card

- `CPA data` ..... the file with CD contents description
- `source` ..... the directory with source codes
  - `CEC1702_AES_CW` implementation of AES and communication with CW
  - `CEC1702_AES_SICAK` . implementation of AES and communication with SICAK
  - `SICAK_adjust` ..... SICAK source code that was changed
- `text` ..... the thesis text directory
  - `thesis.pdf` ..... the thesis text in PDF format
  - `thesis.ps` ..... the thesis text in PS format