TURUN
YLIOPISTO
UNIVERSITY
OF TURKU

PERFORMING ELECTROMAGNETIC SIDE-CHANNEL ATTACK ON A COMMERCIAL AES-256 DEVICE

Mika Kaustinen, Ohto Myllynen,
Tero Jokela, Lauri Koskinen,
Olli Heimo & Tero Säntti

TECHNICAL REPORT | TURKU 2021

TURUN
YLIOPISTO
UNIVERSITY
OF TURKU

PERFORMING ELECTROMAGNETIC SIDE-CHANNEL ATTACK ON A COMMERCIAL AES-256 DEVICE

Mika Kaustinen, Ohto Myllynen, Tero Jokela,
Lauri Koskinen, Olli Heimo & Tero Säntti

TURUN YLIOPISTON JULKAISUJA – ANNALES UNIVERSITATIS TURKUENSIS
TECHNICAL REPORT | TURKU 2021

University of Turku

Faculty of Technology
Department of Computing
Electronics

ISBN 978-951-29-8652-1 (Print)
ISBN 978-951-29-8653-8 (Online)
ISSN 0082-6979 (Print)
ISSN 2343-3183 (Online)
Painosalama, Turku, Finland 2021

UNIVERSITY OF TURKU
Faculty of Technology
Department of Computing
Electronics
MIKA KAUSTINEN, OHTO MYLLYNEN, TERO JOKELA, LAURI KOSKINEN, OLLI HEIMO & TERO SÄNTTI: Performing Electromagnetic Side-Channel Attack on a Commercial AES-256 Device
Technical report, 24 pp.
FitOptiVis Project H2020-ECSEL-2017-2-783162.
November 2021

ABSTRACT

In this paper an electromagnetic side-channel attack on a commercial AES-256 USB-encryption module operating in ECB mode is introduced. In preparation for the attack, oscilloscope, electromagnetic probe with low-noise amplifier and isolated power supply were used together with computer to record 10000 plaintext encryptions. The attack was conducted with the collected plaintext-ciphertext pairs and EM traces corresponding to each encryption. The attack was con-ducted with Correlation Power Analysis method and Matlab software. The power consumption (and thus the EM emission) of the device was modeled using hamming distance metric.

The correlation between modeled power consumption and measured traces allowed the extraction of AES round keys one byte at a time. For AES-256 last two round keys (rounds 13 and 14) were needed to complete the key schedule. Finding these two keys allowed to calculate the original secret key from which they were expanded. For successful attack several trials were required to find right measurement setup for oscilloscope and electromagnetic head position. In this attack 30 out of the 32 round key bytes were found using side-channel attack and the two remaining were found using brute force. The device was found to have some kind of backdoor mechanism.

KEYWORDS: EM, side-channel, cryptography, attack, AES-256, ECB, Matlab, Oscilloscope

Table of Contents

Table of Figures

1 Introduction

In cryptography, a side-channel attack (SCA) is any attack based on information gained from the physical implementation of a cryptosystem, rather than brute force or theoretical weaknesses in the algorithms. For example, electromagnetic leaks can provide an extra source of information, which can be exploited to break the system [1]. Fault-injection and most side-channel attacks require direct access or contact to the device under attack. Such side-channel attacks include power [2] and temperature [3] analysis or cache-based [4] attacks. However, all architectures and code run on the system leak EM side-channel information due to the physical structure of integrated circuits [5] and this has been exploited in side-channel attacks previously [6].

The commercial SCA tools are expensive, to a degree that excludes their use in this context. These commercial attack tools are made for the industry for example by Rambus [7] and Riscure [8]. SAKURA circuit boards [9] have also been used often in an academic study, but they do not offer tools to proprietary circuit boards. On NEWAE Chipwhisperer [10] the clock of the circuit board can be synchronised with the attack device which reduces the synchronising need for the measurings afterwards. However, this requires a specific interface to the circuit boards. Another requirement is controlling the proprietary software of the commercial device automatically during the attack. Typically the commercial de-vices do not contain a standardised interface, excluding the so-called smart cards. This means that it is difficult to use above SCA-tools with commercial devices and the custom SCA tools have to be built.

The aim was to crack a commercial hardware AES-256 ECB encryption device. The method used to perform the attack was electromagnetic side-channel attack. The target was to find the AES-256 secret key. It was unknown if there were any countermeasures on the device. The security of a commercial device was tested against side-channel attack and our own SCA-tools were developed at the same time.

Addonics CipherUSB AES-256 ECB device was selected as a target, illustrated in Figure 1. The device is connected between a PC and USB memory. It is possible to encrypt or decrypt data with the device to the USB-memory or the network.

The analysis of the device was started with physical observations and the supplied specification was read. After removing the shielding, the encryption chip and the power supply lines of the device were identified. The encryption chip of another copy of the device was cut open and inspected under a microscope in order to gain insight of the target areas for EM leakage. The device operation was tested and USB traffic was analysed. Operating the device was dependent on proprietary software supplied by vendor which restricted our examination.

The analysis identified that the data includes a header and a payload. The payload includes several 16 bytes AES blocks. Each AES block position in the payload was found by using special encrypted files.

The device also appears to have some form of a backdoor mechanism in the data header. The header data structure is introduced in Section 2. While uncovering the back-door is out of the scope of this research, it is an interesting target for the further examination.

Correlation power analysis (CPA) [1] was used as a statistical attack method. In this attack Matlab was used to perform CPA over the acquired measurements. The power consumption (and thus the EM emission) of the device was modelled using hamming distance metric.

SikuliX machine vision software [11] controls the device proprietary software and NEWAE EM voltage probe [12] was selected for EM measurements (Figure 1).

The research showed that CipherUSB, which is on commercial use, is vulnerable to side-channel analysis, and side-channel protection to the device is needed.

This paper is organised as follows: In Section I, a brief introduction and target is given, continued with analysis of the target device in Section II. This is followed how the traces were recorded in Section III and how correlation power analysis was used to perform the attack in Section IV. And finally conclusions are drawn in Section V.

Figure 1. Performing electromagnetic side-channel attack on a commercial AES-256 device

2 Analyzing the Target Device

The encrypted files were inspected with a hex editor using different inputs and encryption keys supplied to the device. Files were modified in encrypted form and then fed back to the CipherUSB for decryption. Using this method, significant sections of the file structure used by the device were identified. The device creates 512-byte header data followed by the actual encrypted data (payload) as seen in Figure 2. The two unknown fields in the Figure 2 appeared to change randomly and they could be overwritten by arbitrary data without impairing the functionality of the program or correctness of the decryption. They most likely contain metadata related to the encrypted data such as date and time or some user data. While the payload is the most relevant field in the scope of this article, the header was also analysed in more detail.

Any modification to the first 32 bytes of the file cause the program to output "Incorrect recovery password." error. Modifying other parts of the file shows either "Data may have been modified or altered. Aborting decryption process." error, or no error message at all. The first 32 bytes could correspond to the bytes of an AES-256 key after some transformation. The first 16 bytes of this field changed only if the key used by the device was changed. The latter 16 bytes seem to be changing randomly between encryptions. Recovery key is followed by device model, assumed serial number and some other metadata encoded in ASCII. After device info there is a short segment that contains the length of the payload. Following the length of payload there is a short unknown segment. Modifying this part of the file does not seem to affect anything in the functionality. It might contain encryption time and date. There is a short 4 byte field after the short unknown field which is assumed to be a 32-bit checksum. Peculiarly setting all bytes to the same value for each block sets this field to zero. Following the checksum there is another, this time longer, unknown field. This is the last field of the header before the encrypted data. Finding the location of the payload was straightforward just by looking at which part of the file stays the same when data stays the same and matches the length of the input data.

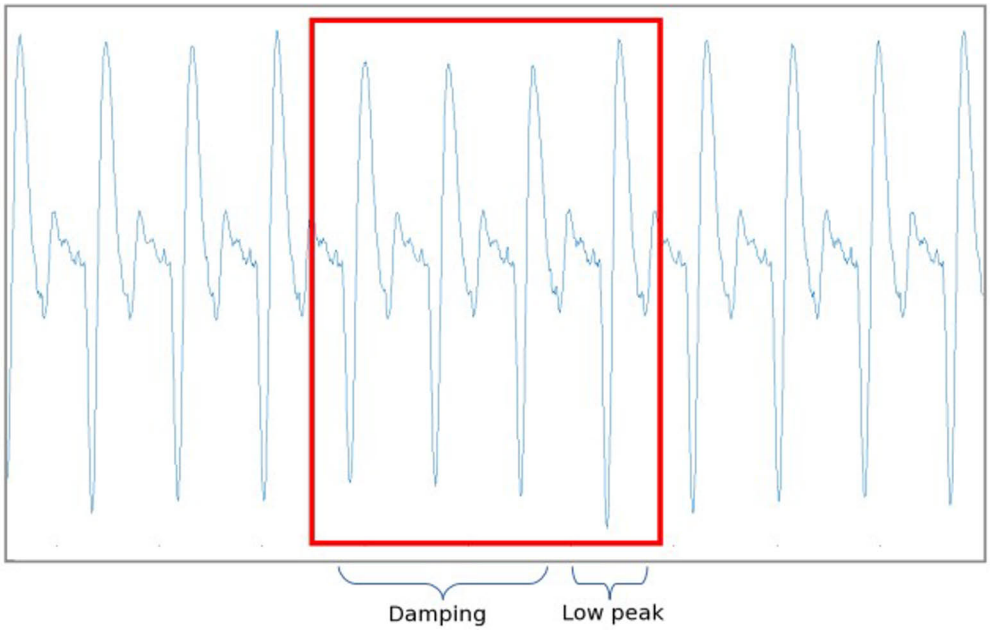

Figure 2. Data fields found in header. Real values overwritten for clarity.

Presence of ECB-mode was verified by encrypting a large bitmap picture. Encrypted bytes of the picture were extracted from the resulting file by stripping the header (first 512 bytes). Bitmap header was then added to the encrypted data and encrypted and unencrypted image were compared with image viewing program. Outline of the original image was clearly visible in the encrypted file as the ECB-mode does not hide large recurring patterns in the data [13].

The file size of the plaintext was found to affect the amount of bursts of EM leakage measured as expected. By trying varying sizes of files it was concluded that each burst corresponded to 512 bytes of data. The AES encryption has a block size of 16 bytes. It follows that there must be 32 block encryption events within a single EM-burst. Positions of the block encryption events were identified by feeding the device specially crafted files sized 512 bytes corresponding to the assumed buffer size. The data bits of the targeted encryption block were set to ’1’, while all other encryption blocks (31) data bits in the file were set to ’0’. Multiple encryptions were performed using the same data and the resulting EM traces were averaged. After around 200 traces the position of the targeted block encryption event started to emerge. Figure 3 showcases the dampening of the amplitude and stronger low peak in the encryption block under testing. Same test was conducted for all block encryption events by moving the position of ’1’ block in the input data. Moving the

’1’ block in the data caused the dampening and low peak to move in the trace correspondingly.

According the user manual, the device is using two-factor authentication. Two-factor authentication provides an additional layer of security and makes it harder for attackers to gain access to files. The password is one factor, something the user knows. The device is second factor, something the user has. Encryption or decryption should not be possible without having both of these, the password and the device. If for some reason the device is lost, encrypted files are not lost because new device can be programmed using the pass-word. There is only one password and it is used to open the device and changing the password.

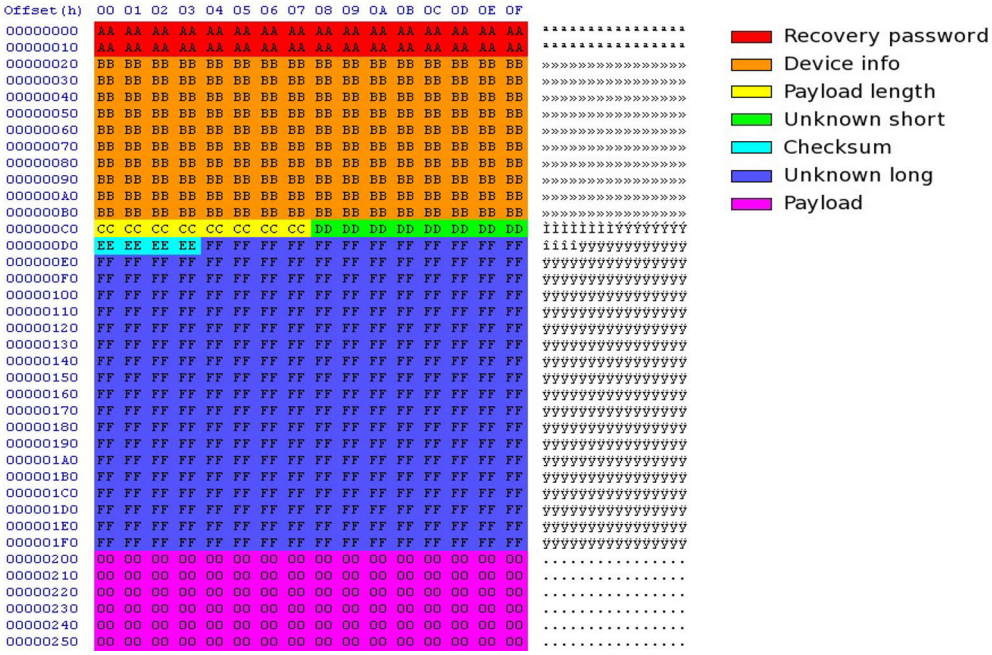

Figure 3. Setting all bits to ’1’ causes the block encryption to differ from surrounding block encryptions. Block encryption 16 out of 32.

This kind of method, that only one password is used, allows several users encrypt/decrypt files easily if they are just using the same password and own separate de-vices. It is therefore concluded, that the device does not actually use two-factor authentica-tion. That is because any device of the same model family, programmed with the same password, can decrypt and encrypt for an unlimited amount of time. There should be an-other password like system administrator password that would

set the AES-256 key of the device. Then it would be two-factor authentication device. Also, there is no way to restore or reset the device. If the password is lost, the device cannot be used anymore.

3 Recording Traces

Before you begin to format your paper, first write and save the content as a separate text file. Complete all content and organizational editing before formatting. Please note sections A-D below for more information on proofreading, spelling and grammar.

SCA attack tools selected were SikuliX machine vision software which controls the device proprietary software, Keysight DSO-S 054A oscilloscope with a voltage probe and a planar H-field EM-probe combined with a low-noise amplifier and isolated power supply by NewAE, a PC and .NET based controller for the oscilloscope (Figure 1) running on the PC.

3.1 Oscilloscope settings and probe position

Running an encryption task of the device caused a voltage drop in the power supply line. The oscilloscope was triggered from the device power supply line. This improved the stability of the EM-leakage signal, compared to triggering directly from the EM-signal.

After experimenting with different oscilloscope settings, it was observed that if the probe positioning is accurate enough, then a 500 MHz sample rate is enough. This re-quires that automatic interpolation setting is turned on to help the alignment of traces. While the strongest point of leakage would seem a natural place to measure the traces, in-stead, the optimal place was on the other side of the chip of the strongest leakage point. The traces on the strongest leakage point and the best leakage point were not distinguish-able to the eyes other than by amplitude. However, the round positions in the correct leakage point were visible to the eyes using special fixed plaintexts and averaging.

A 200 MHz low-pass filter in the oscilloscope was mainly used to pre-process the EM-probe signal. Figure 4 and Figure 5 shows 5000 MSa/s sampling rate sine waves when 200 MHz filter is off and on. Distortion can be seen clearly on signal if low-pass filter is not in use. Applying the low-pass filtering also to the trigger signal improved the accuracy and reduced false trigger events. A small section of the trace gave the best CPA attack results. The first few block encryption events of the traces leaked less than the others. How-ever, it was possible to extract they key from the

device by using only to the first 4 block encryption events i.e. to the beginning of the trace.

Typically, whole 512 bytes file (payload) was recorded, meaning 32 pcs of 16 byte encryptions. Occasionally shorter traces were recorded for increased resolution. For the test a set of 10000 plaintext and 20000 traces was done and the results gave several AES round keys from at least three corners of the chip. The EM-probe is rather big compared to the cryptographic chip of the device, which might be the reason for the result. In the future, it would be better to use a smaller millimeter level EM-probe.

One data burst payload takes about 9 microseconds and consists of 32 encryptions of 16 byte blocks. One encryption takes 9/32 microseconds i.e. 0.28 microseconds. One AES-256 encryption has 14 rounds [14], so each round would take 0.28/14 microseconds i.e. 20 nanoseconds. It means that at least 100 MHz sampling frequency should be used with the device, according the Nyqvist theorem [15].

The interpolation setting in the oscilloscope improved the accuracy of the attack remarkably. Without interpolation it was impossible to find any or just few of the key bytes from a trace set. With interpolation enabled all of the key bytes were uncovered. The interpolation smooths the recorded traces reducing the jitter of signal peaks, allowing more accurate correlation calculations [16].

The oscilloscope settings play a pivotal role to get adequate results. First the signal was measured using normal 8 bit resolution without automatic interpolation, using quite high sampling rate up to 2.0 GHz. After changing the settings to 13-bit resolution and automatic interpolation the results improved significantly, even in the lower sampling rates. Typical recording speed was 1.25 GSa/s. Much higher recording speeds up to 5 GSa/s were also investigated so that automatic interpolation was off-position, but results required significantly more traces.

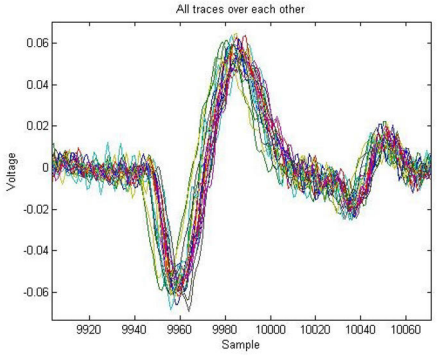

Figure 4. 200MHz filter OFF, unaligned signals

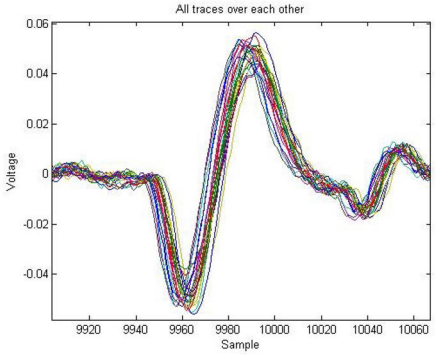

Figure 5. 200MHz filter ON, unaligned signals

Figures 6 and 7, and Figures 8 and 9 show 1.25 GSa/s sine wave heads before and after alignment when interpolation is either off- or on-position. The shape of the signal is rather angular when interpolation is off in addition to which there are outliers. Figure 10 and Figure 11 show 5.0GSa/s sampling rate measurements when interpolation is off-position, not aligned and aligned signal.

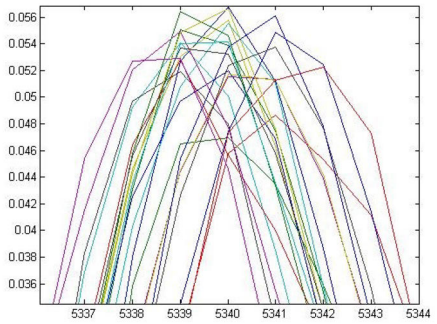

Figure 6. 1.25 GSa/s rec. speed and interpolation OFF, unaligned.

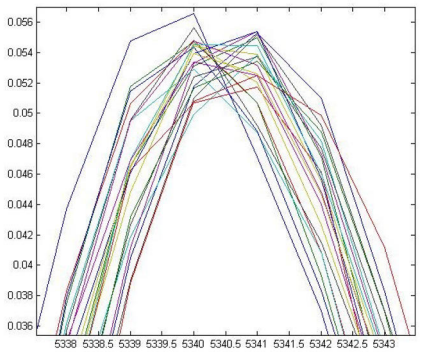

Figure 7. 1.25 GSa/s rec. speed and interpolation OFF, aligned.

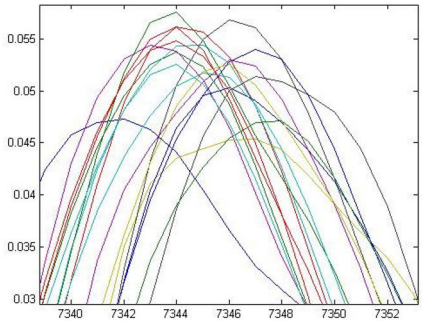

Figure 8. 1.25 GSa/s rec. speed and interpolation ON, unaligned

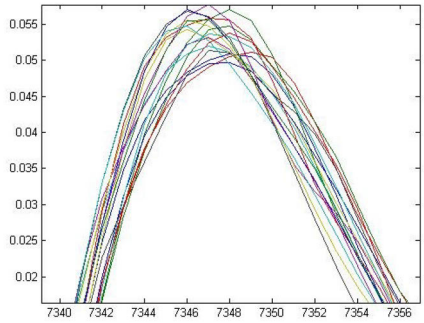

Figure 9. 1.25 GSa/s rec. speed and interpolation ON, aligned.

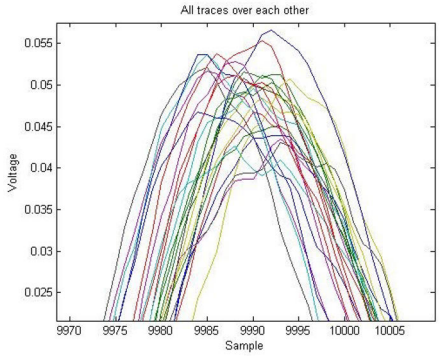

Figure 10. 5.0 GSa/s rec. speed and interpolation OFF, unaligned.

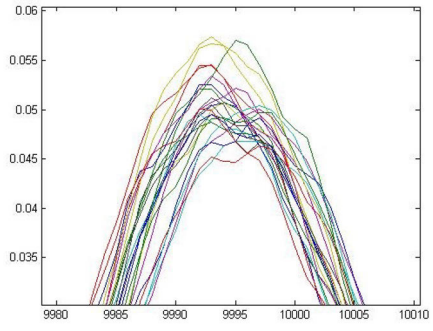

Figure 11. 5.0 GSa/s rec. speed and interpolation OFF, aligned

3.2 Recording process

The device under attack was dependent on the supplied proprietary software featuring a GUI, which hindered the ability to record thousands of traces usually needed for side-channel attacks. The GUI issue was bypassed by creating Python bot leveraging ma-chine vision functions provided by SikuliX. The bot was then used to handle the repetitive operation of the GUI. The bot was coupled with the oscilloscope control program written in C# via local sockets. Oscilloscope control program took care of arming the oscilloscope and storing the recorded data to the disk. Some error recovery capabilities were also added as the synchronization between the bot and

oscilloscope controller enabled more reliable operation and recovery from trigger glitches. This setup allowed the encryption of arbitrary amounts of plaintexts and recording of corresponding EM-traces. The speed of operation was mostly limited by the response time of the GUI. The data files were placed in hierarchical folder structure to avoid unnecessary scrolling in the GUI, which increased the speed of the process.

4 Correlation Power Analysis

The device under attack was running an implementation of AES encryption standard. AES specifies 128-, 192- and 256-byte key lengths and 10, 12 and 14 rounds of operation respectively, with fixed block size of 16 bytes. Each round has an associated round key derived from the original key input for the algorithm. The AES encryption process applies four core operations to the data consecutively each round with exceptions in the first and last round. The core operations are SubBytes, ShiftRows, MixColumns and Ad-dRoundKey. The operations are applied to 4x4 matrix formed from the input data called state. SubBytes is a substitution operation where the input bytes are replaced using special substitution table. ShiftRows shifts the rows 2, 3 and 4 of the matrix by 1, 2, 3 positions respectively. MixColumns performs matrix multiplication with predefined matrix and state in Galois field of two elements. AddRoundKey is just a XOR operation of the state and the key.

In this study Matlab was used to perform the CPA over the acquired measurements. First the EM-measurements were aligned. Simple alignment method was used to synchronize first spike exceeding the threshold. The power consumption of the device was modeled using hamming distance metric. Finding correlation between modeled power consumption and measured traces allows extracting the AES round keys one byte at a time. For AES-256 two last (rounds 13 and 14) round keys are needed to complete the key schedule. Finding the two round keys therefore allows calculation of the full secret key. More details on the most important steps in the attack are described in the following sub-sections. The attack duration could be reduced using other programming languages but is not within the scope of this paper.

For a successful attack several measurements were required to find right measurement setup for oscilloscope and a good position for the EM-probe. In the attack 30 out of the 32 round key bytes were found using side-channel attack and the two remaining were found using brute force. The complexity of using brute force increases exponentially with the number of unknown bytes. Therefore finding more than a few bytes becomes computationally too expensive with the brute force method. The recovered secret key was verified by decrypting some of the ciphertexts.

4.1 Alignment

The traces needed to be aligned due to trigger-signal jitter. A simple alignment method was used to synchronize the first spike over a threshold. It was found that around 60% of maximum signal amplitude gave good results. A Matlab script was used to find the earliest trace in the set and shift other traces accordingly.

4.2 Attack to round 14

Once the measured traces are aligned, the last round key can be attacked. The operation is similar to the attack on AES128, in which obtaining last round key is sufficient to extract the secret key. The attack point used here is at the input of S-box. The difference of the last AES encryption round to the previous rounds is the lack of MixColumns operation, which simplifies the attack. With known ciphertext, the bytes at this point for all possible key-byte values (guesses) can be calculated by performing the AES-encryption algorithm steps in reverse order. The modelled power consumption (hypothesis) is evaluated using hamming distance model, where the assumption is that changes in bit value has an effect on the emitted power. Finding the correlation between the calculated hypotheses and measured traces allows us to evaluate which key byte guesses provide most significant correlation to the measurements, thus serving as key byte candidates. An example of correlation for one key byte is shown in Figure 12. Correlation peak is visible for byte value 28, which is a strong candidate for round key byte value.

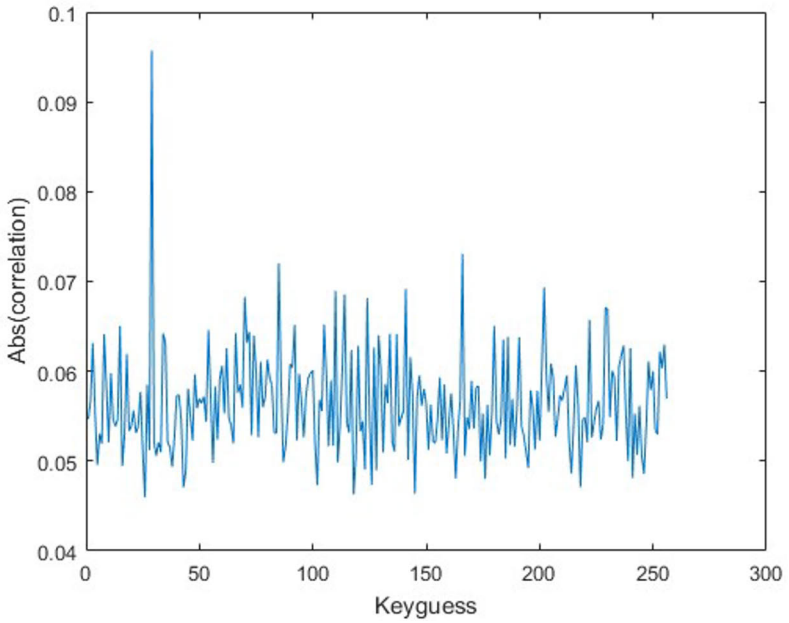

Figure 12. Correlation of different byte values for one round 14 key byte

4.3 Attack to round 13

With the help of the found round 14 key, hypothesis can be calculated further to the input of the S-box in the 13th round. The main difference in the implementation of the 13th round as compared to 14th is the presence of MixColumns function. At first, it would seem that it is not possible to perform the attack byte-by-byte. Luckily, mix-columns is a linear function allowing the separation of the calculation of the hypothesis by using pseudokey [17]. The state at the end of round 13 is

$$X13 = SubBytes^{-}(ShiftRows^{-}(MixColumns^{-}(X14 \oplus K13))) \quad (1)$$

where X14 is the state at the output of round 14 and K13 the 13th round key. As MixColums is linear function i.e.

$$MixColumns(A+B) = MixColumns(A)+MixColumns(B) \quad (2)$$

a pseudokey can be defined as

$$K'13 = ShiftRows^{-}(MixColumns^{-}(K13)) \quad (3)$$

$$X13 = SubBytes^{-}(ShiftRows^{-}(MixColumns^{-}(X14)) \oplus K13) \quad (4)$$

Using this similar attack as for the 14th round can be performed. Once all bytes of the pseudokey are found, it can be transformed back to the real round 13 key:

$$K13 = MixColumns(ShiftRows(K'13) \quad (5,6,7)$$

4.4 Reverse key schedule

Once both 13th and 14th round keys are found, reverse key schedule can be calculated and the original 256-bit secret key obtained. If all byte values are not found, brute force approach can be applied for the uncertain bytes. Using brute force requires calculation of reverse key schedule and decryption attempt for all candidates. That is computationally very demanding if there are several uncertain bytes.

5 Conclusions

Commercial devices are vulnerable to EM side-channel attacks, even when using relatively low cost equipment, in the order of a few thousand Euros. Addonics CipherUSB-device secret key was found in the EM side-channel attack. Using ECB-mode in modern encryption device is a big liability due to weakness of it. The method used to un-cover the positions of the block encryption events could not be used against a device using any pattern hiding mode. All of the encrypted data was decryptable using a computer. Additional data block was also found which might allow decryption without the device, and the header data might include backdoor mechanism.

Side-channel attack might fail if there are not enough traces, we measured up to 10000 plaintexts, which can be done overnight. For a successful attack several measurements were required. Actual attack and analysis took then few hours. Attack might fail also if the power model does not match to actual power consumption or using wrong intermediate values. Incorrect oscilloscope setup or EM-probe positioning could hinder the attack as well. There might also be countermeasures against attacks in a device.

The information leaked in the form of EM emissions enabled the side-channel to find the secret key significantly faster than would have been possible with (impractical) brute-force attack. In consequence, implementations have to be evaluated for their resistivity against such attacks and the incorporation of different countermeasures has to be considered.

CipherUSB supports not only ECB mode, but also CBC mode, and is widely used, with only price differences. In order to show the vulnerability of CipherUSB, it seems necessary to show additional verification in the future for CBC mode targets.

Acknowledgements

The authors would like to thank the Finnish Defence Research Agency (FDRA) CryptoLaboratory in order to fund the project and create deeper ability to evaluate different secure HW implementations. Governments need to be able to say with some level of certainty that crypto implementations are secure. This leads to formal approval and evaluation processes. Evaluation also includes side-channel resistance testing.

This work is part of the FitOptiVis project [18] funded by the ECSEL Joint Undertaking under grant number H2020-ECSEL-2017-2-783162.

List of References

[1] Antti Rantala. Differential power analysis attack against Advanced Encryption Standard. Aalto University, Finland. 2014. Available online: http://urn.fi/URN:NBN:fi:aalto-201412303342

[2] D. Genkin, I. Pipman, and E. Tromer, “Get your hands off my laptop: Physical side-channel key-extraction attacks on PCs,” in Cryptographic Hardware and Embedded Systems—CHES (Lecture Notes in Computer Science), vol. 5, L. Batina and M. Robshaw, Eds. Berlin, Germany: Springer, 2014, pp. 242–260. [Online]. Available: http://dx.doi.org/10.1007/978-3-662-44709-3_14

[3] M. Hutter and J.-M. Schmidt, “The temperature side channel and heating fault attacks,” in Smart Card Research and Advanced Applications (Lecture Notes in Computer Science), vol. 8419, A. Francillon and P. Rohatgi, Eds. Cham, Switzerland: Springer, 2014, pp. 219–235. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-08302-5_15

[4] E. Bangerter, D. Gullasch, and S. Krenn, “Cache games—Bringing access-based cache attacks on AES to practice,” in Proc. IEEE Symp. Secur. Privacy, May 2011, pp. 490–505.

[5] Killing EM Side-Channel Leakage at its Source Debayan Das;Mayukh Nath;Santosh Ghosh;Shreyas Sen 2020 IEEE 63rd International Midwest Symposium on Circuits and Systems (MWSCAS) Year: 2020 | Conference Paper | Publisher: IEEE

[6] A. Zajic and M. Prvulovic, “Experimental demonstration of electro-magnetic information leakage from modern processor-memory systems,” IEEE Trans. Electromagn. Compat., vol. 56, no. 4, pp. 885–893, Aug. 2014.

[7] Rambus DPA Workstation Analysis Platform. Available online: https://www.rambus.com/security/dpa-countermeasures/dpa-workstation-platform/

[8] Riscure Inspector. Available online: https://www.riscure.com/security-tools/inspector-sca

[9] SAKURA hardware security project: Available online: https://satoh.cs.uec.ac.jp/SAKURA/index.html

[10] NEWAE Chipwhisperer SCA tools. Available online: https://www.newae.com/

[11] SikuliX by RaiMan. Available online: http://sikulix.com/

[12] NewAE EM-probe. Available online: http://store.newae.com/probe-set-with-power-supply/

[13] El Fishawy, Nawal F., and Osama M. Abu Zaid. Quality of encryption measurement of bitmap images with RC6, MRC6, and Rijndael block cipher algorithms. IJ Network Security 5.3 (2007): pp. 241–251.

[14] Federal Information Processing Standards Pulication 197, Announcing the Advanced Encryption Standard (AES), NIST, 2001

[15] Nequist frequency. Wikipedia. Available online: https://en.wikipedia.org/wiki/Nyquist_frequency

[16] Sin(x)/x interpolation: an important aspect of proper oscilloscope measurements. EETimes. Available online: https://www.eetimes.com/sinx-x-interpolation-an-important-aspect-of-proper-oscilloscope-measurements

[17] Extending AES-128 Attacks to AES-256. NewAE Technologies WIKI. Available online: https://wiki.newae.com/Extending_AES-128_Attacks_to_AES-256

[18] "The FitOptiVis ECSEL project: highly efficient distributed embedded image/video processing in cyber-physical systems", ACM Int'l Conf. on Computing Frontiers, 2019, pp. 333–338, Available online: https://doi.org/10.1145/3310273.3323437

TURUN
YLIOPISTO
UNIVERSITY
OF TURKU

Painosalama, Turku, Finland 2021

ISBN 978-951-29-8652-1 (Print)
ISBN 978-951-29-8653-8 (Online)
ISSN 0082-6979 (Print)
ISSN 2343-3183 (Online)