Electromagnetic Side-Channel Attack on AES using Low-end Equipment

Oskar Westman[1] and **Martin Hell**[2]

ABSTRACT

Side-channel attacks on cryptographic algorithms targets the implementation of the algorithm. Information can leak from the implementation in several different ways and, in this paper, electromagnetic radiation from an FPGA is considered. We examine to which extent key information from an AES implementation can be deduced using a low-end oscilloscope. Moreover, we examine how the antenna's distance from the FPGA affects the results in this setting. Our experiments show that some key bits indeed can be inferred from the measurements, despite having a far from optimal setting.

Keywords: Side-channel Attack, CEMA, AES, Electromagnetic Radiation, FPGA

1. INTRODUCTION

The security of communications systems heavily rely on cryptographic algorithms to provide confidentiality and integrity to the messages, as well as message authentication to guarantee that the message originates from the claimed sender. The cryptographic algorithms can be either symmetric, where the secret key is shared between sender and receiver, or asymmetric, using a public/private keypair. However, even when asymmetric algorithms are used, data encryption always relies on symmetric algorithms due to performance requirements. Today, the most commonly used data encryption algorithm is the Advanced Encryption Standard (AES), standardized by NIST in 2001 [1].

Attacks on cryptographic algorithms typically focus on weaknesses in the mathematical description, or usage, of the algorithm. In contrast, side-channel attacks is a class of attacks taking advantage of information leaking from the actual implementation of the cryptographic algorithm. Such leakage can stem from e.g., power consumption [2], timing [3,4], acoustics [5,6], temperature [7,8], faults [9,10], or electromagnetic radiation [11,12].

Manuscript received on March 4, 2020 ; revised on April 15, 2020.

Final manuscript received on April 17, 2020.

[1,2] The authors are with the Department of Electrical and Information Technology, Lund Unversity, Lund, Sweden, E-mail: oskar.westman@gmail.com, martin.hell@eit.lth.se.

This work was in part supported by the Swedish Foundation for Strategic Research, grant RIT17-0032.

DOI: 10.37936/ecti-cit.2020142.239925

The side-channel attack model is supported by the well known Kerckhoffs' principle [13], stating, among other things, that a cipher should be secure even if it falls into enemy hands. In other words, the attacker is assumed to have knowledge of both the algorithm description and the implementation and can use this knowledge in an attack.

A circuit will emit an electromagnetic trace due to the switching of the transistors in the circuit, or more specifically, the variation of current in the circuit. This paper focuses on an attack using such side-channels, called electromagnetic side-channel attacks, targeting the well-known AES encryption algorithm.

The interest in, and importance of, electromagnetic side-channel attacks can be attributed to several aspects. It is possible to measure the radiation without being in physical contact with the device, and it is also possible to make measurements from a distance. Moreover, the attacks are difficult to protect against. It has been shown that it is possible to extract AES keys from an FPGA using electromagnetic radiation, using a large number of EM traces together with professional, and very expensive, equipment [14]. In this paper we relax the requirement on the equipment and instead explore how the electromagnetic information leakage can be extracted and exploited using consumer grade equipment. Such equipment is available to a wider range of individuals and organizations, thus allowing the threat to be realized by a wider range of adversaries. Moreover, we assume a non-optimal environment, not shielded from noise from other equipment. This can be seen to simulate a typical lab environment, to which an attacker has temporary access. This is also reflected by the fact that we also assume that the time available for capturing traces is approximately one hour.

Since our setting is not optimized for the attacker, but rather to simulate more restrictive, and in many cases more realistic, conditions, we do not expect performance comparable to more optimal settings. Still, our results show that it is indeed possible to extract key information even in these settings. The information will not lead to full key extraction, but it will lower the number of possible keys to include in a subsequent exhaustive search.

The paper is organized as follows. In Section 2 we give the necessary background on AES and electromagnetic side-channel attacks. Section 3 will present the equipment and the setup used in the experiments, while Section 4 will present our methodology. The re-

sults are given and discussed in Section 5 and related work is discussed in Section 6. The paper is concluded in Section 7.

2. BACKGROUND

This section will provide the necessary background needed to understand the attacks performed in the paper. It will cover the different steps in the attack and also give an overview of the AES encryption algorithm, which is the block cipher subject to our attacks.

2.1 Electromagnetic Side-Channel Cryptanalysis

Several types of side-channels have been exploited for cryptanalysis, e.g., power analysis, cache analysis and other timing information, acoustics, faults and attacks using electromagnetic fields. Side-channel attacks exploiting power traces were pioneered by Kocher et al. in [2], where they introduced simple power analysis (SPA) and differential power analysis (DPA). These attacks were later investigated in more detail by Messerges et al. in [15], targeting smartcards.

Analysing electromagnetic traces for cryptanalysis is very similar to power analysis, for which there is a very good overview in [16]. Instead of analysing the power consumption using power traces, a probe is used to capture the electromagnetic (EM) fields from the processing unit. The captured traces are then called EM traces. Electromagnetic side-channels have the advantage that the attack can be mounted without physically touching the target. The thesis by De Mulder [17] provides are very good introduction to electromagnetic side-channel analysis.

Electromagnetic attacks, similar to power attacks, are divided into simple electromagnetic analysis (SEMA) and differential electromagnetic analysis (DEMA). In SEMA attacks, the electromagnetic traces resulting from running the algorithm are studied in order to infer information. The traces will depend on the executed instructions, which in turn can depend on key bits or other sensitive information used in the cryptographic algorithm. In order to succeed, the attacker must have detailed knowledge about the implemented encryption algorithm. A DEMA attack is much more powerful and can succeed without knowledge of the implemented algorithm. On the other hand, it will require more data in order to be successful. A variant of DEMA attacks are the attacks based on correlation analysis, called Correlation Electromagnetic Analysis (CEMA). These attacks are described in more detailed in the next section.

2.1.1 Correlation Electromagnetic Analysis

As noted above, and already in [2], attacks based on EM traces are very similar to power analysis. Differential electromagnetic side-channel attacks were first explored in [11, 18] and CEMA attacks were developed in [12]. The description of the CEMA attack here is based on the Differential Power Analysis description in [16] and the reader is referred to that book for more details.

The attacker starts by choosing a part of the algorithm to target. The intermediate result is given by Eq. (1),

$$v = f(d, k), \tag{1}$$

where d is a data block and k is a part of the key. The goal is to recover this part of the key.

Then, as a second step, the device, using the secret key, is used to capture EM traces by running the encryption algorithm. For each run, T data points are captured, and this is performed for D different data blocks, $d_0, \ldots, d_{D-1}$. The data points are stored in a $D \times T$ matrix $\mathbf{T}$. Note that one column of this matrix will be EM values for exactly the intermediate result that we are targeting, using the key we wish to recover, see below. Our problem is of course that we do not know exactly which column this is.

T traces are sampled

$$\mathbf{T} = \begin{bmatrix} t_{0,0} & t_{0,1} & \cdots & & t_{0,T-1} \\ t_{1,0} & t_{1,1} & \cdots & & t_{1,T-1} \\ \cdots\ \textit{Power trace for } d_i & & & & \cdots\rightarrow \\ \vdots & & & t_j & \vdots \\ t_{D-1,0} & t_{D-1,1} & \cdots & & t_{D-1,T-1} \end{bmatrix}$$

Actual power consumption for our chosen intermediate values

The correct column is later determined in step 5 of the attack.

In the third step, a hypothetical intermediate value is computed. For each possible subkey $k_j, 0 \leq j \leq K-1$ and each data block $d_i, 0 \leq i \leq D-1$, the intermediate value $v_{i,j} = f(d_i, k_j)$ is computed. These values are stored in a $D \times K$ matrix $\mathbf{V}$.

All K choices for a subkey

$$\mathbf{V} = \begin{bmatrix} v_{0,0} & v_{0,1} & \cdots & & v_{0,K-1} \\ v_{1,0} & v_{1,1} & \cdots & & v_{1,K-1} \\ \vdots & & & v_j & \vdots \\ v_{D-1,0} & v_{D-1,1} & \cdots & & v_{D-1,K-1} \end{bmatrix}$$

Intermediate values for the key k' that is actually used in the device.

Next, in step 4, the values $v_{i,j}$ are mapped to EM values $h_{i,j}$. Using the observation that the electromagnetic radiation is related to the data value, a model can be made. Examples of models are Hamming weight of the data block, or the Hamming distance between the data block and some computation

of the data block that is performed by the algorithm. The Hamming distance is a commonly used model since bit changes typically result in radiation. The results $h_{i,j}$ of this mapping are stored in a $D \times K$ matrix $\mathbf{H}$. Note now that one of the columns in $\mathbf{H}$ will be correlated to the real EM values. This column is the one that corresponds to the correct key k'.

$$\mathbf{H} = \overbrace{\begin{bmatrix} h_{0,0} & h_{0,1} & \dots & & h_{0,K-1} \\ h_{1,0} & h_{1,1} & \dots & & h_{1,K-1} \\ \vdots & & & h_j & \vdots \\ h_{D-1,0} & h_{D-1,1} & \dots & & h_{D-1,K-1} \end{bmatrix}}^{\text{All } K \text{ choices for a subkey}}$$

Values correlated to the real power consumption used when computing intermediate values using key k'.

Now, recall that one of the columns in $\mathbf{T}$ will include the EM data for the intermediate value, when the actual key, k', is used. The final task is to find the columns in $\mathbf{T}$ and $\mathbf{H}$ that are correlated.

Thus, the fifth and final step is to compare the power traces in $\mathbf{T}$ with the hypothetical values in $\mathbf{H}$. An example of how to perform this, which will be used in this paper, is Pearson's correlation coefficient, given in Eq. (2),

$$r_{i,j} = \frac{\sum_{d=1}^{D}(h_{d,i} - \bar{h}_i) \cdot (t_{d,j} - \bar{t}_j)}{\sqrt{\sum_{d=1}^{D}(h_{d,i} - \bar{h}_i)^2 \cdot \sum_{d=1}^{D}(t_{d,j} - \bar{t}_j)^2}}, \quad (2)$$

where $\bar{h}_i$ and $\bar{t}_j$ are the mean values of the columns h_i and t_j respectively. In the attack, it is assumed that the value $r_{i,j}$ with highest correlation will correspond to the column i of $\mathbf{H}$ that are the intermediate values corresponding to the actual key used in the device. Moreover, it will correspond to the column j in $\mathbf{T}$ that contain the EM data for the chosen intermediate value. This allows us to find the (sub)key used to compute the intermediate value.

2.2 AES

This section will give a brief overview of the AES algorithm. For details, refer to e.g., [19]. AES was standardized by NIST in 2001 [1], as a much more secure alternative to DES and more efficient and secure alternative to 3DES. It is based on the Rijndael block cipher (just keeping 128-bit block sizes), and has found its way into a huge number of protocols and implementations. It is today the preferred, and most widely used, block cipher for general purpose encryption.

AES is an SP-network consisting of a key expansion function, an initial round just adding the first round key, then 9, 11, or 13 AES rounds, and lastly a final round, as depicted in Fig. 1. The number of rounds depend on the key size. A 128-bit key uses in total 10 rounds (9 + final round), while 192- and 256-bit keys uses 12 and 14 rounds respectively. The round functions, called SubBytes, ShiftRows, MixColumns, and AddRoundKey, operate on a matrix of 4×4 bytes. The SubBytes step is an application of an S-box applied to each of the 16 bytes. The ShiftRows will cyclically shift row $i, 0 \leq i \leq 3$, i steps to the left. The MixColumn step will linearily transform each column by multiplying it with a fixed polynomial in GF(2^8). Finally, the AddRoundKey function will take a subkey, which is the size of the state, from the key expansion, and binary XOR it with the state.

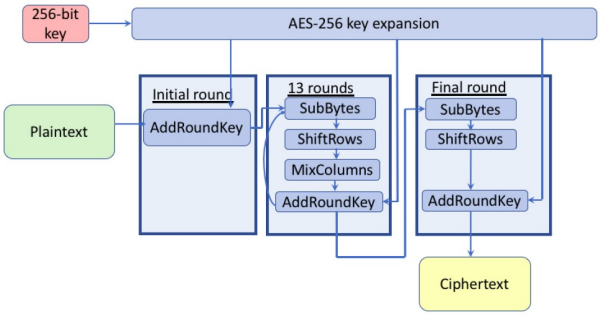

Fig.1: *Overview of the AES encryption algorithm.*

Of particular interest to this paper is the fact that the first round key is exactly the first 128 bits of the key.

3. EQUIPMENT AND SETUP

The FPGA used in the experiments is a Xilinx Artix-7 mounted on a Nexys 4 board.

The oscilloscope used for measuring and sampling the EM traces is a RIGOL DS1054Z with 50 MHz bandwidth. This is an entry level digital oscilloscope with a price tag of about $350, making it easily accessible to basically any organisations and hobbyists. The oscilloscope is connected to a PC, using LabView to control the oscilloscope and to gather the data. The internal system clock of the FPGA is 100 MHz, which is too fast for the oscilloscope's bandwidth according to the Nyquist theorem. A clock divider was used to encrypt at 12.5 MHz instead.

To extract the electromagnetic signal, a loop antenna was used. The FPGA will create small current loops, and extracting the magnetic field from these current loops require a loop antenna which can encapsulate the currents on the FPGA. A larger diameter on the loop antenna will will result in more extracted signals, but they will instead be weaker in amplitude. It is well known, see e.g., [20], that the optimal relationship between the radius of the loop antenna, r_l,

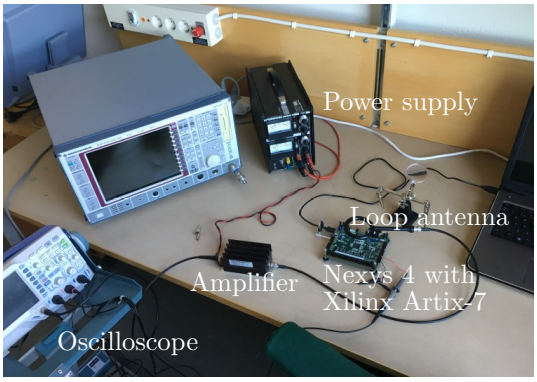

Fig.2: *Overview of setup.*

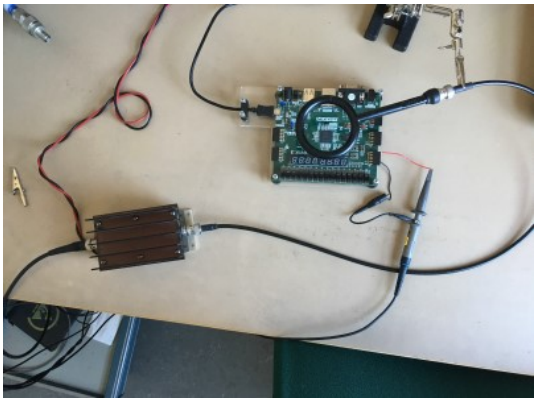

Fig.3: *The measurement setup with a 5 cm loop antenna.*

and the distance r_d to the circuit is

$$\lim_{r_d \to 0} \frac{r_l}{r_d} = \sqrt{2}. \tag{3}$$

Using Eq. (3), for measurements directly above the FPGA, we use a loop antenna with diameter 1 cm. With longer distance r_d, the antenna needs a larger diameter for best performance. For those cases we use a loop antenna with 5 cm radius. Due to the weak signals, an amplifier was needed to amplify the signals before entering the oscilloscope. Using the amplifier, we were able to get signals in the order of millivolts.

A VHDL wrapper is used to encapsulate the AES implementation. It sends the plaintext (data) to the encryption algorithm and sends a trigger signal to the oscilloscope every time an encryption starts. The trigger signal is needed to align the EM traces captured by the oscilloscope. In a real-world scenario, such a trigger signal would not be available. Then traces can instead be aligned using pattern matching and the fact that e.g., certain parts of the algorithm induce a trace that can be used for alignment. There are several methods proposed for alignment even in the case when the manufacturer deliberately inserts random delays in order to misalign traces, see e.g., [21–23], which are described in the context of power analysis.

An overview of the setup is depicted in Fig. 2 and a more detailed view of the measurement setup is given in Fig. 3.

4. METHOD

The target intermediate value chosen for the experiments is the output of the SubBytes operation in the first round of AES, and plaintext is used as data. The nonlinearity of the S-boxes will induce enough transistor switches to generate EM emissions from the circuit. The AddRoundKey operation preceding the SubBytes operation is linear and will not generate enough EM emissions for the used equipment to detect. The first round is also the easiest one to target since the bytes in the start of this round does not depend on all other bytes in the data, as is the case in later rounds. Similarly, using the ciphertext as data, it is easiest to target the last round, using the inverse of the S-box to compute the intermediate values.

Targeting the S-box in the first round, we can write the intermediate value as Eq. (4),

$$v = f(d, k) = Sbox(d, k), \tag{4}$$

where d is one byte of the plaintext and k is one byte of the round key. Retrieving this will immediately give 8 bits of the encryption key. In the simulations, we use an AES implementation retrieved from an industry partner, producing military grade solutions for encryption.

Continuing the steps from Section 2.1, the traces in the second step are collected by running the encryption algorithm and using the trigger signal from the FPGA. The trigger signal is used to signal to the oscilloscope when to start capturing EM emissions. As long as the signal is high, emissions are captured. In each captured trace, we record $T = 1200$ data points. The data values, i.e., the plaintexts, are generated by an LFSR with a known initial state (seed). This allows us to generate a large number of predictable and repeatable 128-bit data values. A total of 10000 traces were collected, aligned using the trigger signal. This allows us to evaluate the performance of the attack with up to 10000 traces, i.e., plaintexts. The capturing process took around 70 minutes and was chosen since it roughly represents the time an attacker has access to a device during e.g., a lunch break. The matrix $\mathbf{T}$ constructed in this step is thus a 10000×1200 matrix. In order to evaluate how different parameters affect the attack, we simulate the following measurement/attack scenarios.

- **Measurement 1**: A 1 cm loop antenna is used and the emissions are measured directly above the FPGA, i.e., at distance 0 cm.
- **Measurement 2**: A 5 cm loop antenna is used and the emissions are 10 cm above the FPGA.
- **Measurement 3**: A 5 cm loop antenna is used and the emissions are 20 cm above the FPGA.

$\mathbf{RoundKey}_{\text{init}} =$

0x00	0x01	0x02	0x03
0x04	0x05	0x06	0x07
0x08	0x09	0x0A	0x0B
0x0C	0x0D	0x0E	0x0F

Fig.4: *The AES roundkey used in the first round is the first 16 bytes of the key.*

• **Measurement 4**: A 5 cm loop antenna is used and the emissions are 30 cm above the FPGA.

The key used for encryption was fixed to

0x000102030405060708090A0B0C0D0E0F
101112131415161718191A1B1C1D1E1F.

Thus the key matrix used in the initial AES round will be as shown in Fig. 4.

In the third step, we compute the hypothetical intermediate values. Using the LSFR with known seed, we generate 10000 plaintexts, d_i ($D = 10000$). For each 8-bit key k_j and each data byte in the plaintext matrix (16 in total), the value $v_{i,j}$ as given by Eq. (5),

$$v_{i,j} = Sbox(d_i, k_j), \tag{5}$$

is computed using the AES S-box. These are in the fourth step mapped to hypothetical EM traces. Here we use the Hamming distance between the data d_i, which is used as input to the S-box, and the resulting $v_{i,j}$. In CMOS technology, operating at high frequency, the primary EM emissions come from the dynamic state of the transistor. According to the Ampère-Maxwell law, a change in current will create a dynamic magnetic field, inducing a current in a wire, i.e., the loop antenna. The matrix $\mathbf{H}$ will thus be of dimension $10000 \times 256 \times 16$. Since we attempt to recover the key bytes one byte at a time, this can be seen as 16 matrices $\mathbf{H}_1 \dots \mathbf{H}_{16}$ of dimension 10000×256 each.

In the fifth, and last, step of the attack, the hypothetical EM values in the $\mathbf{H}_i$ matrices are compared to the actual traces from the measurements, stored in $\mathbf{T}$. We do this by computing the Pearson correlation coefficient between the columns in $\mathbf{H}_i$ and $\mathbf{T}$. With 1200 measurement values per trace in $\mathbf{T}$ and 256 possible keys in $\mathbf{H}_i$, the matrix $\mathbf{R}_i$ will be of dimension 256×1200, with each row representing the correlation values for one key and the 1200 measurement values. Based on the assumption that one of the 1200 measurements will be correlated to the hypothetical values, each row is represented by the highest correlation value. This value is then seen as the correlation value for that key, and the keys are sorted by their correlation value. Now, since we know the correct key used in FPGA, we can determine the ranking of this key, among all 256 possible keys.

4.1 Performance Metrics

The "rank" of a key is used to determine the performance of our experiments. The rank is here defined as the index of the correct key when sorting all keys in order of correlation coefficient given by Eq. (2), with highest correlation having index 0.

A common metric for determining the success of CEMA attacks is to use the Guessing Entropy (GE) [24]. The Guessing Entropy is the expected number of guesses that the attacker needs to make after the collection of side channel information. The optimal strategy for guessing the correct key is to iteratively guess the key with the next highest correlation coefficient. When the correct key is ranked at index i, i.e., $i = \text{rank}(k')$ the number of key guesses is given by $i + 1$. The guessing entropy is then given by

$$GE = \sum_{i=0}^{K-1} (i+1) \Pr(i = \text{rank}(k')). \tag{6}$$

The Partial Guessing Entropy is a similar metric, but focusing only on one subkey. We generalize this slightly and here define the Partial Guessing Entropy over a subset γ of the subkeys, PGE^{γ}. For sample size k samples, $k = 1000, 2000, \dots 9000$, the average key rank is computed by taking the average key rank over the 10 folds of the samples. In other words, we take the average for the k samples starting at $j * 1000$ and ending at $j * 1000 + k - 1 \bmod 10000, 0 \leq j \leq 9$. For $k = 10000$ samples, the key rank is given by just using all 10000 samples. This will give an *expected* key rank for each subkey.

5. RESULTS

In this section, we present the results from the measurements. Several measurements were made and the numbers shown in this section should be seen as representative for the measurements (they are taken from one specific measurement). The general pattern was consistent and conclusions will be drawn from these patterns.

Recall that, in the first measurement, we recorded traces using a 1 cm loop antenna placed directly on the FPGA. In measurements 2-4, a 5 cm loop antenna is used at distances 10, 20, and 30 cm respectively. The total key entropy is computed according to Eq. (6). The result for all measurements are given in Fig. 5.

As can be seen in Fig. 5, all distances gives approximately the same result with only 1000 samples.

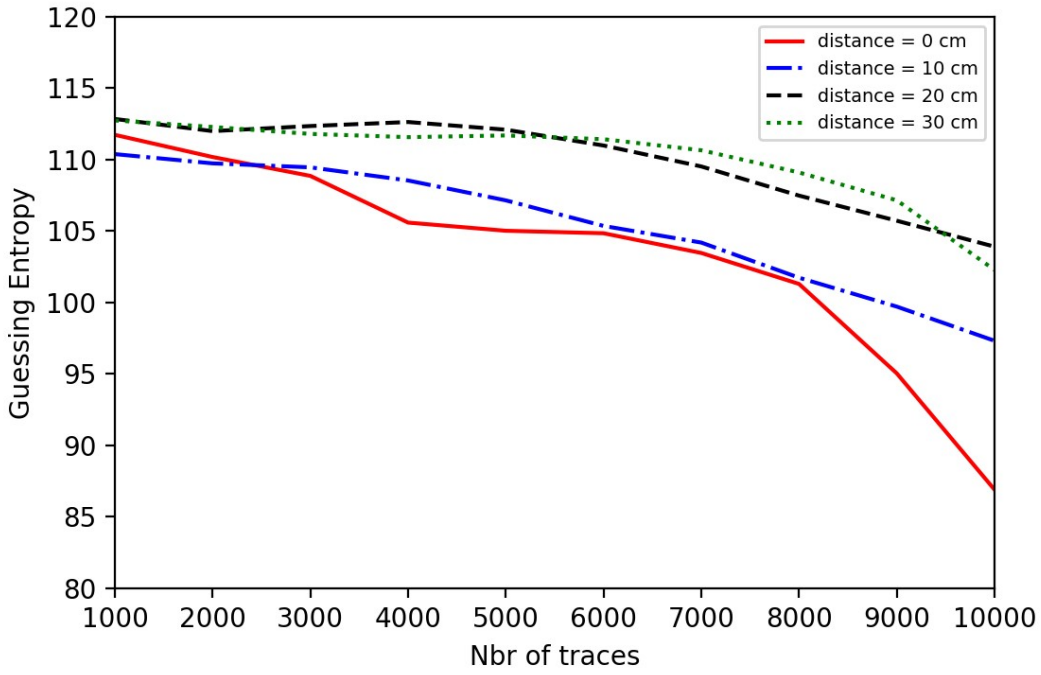

Fig.5: *The resulting guessing entropy ($\log_2$) as a function of the number of samples when measuring at different distances.*

Table 1: *The expected Partial Guessing Entropy for the different subkeys for the 0 cm distance measurements.*

Subkey	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
0x00	154.5	151.3	187.2	154.4	159.7	187.7	193.0	203.4	200.2	192.0
0x01	147.3	97.8	73.5	57.4	45.0	32.5	23.6	15.2	8.0	3.0
0x02	88.4	81.1	67.3	63.7	81.3	93.3	105.4	113.1	110.9	124.0
0x03	103.2	88.8	69.7	46.7	45.2	26.9	26.2	19.7	7.5	1.0
0x04	127.8	104.9	79.8	56.7	35.3	35.3	22.3	18.1	10.9	9.0
0x05	93.0	84.3	48.3	44.1	81.1	110.0	138.6	147.2	149.5	139.0
0x06	141.8	149.5	153.6	141.8	142.9	132.8	133.3	126.9	107.4	103.0
0x07	184.5	172.4	202.0	212.7	208.4	210.1	206.7	205.5	189.6	190.0
0x08	136.8	110.4	113.5	129.0	129.5	165.3	186.2	176.6	175.3	174.0
0x09	127.0	150.9	150.6	148.0	158.7	179.8	213.1	217.1	228.4	246.0
0x0A	130.9	133.3	120.5	79.3	85.8	85.5	76.7	64.0	57.1	38.0
0x0B	178.1	164.2	163.3	152.7	146.4	155.8	143.8	145.4	131.6	160.0
0x0C	128.3	125.7	130.7	115.3	116.8	101.5	79.0	59.5	35.5	11.0
0x0D	124.9	136.1	129.1	129.3	117.7	125.9	109.3	102.6	74.5	60.0
0x0E	104.3	88.4	87.3	65.7	38.4	30.8	26.7	27.4	13.3	5.0
0x0F	99.3	110.5	154.9	136.6	134.8	141.8	130.3	109.8	87.6	97.0

With an increasing number of samples, the effect of the distance becomes more clear. Still, it seems that 20 cm and 30 cm gives very similar results.

The results becomes more interesting when looking at specific subkeys. While some subkeys do not leak any information at all, other subkeys leak very much information. Table 1 gives the specific ranking of the different subkeys, while Fig. 6 shows the partial guessing entropy when we isolate the results for the subkeys $\gamma = \{1, 3, 4, 12, 14\}$ (at distance 0 cm).

These five subkeys have an entropy reduction of 24.1 bits, out of the total of 24.8 bits entropy reduction for all keys. Thus, the other subkeys leak very little information, while these 5 subkeys leak very much.

To get an idea of how the correlation coefficient varies for the different key guesses, Fig. 7 plots this distribution for the case of 9000 traces for the fourth key byte, i.e., when 0x03 is the correct key. Note that the figure depicts the correlation in one measurement, in which the correct key had the highest correlation.

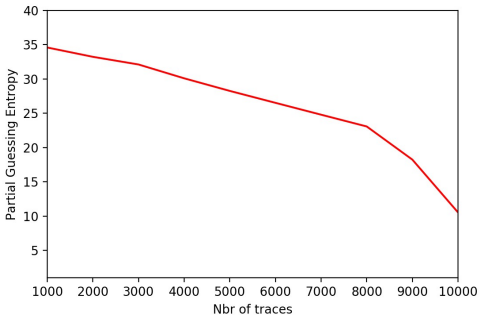

Fig.6: *The Partial Guessing Entropy ($\log_2$) when looking at only subkeys 1, 3, 4, 12, and 14.*

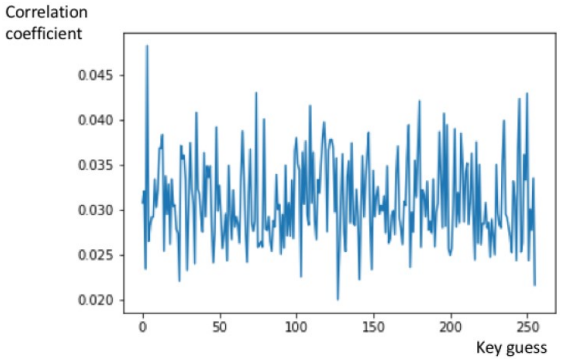

Fig.7: *Correlation coefficient for 9000 traces, key 0x03, and distance 0 cm measurement.*

Performing the attack again, with a new measurement, gives a similar result, but with other keys leaking information. Not knowing which keys have leaked information will add to the complexity of the attack. If we need to guess which 5 subkeys are leaking information, this will require

$$\binom{16}{5} \approx 2^{12.1}$$

guesses. i.e., approximately half of the entropy. Still, the attack is still around a factor 1000 faster than brute force.

We attribute this effect primarily to the relatively low bandwidth of the oscilloscope. The internal system clock at 100 MHz is too fast for the oscilloscopes bandwidth at 50 MHz, according to the Nyquist theorem. Therefore, a 12.5 MHz clock was used as the encryption frequency. The 12.5 MHz is created by dividing 100 MHz with a hardware implemented counter. If the clock is too fast for the oscilloscope (which has 50 MHz bandwidth), the extracted data becomes corrupted and represents wrong values from the encryption.

The fact that different subkeys are successfully extracted in different measurements suggests that the leaked information is not dependent on the actual key value.

6. RELATED WORK

There have been several papers performing differential electromagnetic side-channel attacks on various platforms, targeting different algorithms. The first attacks using DEMA were on smartcards [11]. Later, FPGAs became subject to attacks. The first attacks targeting FPGAs were on elliptic curve cryptography implementations in [25, 26]. DEMA attacks on AES implementations on FPGAs, which is also the focus of this paper, have been previously considered. The first such attack was given by Carlier et al. [27]. They use an ALTERA Cyclone FPGA and a 500 MHz bandwidth oscilloscope. Using 7000 traces, they manage to retrieve some key information by combining the attack with the square attack. Our attack uses a different approach as it is based on correlations using a Hamming distance model. Moreover, we use an oscilloscope that is cheaper and with much lower bandwidth.

The correlation based approach used in this paper (CEMA) were first introduced for correlation power analysis (CPA) in [28] and then used for EM signals in [12]. Their main property is that they use electromagnetic traces that are correlated with a hypothetical EM model. The two most common models are Hamming weight and Hamming distance, as discussed in Section 2.1. A CEMA attack that has a similar approach to that in this paper is the one in [14]. In that paper the authors mount CEMA attacks on AES FPGA implementations. However, they use a custom made board [29] with a Kintex-7 XC7K325T FPGA dedicated to take maximum advantage of the information leaked from the FPGA. Moreover, the oscilloscope used was Agilent DSO6104A, costing around $22,000. Using that board, they are able to extract the secret key using 7000 EM traces. Indeed, this shows the potential power of CEMA when using dedicated and high-end equipment. In this paper, we investigate how much information is leaked when not using dedicated boards, and using a low-end $350 oscilloscope. Another low-cost approach was taken in [30], where the authors used a modified 3D-printer to find the best location of the probe. Such an approach would also be possible in our case, most likely allowing fewer traces for a similar key leakage.

It seems that the CEMA attacks using a Hamming distance model are more efficient than the DEMA attacks that uses a distance of means test. A more recent idea is to use machine learning algorithms in side-channel analysis [31]. This is used to improve the understanding of the device, and in the case of CEMA, to better correlate the electromagnetic measurements with training data. This was proposed in [32]. This is an example of a template attack [33], in which it is assumed that the attacker has access to the device and can analyze it under different keys.

These are also known as profiled attacks and can indeed be very powerful. In [34], the key was extracted using less than 100 LoRaWAN communication packets. However, these uses a stronger assumption than the CEMA attack in this paper, where we only assume that the attacker has access to the device for a short time, and can only capture traces for the specific key under attack.

The recent focus has been on improving the machine learning aspects of side-channel attacks. In particular, deep learning approaches have been used to improve attacks, see e.g., [35, 36] and the recent survey in [37]. Also the attack in [34] took advantage of deep learning. As the attacks becomes more powerful, protection becomes more important. Protections against attacks based on electromagnetic side-channels have recently been proposed in e.g., [38, 39].

7. CONCLUSIONS

In this paper, we have evaluated the possibility of performing a correlation electromagnetic analysis of the AES algorithm implemented on a commonly used FPGA. By relaxing the requirements on the equipment, by only using a consumer grade oscilloscope, we try to better understand to which extent the attack is feasible for a wider range of attackers. Our results show that the implementation does leak information about the key used for encryption. Still, since only one or a few subkeys leak information the complexity of the resulting brute force attack is not improved enough to make these attacks a threat to the confidentiality of the encryption key. It seems clear that a more powerful oscilloscope and more traces are needed in order to reliably obtain enough information to recover the full key in practice. A future work that will give more insight would be to include a larger variety of oscilloscopes for the measurements. This will allow us to better understand the equipment requirements and which resources are needed to successfully perform the attack. Also, a more careful placement of the loop antenna, by trying out several alternatives, could improve the results. In a similar line of research, it would be interesting to vary the background noise and measure its effect.

References

[1] N. I. of Standards and Technology, "Advanced encryption standard," *NIST FIPS PUB 197*, 2001.

[2] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology-CRYPTO99*, ser. LNCS, M. Wiener, Ed., vol.1666. Springer-Verlag, 1999, pp. 388-397.

[3] P. C. Kocher, "Timing attacks on implementations of diffe-hellman, rsa, dss, and other systems," in *Annual International Cryptology Conference*. Springer, 1996, pp. 104-113.

[4] D. Brumley and D. Boneh, "Remote timing attacks are practical," *Computer Networks*, vol. 48, no. 5, pp. 701-716, 2005.

[5] D. Genkin, A. Shamir, and E. Tromer, "Rsa key extraction via low-bandwidth acoustic cryptanalysis," in *Annual Cryptology Conference*. Springer, 2014, pp. 444-461.

[6] M. Backes, M. Dürmuth, S. Gerling, M. Pinkal, and C. Sporleder, "Acoustic side-channel attacks on printers," in *USENIX Security symposium*, 2010, pp. 307-322.

[7] J. Brouchier, T. Kean, C. Marsh, and D. Naccache, "Temperature attacks," *IEEE Security & Privacy*, vol. 7, no. 2, pp. 79-82, 2009.

[8] M. Hutter and J.-M. Schmidt, "The temperature side channel and heating fault attacks," in *International Conference on Smart Card Research and Advanced Applications*. Springer, 2013, pp. 219-235.

[9] D. Boneh, R. A. DeMillo, and R. J. Lipton, "On the importance of checking cryptographic protocols for faults," in *International conference on the theory and applications of cryptographic techniques*. Springer, 1997, pp. 37-51.

[10] M. Joye, A. K. Lenstra, and J.-J. Quisquater, "Chinese remaindering based cryptosystems in the presence of faults," *Journal of cryptology*, vol. 12, no. 4, pp. 241-245, 1999.

[11] J.-J. Quisquater and D. Samyde, "Electromagnetic analysis (ema): Measures and countermeasures for smart cards," in *International Conference on Research in Smart Cards*. Springer, 2001, pp. 200-210.

[12] A. Dehbaoui, V. Lomne, P. Maurine, L. Torres, and M. Robert, "Enhancing electromagnetic attacks using spectral coherence based cartography," in *IFIP/IEEE International Conference on Very Large Scale Integration-System on a Chip*. Springer, 2009, pp. 135-155.

[13] A. Kerckhoffs, "La cryptographie militaire. (French) [Military cryptography]," *Journal des Sciences Militaires*, vol. IX, pp. 5-83, Jan. 1883.

[14] Y. Hori, T. Katashita, A. Sasaki, and A. Satoh, "Electromagnetic side-channel attack against 28-nm fpga device," *Pre-proceedings of WISA*, 2012.

[15] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Investigations of power analysis attacks on smartcards," *Smartcard*, vol. 99, pp. 151-161, 1999.

[16] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards (Advances in Information Security)*. Berlin, Heidelberg: Springer-Verlag, 2007.

[17] E. De Mulder, "Electromagnetic techniques and probes for side-channel analysis on cryptographic devices," *Diss. PhD Thesis*, 2010.

[18] K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic analysis: Concrete results," in *In-*

ternational workshop on cryptographic hardware and embedded systems. Springer, 2001, pp. 251-261.

[19] J. Daemen and V. Rijmen, *The Design of Rijndael: AES - The Advanced Encryption Standard*. Berlin, Heidelberg, New York: Springer Verlag, 2002.

[20] W. Aerts, E. De Mulder, B. Preneel, G. A. E. Vandenbosch, and I. Verbauwhede, "Dependence of rfid reader antenna design on read out distance," *IEEE Transactions on Antennas and Propagation*, vol. 56, no. 12, pp. 3829-3837, Dec 2008.

[21] F. Durvaux, M. Renauld, F.-X. Standaert, L. van Oldeneel tot Oldenzeel, and N. Veyrat-Charvillon, "Effcient removal of random delays from embedded software implementations using hidden markov models," in *Smart Card Research and Advanced Applications*, S. Mangard, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 123-140.

[22] D. Strobel and C. Paar, "An effcient method for eliminating random delays in power traces of embedded software," in *Information Security and Cryptology - ICISC 2011*, H. Kim, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 48-60.

[23] J. G. J. van Woudenberg, M. F. Witteman, and B. Bakker, "Improving differential power analysis by elastic alignment," in *Topics in Cryptology -CT-RSA 2011*, A. Kiayias, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp.104-119.

[24] J. L. Massey, "Guessing and entropy," in *Proceedings of the 1994 IEEE International Symposium on Information Theory*, 1994, p. 204.

[25] E. De Mulder, S. Ors, B. Preneel, and I. Verbauwhede, "Differential electromagnetic attack on an fpga implementation of elliptic curve cryptosystems," in *2006 World Automation Congress*. IEEE, 2006, pp. 1-6.

[26] E. De Mulder, P. Buysschaert, S. Ors, P. Delmotte, B. Preneel, G. Vandenbosch, and I. Verbauwhede, "Electromagnetic analysis attack on an fpga implementation of an elliptic curve cryptosystem," in *EUROCON 2005-The International Conference on "Computer as a Tool"*, vol. 2. IEEE, 2005, pp. 1879-1882.

[27] V. Carlier, H. Chabanne, E. Dottax, and H. Pelletier, "Electromagnetic side channels of an fpga implementation of aes," in *Cryptology eprint archive, report 2004/145*. Citeseer, 2004.

[28] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *International Workshop on Cryptographic Hard ware and Embedded Systems*. Springer, 2004, pp. 16-29.

[29] Y. Hori, T. Katashita, A. Sasaki, and A. Satoh, "Sasebo-giii: A hardware security evaluation board equipped with a 28-nm fpga," in *The 1st IEEE Global Conference on Consumer Electronics 2012*. IEEE, 2012, pp. 657-660.

[30] J. Danial, D. Das, S. Ghosh, A. Raychowdhury, and S. Sen, "Scniffer: Low-cost, automated, effcientelectromagnetic side-channel sniffng," *arXiv preprint arXiv:1908.09407*, 2019.

[31] G. Hospodar, B. Gierlichs, E. De Mulder, I. Verbauwhede, and J. Vandewalle, "Machine learning in side-channel analysis: a first study," *Journal of Cryptographic Engineering*, vol. 1, no. 4, p. 293, 2011.

[32] P. Robyns, P. Quax, and W. Lamotte, "Improving cema using correlation optimization," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2019, pp. 1-24, 2018.

[33] S. Chari, J. R. Rao, and P. Rohatgi, "Template attacks," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2002, pp. 13-28.

[34] J. Xu, Y. Tang, Y.Wang, and X.Wang, "A practical side-channel attack of a lorawan module using deep learning," in *2019 IEEE 13th International Conference on Anti-counterfeiting, Security, and Identification (ASID)*, 2019, pp. 17-21.

[35] M. Carbone, V. Conin, M.-A. Cornélie, F. Dassance, G. Dufresne, C. Dumas, E. Prouff, and A. Venelli, "Deep learning to evaluate secure rsa implementations," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2019, no. 2, pp. 132-161, Feb. 2019.

[36] Y. Zhou and F.-X. Standaert, "Deep learning mitigates but does not annihilate the need of aligned traces and a generalized resnet model for side-channel attacks," *Journal of Cryptographic Engineering*, 04 2019.

[37] S. Jin, S. Kim, H. Kim, and S. Hong, "Recent advances in deep learning-based side-channel analysis," *ETRI Journal*, vol. 42, no. 2, pp. 292-304, 2020.

[38] D. Das, M. Nath, B. Chatterjee, S. Ghosh, and S. Sen, "Stellar: A generic em side-channel attack protection through ground-up root-cause analysis," in *Proc. 2019 IEEE Int. Symp. Hardw. Oriented Security Trust*, 2019.

[39] H. Ma, J. He, Y. Liu, Y. Zhao, and Y. Jin, "Cad4em-p: Security-driven placement tools for electromagnetic side channel protection," in *2019 Asian Hardware Oriented Security and Trust Symposium (AsianHOST)*. IEEE, 2019, pp. 1-6.

Oskar Westman graduated from Lund University in 2018 with an MSc in electrical engineering. His master thesis focused on electromagnetic side-channel analysis of cryptographic primitives. Oskar is a former military officer with eight years of service within the Swedish armed forces. Since 2018 he is working within the defense sector, designing electronic systems for extreme environments.

Martin Hell Martin Hell received an MSc in Electrical Engineering in 2002, and a PhD in Information theory in 2007, both at Lund University, Lund, Sweden. Since 2011, he holds a position as associate professor at the Department of Electrical and Information Technology, Lund University. His research interest is software security, vulnerability assessment and cryptography. He is co-author of the Grain family of stream ciphers where one version is now a standard (ISO/IEC DIS 29167-13). He has published about 60 papers in journals and peer-reviewed conferences and holds 5 patents. He has served as PC-member in numerous international conferences.