Journal of Cryptographic Engineering
https://doi.org/10.1007/s13389-024-00347-3

RESEARCH ARTICLE

Amplitude-modulated EM side-channel attack on provably secure masked AES

Huanyu Wang[1,2]

Received: 28 September 2022 / Accepted: 12 February 2024

Abstract
Recently a new type of side channels was discovered, called amplitude-modulated electromagnetic (EM) emanations from mixed-signal circuits. Unlike power analysis or near field EM analysis, attacks based on amplitude-modulated EM emanations do not require the close physical access to the victim device, which makes the attack particularly threatening. However, all existing amplitude-modulated EM attacks on AES focus on implementations of unprotected TinyAES, which is less likely to be used when the implementation is not overly resource constrained. This paper presents the first deep learning based side-channel attack on AES-128 with a Rivain–Prouff masking scheme by using amplitude-modulated EM emanations as the side channel. Rivian–Prouff masking scheme is a provably secure higher-order masking scheme for AES. To bypass the theoretical strength of the addition-chain based Boolean masked *SBox*, we train neural networks on trace segments corresponding to the *MixColumns* operation in which the data loading instructions for *SBox* output leak information. By comparing two different training strategies, we show that it is feasible to recover the key from an ARM Cortex-M4 CPU implementation of AES-128 with a Rivain–Prouff masking scheme by using the amplitude-modulated EM emanations leaked from the victim device, which has a Bluetooth module embedded on the board.

Keywords Side-channel attack · Amplitude-modulated EM emanations · Deep learning · AES · Rivian–Prouff masking scheme

1 Introduction

Side-Channel Attacks (SCAs) [1, 2] exploit the weakness of physical implementations of cryptographic algorithms by aiming at nonprime, unintentional physical leakage during the execution of algorithms. In the past two decades, power consumption [3–5] and near field EM emissions [6–8] have become two of the most widely and successful exploited side channels. However, these attacks require adversaries to stand close or even have the physical access to the victim device to collect side-channel measurements, which makes the attack less threatening.

Amplitude-modulated EM emanation is a new type of side channel which waves the requirement of the close physical proximity to the device under attack. In 2018, [9] presented the first amplitude-modulated EM emanations based side-channel attack against a Bluetooth device implementation of Advanced Encryption Standard (AES), in which they observed that side channels from an implementation on a mixed-signal chip might be mixed with radio carrier signal and unintentionally transmitted by the on-chip antenna. They experimentally showed that it is feasible to detect the leakage and extract the secret at a much longer distance than attacks based on power consumption and near field EM emissions. By following the work in [9], the template attack in [10] managed to use 5K traces captured at 15 m distance with 1K repetitions to recover a subkey of TinyAES [11], by using a key enumeration up to 2.

Recently, [12, 13] utilized deep-learning techniques to make amplitude-modulated EM side-channel attacks more efficient. By applying models trained on 'clean' traces captured by coaxial cable to classify noisy traces captured at distance to the victim device, [13] is able to achieve a four orders of magnitude improvement over the template attack presented in [10].

✉ Huanyu Wang
huanyu@kth.se

1 School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, China

2 School of EECS, KTH Royal Institute of Technology, Stockholm, Sweden

Published online: 14 March 2024

In addition to AES, [14] presented the first deep-learning based far field EM side-channel attack on Saber Key Encapsulation Mechanism (KEM) [15], which is a finalist of the NIST post-quantum cryptography standardization project. In [16], they experimentally showed that their deep-learning model can recover each bit of the session key with around 90% probability. Afterwards, the presented attack in [16] achieved to recover messages with the probability 100% from the profiling device and with around 74% probability from the victim device.

To the best of our knowledge, all existing amplitude-modulated EM side-channel attacks on AES [9, 10, 12, 13, 17–19] focus on implementations of TinyAES [11]. They do not take into account that TinyAES is unlikely to be used when implementations are not excessively resource constrained. In security-sensitive applications, countermeasures against SCAs are typically applied to protect secret in implementations of cryptographic algorithms. Among all existing countermeasures, masking [20] is the most widely applied one for both software and hardware implementations of AES. In general, a d_{th}-order masking scheme splits the sensitive intermediate value processed by the device into $d+1$ shares, where d is called the order of the mask. Any d among $d+1$ shares are supposed to be random and do not reveal the key-dependent intermediate value. Thus, the correlation between the sensitive intermediate value and side-channel measurements are mitigated. Due to its characteristic of provable security against any attack of order lower than $d+1$, Rivain–Prouff (RP) masking scheme, which is an addition-chain based scheme, became one of the most popular approaches in security-sensitive circuits [21].

In this paper, we present the first result of an amplitude-modulated EM emanations based side-channel attack on implementations of AES-128 with a Rivain–Prouff masking scheme. Our experiments are conducted on an Nordic Semiconductor nRF52832 development board embedded with a Cortex M4 CPU, which supports Bluetooth 5. Our contributions can be summarized as follows:

1. We present the first deep-learning amplitude-modulated EM attack on a Bluetooth device implementation of masked AES-128. We use two different training strategies to build neural networks and show that both approaches are feasible to recover the key.
2. We show that by utilizing the trace segments related to the data loading instructions of the *MixColumn* operation, instead of the commonly used *SBox* segments, we can successfully bypass the strength of addition-chain based Boolean masked *SBox* implementation.

Algorithm 1 Pseudo-code of the *SBox* operation in AES-128 with RP masking scheme [21].

```
// SecSBox
// in: share x_i satisfying ⊕_i x_i = x
// out: share y_i satisfying ⊕_i y_i = SBox(x)
for i = 0 to d do
    z_i ← x_i^2                                                    ▷ ⊕_i z_i = x^2
end for
RefreshMasks(z_0, z_1, ..., z_d)
(y_0, y_1, ..., y_d) ←SecMult((z_0, z_1, ..., z_d), (x_0, x_1, ..., x_d))   ▷ ⊕_i y_i = x^3
for i = 0 to d do
    w_i ← y_i^4                                                    ▷ ⊕_i w_i = x^12
end for
RefreshMasks(w_0, w_1, ..., w_d)
(y_0, y_1, ..., y_d) ←SecMult((y_0, y_1, ..., y_d), (w_0, w_1, ..., w_d))   ▷ ⊕_i y_i = x^15
for i = 0 to d do
    y_i ← y_i^16                                                   ▷ ⊕_i y_i = x^240
end for
(y_0, y_1, ..., y_d) ←SecMult((y_0, y_1, ..., y_d), (w_0, w_1, ..., w_d))   ▷ ⊕_i y_i = x^252
(y_0, y_1, ..., y_d) ←SecMult((y_0, y_1, ..., y_d), (z_0, z_1, ..., z_d))   ▷ ⊕_i y_i = x^254
for i = 0 to d do
    y_i ← Af(y_i)
end for
if d mod 2 = 1 then
    y_0 ← y_0 ⊗ 0x63
end if
```

1.1 Related works

The side-channel security of addition chain based SBoxes has been first studied by Prouff and Rivain [22] and Alexandre Duc et al. [23]. However, instead of using actual physical implementations of RP-masked AES, they illustrate the security evaluation according to theoretical proofs [22] and simulation paradigms [23]. By following the step of [22, 23], [24] evaluates the side-channel resistance of an ARM Cortex-M4 based STM32F407 development board implementation of AES with the first-order addition-chain based masked SBox. They show that by using the power consumption as the side channel, their template attack managed to recover the SBox output by using about 60 traces with a 80% success rate. However, when it comes to the case that using near-field EM emissions as the side channel, the success rate of the template attack in [24] drops to 40% with 1K traces. To further investigate how deep learning can help side-channel attacks on implementations AES with the addition-chain based masked SBox, [25] train a convolutional neural network (CNN) as the classifier to conduct the attack. Under the same conditions as in [24], the trained CNN model in [25] achieved to recover the SBox output by using about 300 power consumption traces with a 100% success rate. However, for the near field EM traces which have much lower signal-to-noise ratio (SNR), the CNN model failed to recover the SBox output. Even though these existing works [24–26] have already investigated the resistance of implementations of AES with addition-chain based masked SBox, they use power consumption or near field EM emissions as side channels. There is still a lack of study on the resistance of implementations of the addition-chain based masked AES against SCAs based on amplitude-modulated EM emanations.

Algorithm 2 Pseudo-code of the secure multiplication in AES-128 with RP masking scheme [21, 27]

```
// SecMult
// in: share a_i satisfying ⊕_i a_i = a, share b_i satisfying ⊕_i b_i = b
// out: share c_i satisfying ⊕_i c_i = ab
for i = 0 to d do
    for j = i + 1 to d do
        r_{i,j} ← random()
        r_{j,i} ← (r_{i,j} ⊕ a_i b_j) ⊕ a_j b_i
    end for
end for
for i = 0 to d do
    c_i ← a_i b_i
end for
for j = 0 to d, j ≠ i do
    c_i ← c_i ⊕ r_{i,j}
end for
```

In this paper, we go one step further by using the amplitude-modulated EM emanations (far field EM emanations) as the side channel to investigate the resistance of implementations of the addition-chain based masked AES. By doing this, we aim to investigate to which extent remote attacks can be a threat to implementations of the addition-chain based masked AES.

The rest of the paper is organized as follows. Section 2 provides background information on AES algorithm with RP masking scheme and reviews how deep learning side-channel attacks work. Section 3 explains why the mixed-signal circuits can generate and transmit EM emissions to long distance. Section 4 presents our experimental setup and shows how we pre-process the captured traces. Sections 5 shows our leakage detection process and how the trace segments for training neural networks are decided. Sections 6 describes the profiling stage to build our deep-learning models. Section 7 summarizes the experimental results and Sect. 8 concludes the paper.

2 Background

In this section, we start by reviewing the AES-128 algorithm and Rivain–Prouff masking scheme. Afterwards, we describe how deep-learning techniques help side-channel attacks.

2.1 AES-128 with Rivain–Prouff masking scheme

The AES [28] is a symmetric encryption algorithm standardized by NIST in FIPS 197 and included in ISO/IEC 18033-3. In general, it uses an n-bit secret key $K, n = \{128, 192, 256\}$ to encrypt a 128-bit block of plaintext $\mathcal{P}$. The output of the AES algorithm is a 128-bit block of ciphertext $\mathcal{C}$. The AES algorithm in our experiment is called AES-128 with the key size set to $n = 128$. There are ten encryption rounds in AES-128 and for each round (except the last round) it repeats the following four steps: non-linear substitution, *SubBytes*, transposition of rows, *ShiftRows*, mixing of columns, *MixColumns*, and round key addition, *AddRoundKey*.

Protecting AES from side-channel attacks has become a realistic concern for cryptographic community. In general, masking [20] is one the most widely used countermeasures which aims to split sensitive intermediate value into random shares, to mitigate the dependency between the sensitive variable and side-channel measurements. Consequently, masked AES is not vulnerable to traditional side-channel attacks which aim to recover the sensitive variable directly from a single leakage segment. When designing a mask scheme to protect a block cipher, addition chain is one of the most widely used approaches to implement masked non-linear transformations (*SBox* operation in AES) which cost less than the conventional lookup-table based approaches in higher-order mask scheme.

Rivain–Prouff masking scheme [21] is the first provably secure addition-chain based approach which supports any order of mask. In an addition-chain based approach, the non-linear operations can be unrolled and expressed as a combination of squares and multiplications over a finite field $\mathbb{F}_{2^n}$. In [21], the nonlinear multiplications are implemented by using the Ishai-Sahai-Wagner's (ISW) scheme [27], as shown in Algorithm 2. Based on the secure multiplication **SecMult()** as shown in Algorithm 2, Algorithm 1 shows the pseudo code of masked *SBox* in [21]. Instead of using a memory lookup table to implement the non-linear substitution operation, the addition-chain based approach unrolls and expresses the SBox as a combination of squares and multiplications over a finite field, as shown in Fig. 1. Afterwards, the **RefreshMasks()** function is described in Algorithm 3, in which we use $Af()$ to denote the affine transformation function.

For the *ShiftRows*, *MixColumns* and *AddRoundKey* operations in RP masking scheme, see [21].

2.2 Deep-learning side-channel attack

Machine learning techniques started helping power based SCAs on AES in 2011 [29]. In general, a deep learning based side-channel attack can be divided into two stages: profiling and attack stage.

During the profiling stage, we assume that the attacker has a full control to at least one profiling device which is identical to the victim device. This means the attacker is able to encrypt a large amount of plaintexts with known keys and record the generated EM traces. Afterwards, the adversary can use the captured traces to train deep-learning models which learn the correlation between the side-channel measurements and the corresponding key-dependent label.

At the attack stage, we assume that the attacker can capture some far field EM traces at a distance to the victim during the execution of the cryptographic algorithm. Then, the

Algorithm 3 Pseudo-code of refreshing masks in AES-128 with RP masking scheme [21]

```
// RefreshMasks
// in: share x_i satisfying ⊕_i x_i = x
// out: share x_i satisfying ⊕_i x_i = x
for i = 0 to d do
    mask ← random()
    x_0 ← x_0 ⊗ mask
    x_i ← x_i ⊗ mask
end for
```

attacker can use the trained deep-learning model to classify these traces and derive the secret key.

3 EM emissions as side-channel

This section describes how the EM emissions are generated, modulated, and transmitted when AES algorithm is executed. We also show how the center frequency of the receiver can be determined to collect these emissions.

3.1 EM emissions in a mixed-signal circuit

A mixed-signal circuit is an integrated circuit that contains both analog part and digital part. A mixed-signal circuit usually costs less materials and has a smaller board size, this satisfies the market demands for cheaper and more portable electronic devices. However, a big problem of integrating digital and analog circuits on the same silicon die is how to handle the noise generated during the execution of instructions in the digital part, since the *RF block* contained in the analog part is extremely sensitive to the noise [30]. In SCAs' contexts, this means that the RF block may also be affected by the side-channel leakage generated by the executions of crypto block in the digital circuit, when the RF block is placed close to the digital circuit. Afterwards, the analog circuit may unintentionally amplify and broadcast the leakage along with the wireless transmission channel.

In our experiments, the encryption operations executed in the *Crypto block* are interpreted as bit flips ($0 \rightarrow 1$ or $1 \rightarrow 0$) controlled by the internal system clock from the *CPU core* since the *Crypto block* is contained in the digital part. Meanwhile, the CPU core in the digital part generates a square wave noise since the frequently switching clock signal. Side channels from the *Crypto block* get modulated by this square wave and then the resulting signal couples with the baseband signal of *Voltage-Controlled Oscillator (VCO)* in the analog part. This effect is because of the substrate coupling [31]. Finally, the *RF block* modulates the signal to a high frequency defined by the wireless transmission protocol and sends it through the antenna. Consequently, it is possible for adversaries to detect the indirect EM emissions at a long distance.

3.2 Center frequency of the receiver

We use f_s to denote the clock frequency of the square wave. The clock signal $s(t)$ can be presented by the formula 1.

$$s(t) = \sum_{n=-\infty}^{+\infty} A_n e^{i2n\pi f_s t}, \tag{1}$$

Formula 2 denotes the corresponding Fourier transform $S(f)$, in which we use A_n to present the Fourier series coefficients. Notice that the even terms of the Fourier series are not exactly equal to zero since the square wave noise is not an ideal square wave.

$$S(f) = \sum_{n=-\infty}^{+\infty} A_n \delta(f - nf_s) \tag{2}$$

In A_n, τ is the duty cycle of the square wave and δ is the impulse function.

$$A_n = \frac{sin\, n\pi\tau}{n\tau} \tag{3}$$

As we mentioned above, the side channel signal from the *Crypto block* $c(t)$ is first modulated by the square wave of the clock signal $s(t)$ and the resulting signal can be described as $c_1(t) = c(t) \cdot s(t)$ in time domain. In frequency domain, the resulting signal is shown below.

$$C_1(f) = C(f) * S(f) = \sum_{n=-\infty}^{+\infty} A_n C(f - nf_s) \tag{4}$$

The modulated signal $c_1(t)$ is then coupled with the digital signal which representing the ciphertext because of the substrate coupling effect and afterwards transmitted to the RF block in the analog part. In this part, the RF block modulates the signal to a high frequency. We use $e^{i2\pi f_c t}$ to denote the radio carrier, in which f_c is the carrier frequency. Then, the modulated signal in time and frequency domains is given by formula 5. Afterwards, the signal is transmitted through the antenna on chip.

$$\begin{aligned} c_2(t) &= \sum_{n=-\infty}^{+\infty} A_n c(t) e^{i2\pi(nf_s + f_c)t} \\ C_2(f) &= \sum_{n=-\infty}^{+\infty} A_n C(f - nf_s - f_c). \end{aligned} \tag{5}$$

At the receiver side, if the center receiving frequency is set to $Nf_s + f_c$, the received signal is given by formula 6. To recover the side-channel signal $c(t)$, the adversary can use a

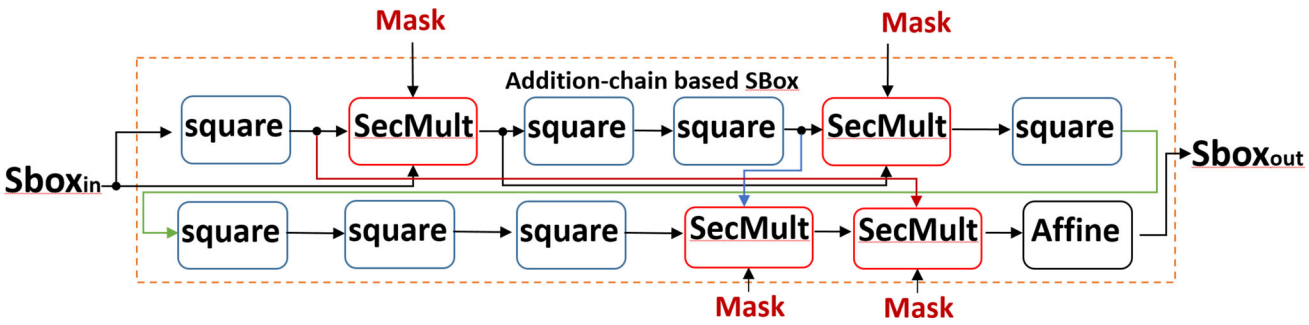


Fig. 1 Illustration of the addition-chain based masked SBox in RP masking scheme

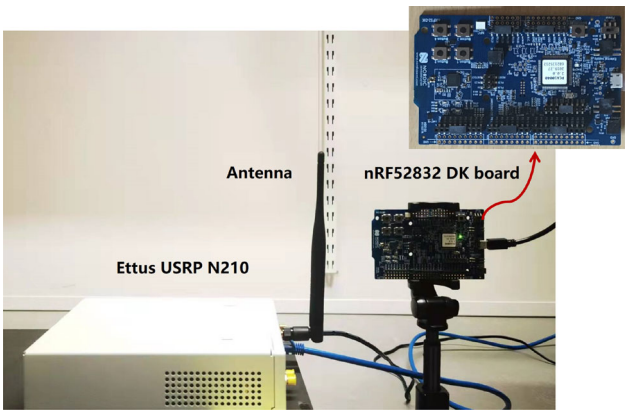


Fig. 2 Experimental setup

low pass filter.

$$\begin{aligned} r(t) &= \sum_{n \neq N} A_n c(t) e^{i2\pi(n-N)f_s t} + A_N c(t) \\ R(f) &= \sum_{n \neq N} A_n C(f - (n-N)f_s) + A_N C(f). \end{aligned} \tag{6}$$

4 Experimental setup

In this section, we describe the equipment we used for capturing and how the captured traces are pre-processed.

4.1 Equipment

Except the antenna, most of the equipment used in our experiments are as the same as in [9, 12, 13]. Figure 2 shows our experimental setup in an office environment.

At the transmitter side, the victim device in our experiments is an nRF52 development kit, which is a versatile single board for Bluetooth Low Energy, Bluetooth mesh, NFC, ANT and 2.4 GHz proprietary development on the nRF52832 SoC. The nRF52832 SoC supports Bluetooth 5 with the data transmission rate 2Mbps and it is built around an Arm Cortex-M4 CPU with floating point unit running at 64 MHz. During the experiment, we implement AES-128 with RP masking scheme on the nRF52832 SoC.

At the receiver side, we use an Ettus Research USRP N210 networked Software Defined Radio (SDR) connected to a vertical antenna VERT2450 with 3dBi gain to receive the signal. The center receiving frequency is $2f_{\text{clock}} + f_{\text{Bluetooth}} = 2.528\text{GHz}$, where $f_{\text{Bluetooth}} = 2.4\text{GHz}$. In our experiments, we use the same sampling frequency as in [13], which is 5Mhz.

4.2 Trace acquisition and pre-processing

In the experiments, we use the nRF52832 device to encrypt plaintext and send the data continuously. At the same time, we use the USRP N210 SNR at the receiver side to capture the amplitude-modulated EM traces. We can find traces corresponding to the AES execution in periodic blocks from the received signal and we use a trigger signal to locate the first round of AES for each AES execution block. The bottom plot in Fig. 3 shows a 7000-point trace segment which represents the the first round of AES-128 with the first-order RP masking scheme. We can see there are two executions of *AddRoundKey* operations from the trace. The patterns within approx. 500–700 points are for the *AddRoundKey* operation before the 1st round and the patterns within approx. 6600–6800 points represent the *AddRoundKey* operation of the 1st round. Patterns within approx. 700–5900 points, 5900–6000 points and 6000–6600 points represent the *SubBytes*, *ShiftRows* and *MixColumns* operation, separately

By following the training approach in [13], the training set used in the profiling stage contains 'clean' traces which are captured by coaxial cable and each trace is the average of 100 measurements with the same encryption. A coaxial cable is capable to transmit signals that oscillate at high RF frequencies without radiating them outside. When the transmitter and the receiver are connected by such a cable, the latter receives the EM emission directly, in the form of the RF signal sent from the RF block on the chip.

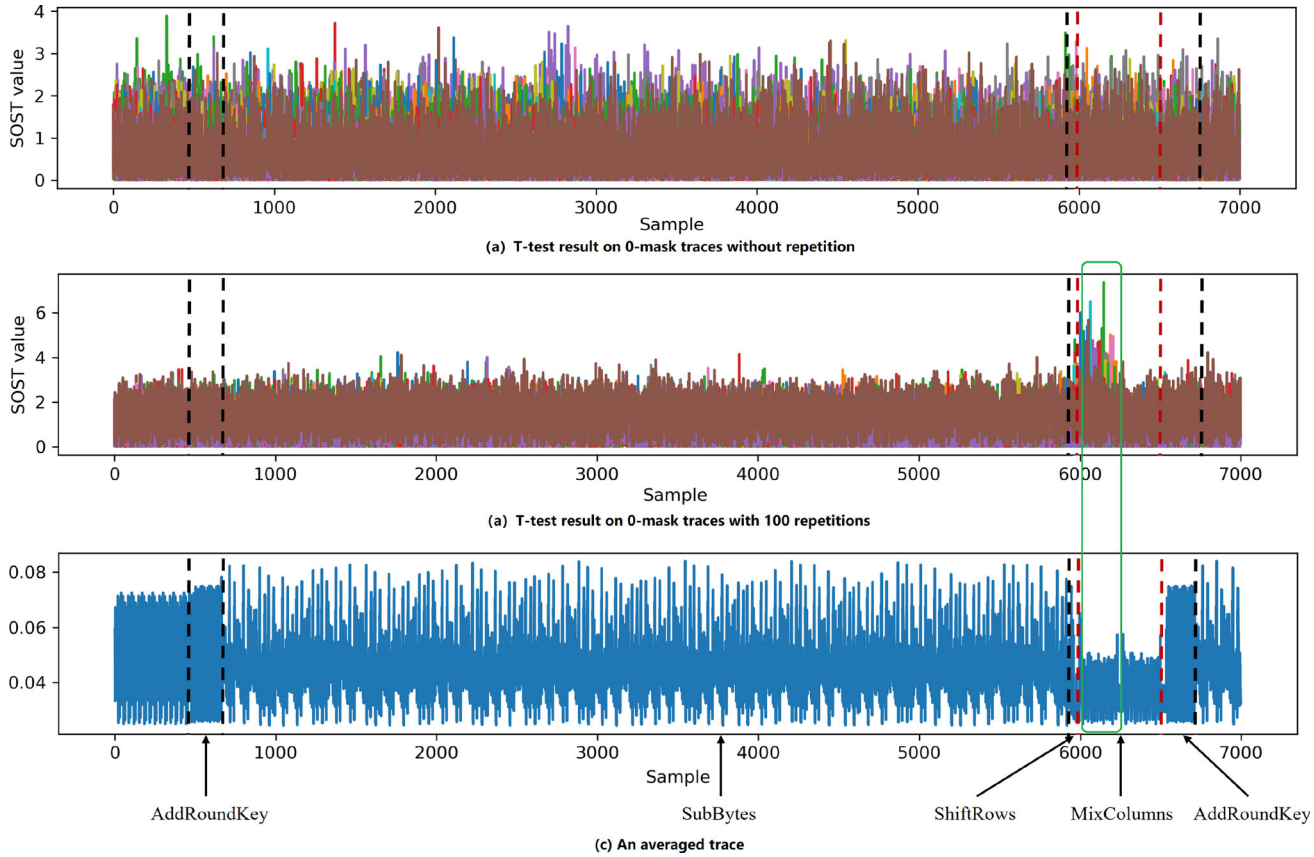


Fig. 3 T-test results on traces of AES-128 with RP masking scheme with all masks set to 0

The scaling is necessary since we followed the profiling strategy in [13]. During the training stage, the model is trained on traces captured by a coaxial cable. At the testing stage, we test our models on traces patured remotely in some experiments. For this reason, traces for profiling and testing may not at the same scale as shown in [13]. A coaxial cable is capable to transmit signals that oscillate at high RF frequencies without radiating them outside. When the transmitter and the receiver are connected by such a cable, the latter receives the EM emission directly, in the form of the RF signal sent from the RF block on the chip.

We use *max-min scaling* [32] to map the amplitude of all traces to the interval [0,1]. Given a set of traces $\boldsymbol{\mathcal{T}}$, each trace $\mathcal{T} = (\tau_1, \ldots, \tau_m) \in \mathbb{R}^m$ of $\boldsymbol{\mathcal{T}}$ is mapped into $\mathcal{T}' = (\tau'_1, \ldots, \tau'_m) \in \mathbb{I}^m$ for all $i \in \{1, \ldots, m\}$,

$$\tau'_i = \frac{\tau_i - \tau_{\min}}{\tau_{\max} - \tau_{\min}}, \tag{7}$$

where $\tau_{\min}$ and $\tau_{\max}$ are the minimum and the maximum data points in $\boldsymbol{\mathcal{T}}$.

5 Leakage detection

In side-channel analysis' context, the task of identifying the leakage interval of side-channel measurements related to the secret-dependent information is called leakage detection. In most side-channel attacks, the first step for adversaries is to find the points of interest (POIs) in traces by using leakage detection methods. Among all these leakage detection methods, the Test Vector Leakage Assessment (TVLA) [33] which is based on the well-known Welch's t-test [34], becomes to one of the most widely used statistical techniques to detect the first-order leakage [14, 16, 35, 36]. We use TVLA to assess the first-order leakage and to find the POIs in far field EM traces.

In a TVLA, side-channel traces are firstly divided into two sets according to the corresponding secret-dependent intermediate value (label) processed by the device. For example, one set $\boldsymbol{\mathcal{T}}_0$ can be traces with $HW(label) > t$ and another $\boldsymbol{\mathcal{T}}_1$ contains traces with $HW(label) < t$. We use $HW(label)$ to denote the Hamming weight of the processed intermediate value $label$, which is the number of 1's in binary representation of $label$. The t-test takes a sample from each of the two trace sets and establishes whether they differ by assuming a null hypothesis that the means of two sets are equal. In a TVLA, the adversaries can compute the the sum

of squared pairwise t-differences (SOST) of two trace sets $\mathcal{T}_0$ and $\mathcal{T}_1$ to assess the leakage [37].

The SOST value of of two trace sets $\mathcal{T}_0$ and $\mathcal{T}_1$ is defined by formula 8. In formula 8, μ_i and σ_i denote the mean and standard deviation of trace set $\mathcal{T}_i$ and n_i represents the number of traces in set $\mathcal{T}_i$, for $i \in \{0, 1\}$

$$SOST = \left(\frac{\mu_0 - \mu_1}{\sqrt{\frac{\sigma_0^2}{n_0} + \frac{\sigma_1^2}{n_1}}} \right)^2 \tag{8}$$

In the following leakage detection process, we use the value of *SBox* input and output in the first round of AES-128 as the attack point to detect the leakage and locate the POIs for training neural networks. The value of *SBox* output is a common attack point for software implementations of AES-128. An attack point is an intermediate value during the execution of a cryptographic algorithm which can be used to describe the side-channel measurements of the victim device.

5.1 Traces with 0-mask

We first run t-test on traces of first-order masked AES-128 with all masks set to 0.

The top picture in Fig. 3 shows the t-test results for each subkey k_i on 0-mask traces without repetition. We use 500K traces in total and traces are divided into two sets: $HW(label) < 4$ and $HW(label) > 4$. From the top picture in Fig. 3, we cannot see any clear leakage on 0-mask traces without repetition and we believe it is because the quality of the trace. Thus, we further do t-test on averaged traces. In [13], it is proved that averaging is a technique that efficiently reduces the external noise and interference.

The middle picture in Fig. 3 shows the t-test results for each subkey k_i on 5K 0-mask traces and each trace is the average of 100 measurements with the same encryption. From the middle picture in Fig. 3, we can clearly see a leakage interval. Notice that in a first-order masking scheme, the value of *SBox* output $SBox(p_i \oplus k_i)$ in the first round of AES-128 is splitted into two shares $s_0(SBox(p_i \oplus k_i))$, $s_1(SBox(p_i \oplus k_i))$, where:

$$SBox(p_i \oplus k_i) = s_0(SBox(p_i \oplus k_i)) \oplus s_1(SBox(p_i \oplus k_i)) \tag{9}$$

Since we set the value of all masks to 0, the second share of *SBox* output is computed as $s_1(SBox(p_i \oplus k_i)) = SBox(0) = 99$ and the first share is consequently derived from formula 9, which is $s_0(SBox(p_i \oplus k_i)) = SBox(p_i \oplus k_i) \oplus 99$.

To locate the leakage interval, we draw an averaged trace in the bottom picture in Fig. 3. In the bottom picture of Fig. 3, the dashed red lines illustrate the beginning and the end of the *MixColumns* operation in the first round of AES. From the assembly code of AES-128 with first-order RF masking scheme, we know that the leakage showed in the middle picture of Fig. 3 is from the first share of the *MixColumns* input, which is loaded to the register. From AES-128, we know the value of the first share of the *MixColumns* input equals to the first share of the *SBox* output. This explains why we can see the leakage by using the value of the *SBox* output as the attack point in the interval related to the *MixColumns* operation.

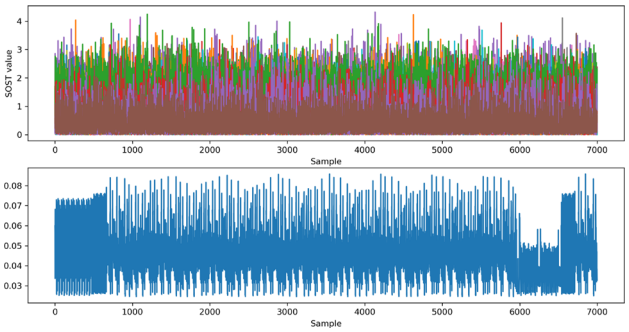


Fig. 4 T-test results on traces of AES-128 with RP masking scheme with all masks set to random

The green box crossing the middle and bottom pictures shows the leakage interval, which is approx. [5999 : 6267] points in the bottom picture of Fig. 3. Close to the leakage interval [5999 : 6267], we can see another interval [6267 : 6535] which contains the same pattern as the leakage interval. These two identical patterns are corresponding to the executions of *MixColumns* for two shares. Figure 5 shows how well the trace segment corresponding to the first share of *MixColumns* fits segment of the second share.

5.2 Traces with random-mask

Next, we run t-test on traces of first-order masked AES-128 with all masks set to random.

Figure 4 shows the t-test results for each subkey k_i on r-mask traces without repetition. We use 500K traces in total and traces are divided into two sets: $HW(label) < 4$ and $HW(label) > 4$. From the top picture in Fig. 4, we cannot see any leakage on r-mask traces. However, unlike the 0-mask case, we cannot apply averaging to r-mask traces since the value of masks are changed for every execution and every trace contains different unknown masks (Fig. 5).

6 Profiling stage

This section describes how we train neural networks at the profiling stage. We use $\mathcal{T} = \{\mathcal{T}_1, \ldots, \mathcal{T}_{|\mathcal{T}|}\}$, where $\mathcal{T}_i \in \mathbb{R}^m$, for $i \in \{1, \ldots, |\mathcal{T}|\}$, to denote a set of traces corresponding to the computation of the first round of AES-128 with first-

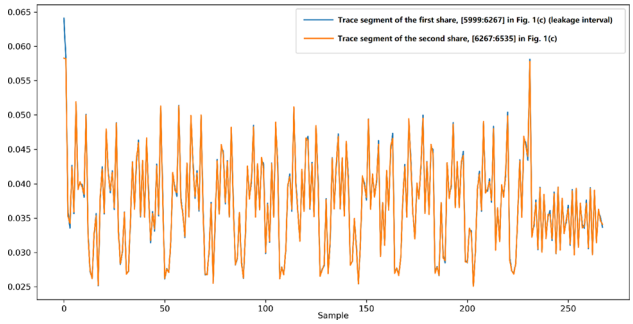


Fig. 5 Trace segment of the first share fits segment of the second share well

order RF masking scheme. In $\mathcal{T}$, traces are captured from the profiling device(s) for randomly generated plaintexts $\mathcal{P}_i \in \{0,1\}^{128}$ and a fixed key $K \in \{0,1\}^{128}$. For kth byte of a 16-byte plaintext $\mathcal{P}_i$, we use $\mathcal{P}_{i,k}$ to represent it, with $k \in \{0,1,\ldots,15\}$.

For a dth-order RP masking scheme, no side-channel attack with order lower than $d+1$ is possible to break it [21]. Thus, to reveal the key from an implementation of first-order masked AES, the deep-learning model is required to deal with two instants to derive the key-dependent value. In our experiments, we compare two different training strategies for utilizing two shares of the sensitive intermediate value.

6.1 Multi-step strategy

In Sect. 5, we show that by using the value of the *SBox* output as the attack point, the leakage interval is approx. [5999 : 6267] points in the bottom picture of Fig. 3. Close to the leakage interval [5999 : 6267], we can see another interval [6267 : 6535] which contains the same pattern as the leakage interval. These two identical patterns are corresponding to the executions of *MixColumns* for two shares. The loading instructions of the first and second share of *SBox* output are the reason of the leakage.

To implement a higher-order attack, the multi-step strategy is to train a neural network on traces with interval corresponding to the first share of a key-dependent intermediate value. At the profiling stage, the adversary can first set all masks to 0 and afterwards derive the first share of *SBox* output by using the recorded plaintexts and keys, which is $SBox(p_i \oplus k_i) \oplus 99$ as discussed in section 5. To train the model, each trace is the average of 100 measurements with the same encryption and labeled by $SBox(p_i \oplus k_i) \oplus 99$.

Afterwards, we apply the trained model to classify traces on interval corresponding to different shares respectively and combine these classification results to derive the key-dependent intermediate value. Figure 6 shows how to use the multi-step strategy to implement a second-order attack.

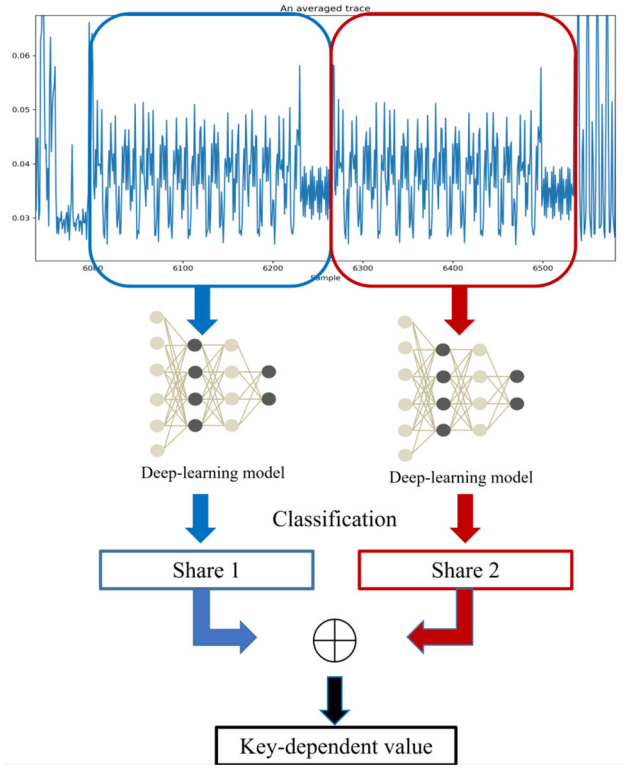


Fig. 6 An illustration of the multi-step strategy for the second order attack

6.2 Single-step strategy

The single-step strategy is to train a neural network on traces with interval contains information related to all shares of a key-dependent intermediate value. Afterwards, we apply the trained model to classify traces on interval corresponding to all shares and let neural network to deal with these shares. Figure 7 shows how to use the single-step strategy to implement a second-order attack.

At the profiling stage, the adversary can first set all masks to a non-zero constant so that the *SBox* output is splitted to two random values. Afterwards, the model is trained on traces interval which is approx. [5999 : 6535] points and each trace is the average of 100 measurements with the same encryption. Since all masks are set to a constant at the profiling stage, we can repeat the same encryption. The label for the single-step strategy is $SBox(p_i \oplus k_i)$.

6.3 Model structure

Architecture of Multi-Layer Perceptron (MLP) used in our experiments for two strategies is shown in Table 1. The Nadam optimizer is used and the learning rate α is 0.00002. From the training set, 20% of traces are randomly selected for validation.

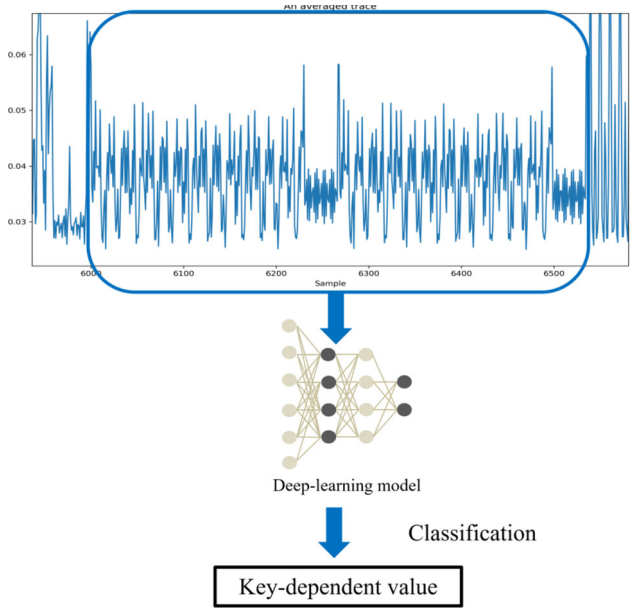


Fig. 7 An illustration of the single-step strategy for the second order attack

Table 1 Model architecture summary

Layer type	Output shape
Batch normalization	(None, 268) (multi-step)
	(None, 536) (single-step)
Dense	(None, 1024)
Dense	(None, 512)
Dropout	(None, 512)
Dense	(None, 256)
Output (dense)	(None, 256)

MLP model called $\mathcal{MLP}_1$ is trained on 100K 0-mask traces captured from the profiling device by using the multi-step strategy. The input size of $\mathcal{MLP}_1$ is 268. Model called $\mathcal{MLP}_2$ is trained on 100K constant-mask traces captured from the profiling device by using the single-step strategy. The input size of $\mathcal{MLP}_2$ is 536. To train both models, each trace in the training set is the average of 100 measurements of the same encryption and with max-min normalization.

7 Experimental results

This section presents the results of our experiments. We first run first-order attacks on traces with all masks set to 0 captured by coaxial cable and at 10 cm distance to the victim. Afterwards, we run the second-order attack on traces with all masks set to random captured by coaxial cable and at 10 cm distance. All attacks were carried out in an office environment. The subkey k_1 is selected as the target subkey to be recovered and the choice of the subkey does not seem to affect the average results. To get the averaged result, we repeat multiple tests for each attack. For each test, we randomly select a subset from the test traces. Instead of using the averaged *Partial Guessing Entropy* (PGE) [38] to derive the averaged number of traces required for an attack, we use the point where the PGE of the real subkey reaches 0 in the majority of tests, which is a termination condition suggested on [13].

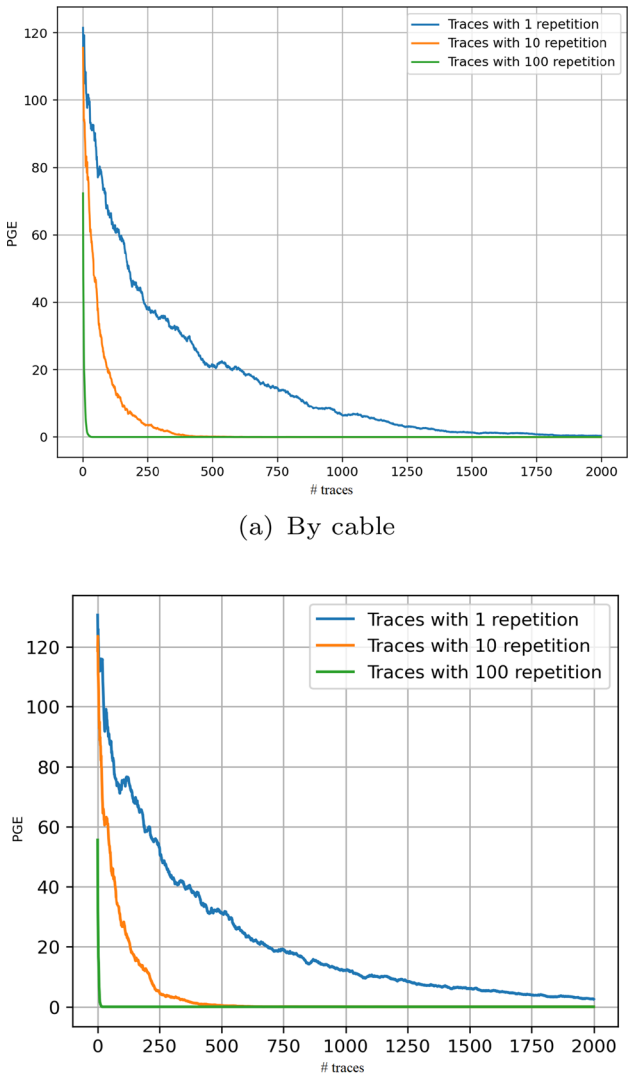


Fig. 8 Average PGE of $\mathcal{MLP}_1$ tested on 0-mask traces captured by cable and at 10 cm

7.1 Results on 0-mask traces

Although 0-mask implementations are impractical in real attack scenario, our experiments on 0-mask traces aim to explore to which extent the mask can make deep-learning

Table 2 Average number of traces required by $\mathcal{MLP}_1$ to recover a subkey from traces with 0-mask captured by coaxial cable and at 10 cm to the victim device (for 100 tests)

Distance to the victim device	# Repetitions		
	$N = 100$	$N = 10$	$N = 1$
Cable	13	227	1285
10 cm	22	300	2806

Each trace is the average of N measurements of the same encryption

Table 3 Average number of traces required by $\mathcal{MLP}_1$ and $\mathcal{MLP}_2$ to recover a subkey from traces with random-mask captured by coaxial cable and at 10 cm to the victim device (for 50 tests)

Model	Distance to the victim device	# Traces
$\mathcal{MLP}_1$	Cable	192,775
$\mathcal{MLP}_2$	Cable	168,152
$\mathcal{MLP}_1$	10 cm	290,848
$\mathcal{MLP}_2$	10 cm	PGE = 22.6/300 K

based attacks less efficient by comparing to the real random-mask cases.

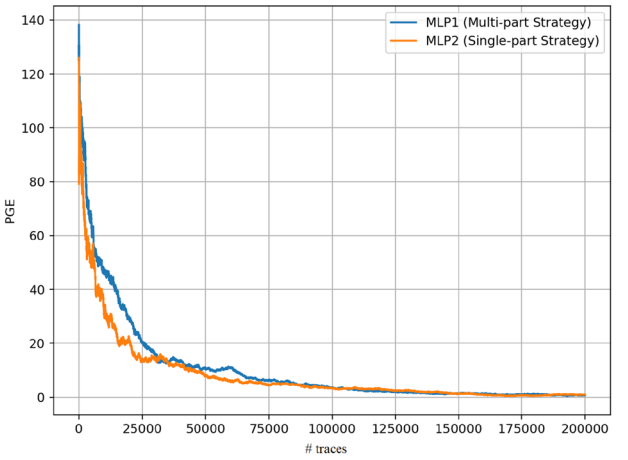
In this experiment, all traces in test sets are with 0-mask. Thus, the key-dependent information is only contained in the first share of the intermediate value. We run the first order attack on traces with segment corresponding to the executions of the first share of *MixColumns* input (*SBox* output) by using $\mathcal{MLP}_1$.

Figure 8 shows the average PGE plots of $\mathcal{MLP}_1$ tested on 0-mask traces captured by cable and at 10 cm. Table 2 shows the average number of traces required by $\mathcal{MLP}_1$ to recover the subeky from traces captured from the victim device by coaxial cable and at 10 cm. For each test, we use 5K traces in total. To compute the average number of traces required by the tested model to recover the subkey, we permute the trace set $\hat{\mathcal{T}}$ 100 times and calculate the point where the PGE result of the correct subkey value reaches 0 in the majority of test sets $\hat{\mathcal{T}}_j$, $j \in \{1, \ldots, 100\}$. From Table 2, we can find that even though our deep-learning model is trained on trace segments corresponding to the *MixColumns* operation to bypass the protection of the addition-chain based masked *SBox*, the model can still recover the key efficiently compared to the results against TinyAES in [9, 10, 12].

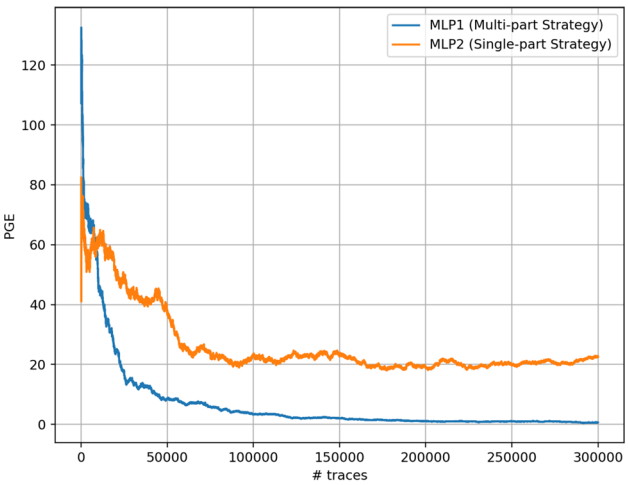
Next, we demonstrate second order attacks on random-mask traces by using both multi-step strategy with $\mathcal{MLP}_1$ and single-step strategy with $\mathcal{MLP}_2$.

7.2 Results on random-mask traces

In this experiment, we investigate the attack efficiency of deep-learning models on traces with random masks captured by coaxial cable and at 10 cm distance. We repeat the first-order attack as presented in the 0-mask case and as we expected, the first-order attack fails. In this scenario, key-dependent information of an intermediate value is split into two random shares. Recovering only one share cannot lead us to derive the key. Thus, it is necessary to run second-order attack as shown in Figs. 6 and 7. Notice that in this experiment, we do not have test sets in which each trace is the average of multiple measurement with the same encryption, since the masks are set to random for each encryption.



(a) By cable



(b) At 10 cm

Fig. 9 Average PGE of $\mathcal{MLP}_1$ and $\mathcal{MLP}_2$ tested on random-mask traces captured by cable and at 10 cm

We first use $\mathcal{MLP}_1$ to run the attack as shown in Fig. 6, to recover the subeky from traces captured from the victim device by coaxial cable and at 10 cm. Afterwards, we use $\mathcal{MLP}_2$ as shown in Fig. 7. For the case that traces are captured by cable, we use 200K traces in total for each test. For the case of 10 cm distance, we use 300K traces for each test. To compute the averaged result, we permute the trace set 50 times and calculate the point where the PGE of the real subkey reaches 0 in the majority of test sets. Figure 8a shows the PGE

(a) The transmitter side.

(b) The receiver side.

Fig. 10 Experimental setup for longer attack distances

plot of $\mathcal{MLP}_1$ and $\mathcal{MLP}_2$ on random-mask traces captured by coaxial cable and Fig. 9 shows the PGE plots of $\mathcal{MLP}_1$ and $\mathcal{MLP}_2$ on traces captured at 10 cm to the victim device.

Table 3 summarizes these results. We can see that when it comes to the case that traces are captured at distance to the victim device, multi-step strategy is a better choice than the single-step one. We can also conclude that, even if we can bypass the theoretical strength of the addition-chain based masked *SBox*, it takes a very large number of traces to recover a subkey. Considering that the implementations in our experiments are with first-order masks, higher-order RP masking scheme seems to be considerably more secure than unprotected AES to attacks based on far field EM emissions.

7.3 Experiments at other distances

Next, we further capture traces at distances of 0.5, 1, 3, 6, 9, 12, and 15 ms from the victim device to investigate whether we can detect the leakage at longer distances, like the experiments on far field EM attacks on TinyAES as shown in [10, 12, 13].

In this experiment, to obtain traces with higher SNR, we change the antenna from the vertical antenna VERT2450 (as shown in Fig. 2) to a grid parabolic antenna TL-ANT 2424B with 24dBi Gain, which is the same antenna used in [10, 12, 13]. Figure 10 shows the overall experimental setup at 15 m distance to the victim device. At the transmitter side (Fig. 10a), the nRF52 device periodically runs AES with the RP masking scheme and transmits the ciphertexts through the on-chip antenna. At the transmitter side (Fig. 10b), we use an Ettus Research USRP N210 SDR with a grid parabolic antenna attached to receive the signal. The center receiving frequency is set to 2.528 GHz. All the traces are captured in an office corridor environment.

We first compare the plots of traces captured at different distances in order to know how traces changes with distance. Figure 11 shows the plots obtained by averaging 100 measurements with the same encryption captured at different distances. No scaling is applied to the traces. The dashed dark blue lines show the zoomed-in view of the trace captured at 15 m from the victim device. It is evident that the signal strength of the EM traces captured by coaxial cable are considerably stronger than traces captured at other distances. But we can still distinguish different executions from the traces captured at 15 m distance. Thus, we believe it is feasible for adversaries to conduct a successful attack on traces captured at a long distances ($\geq$ 10 m) with more traces. Compared with the attack results based on power consumption and near field EM emissions shown in [24, 25], our experiments show a great potential of using amplitude-modulated EM emissions as the side channel to conduct a remote attack on implementations of the addition-chain based masked AES.

Next, we run t-test on traces captured at 0.5, 1, 3, 6, 9, 12, 15 m distance to the victim device with all masks set to 0.

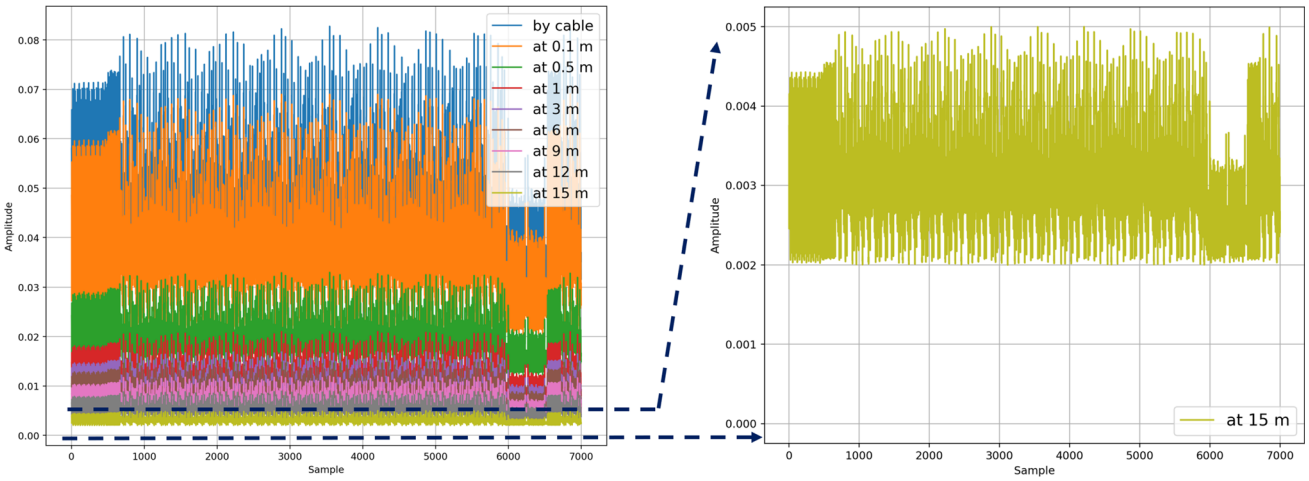


Fig. 11 The plots obtained by averaging 100 measurements with the same encryption captured at different distances

According to the results presented in Section. 5.1, we run the leakage detection process on traces which are the average of 100 measurements with the same encryption. All traces are divided into two sets: $HW(label) < 4$ and $HW(label) > 4$. The t-test results showed that, except for traces captured at 0.5 m from the victim equipment, it is difficult to detect obvious leakage from traces captured at longer distances by using 5K traces with 100 repetitions. Our hypothesis is that more traces are required for identifying the leakage.

In this section, we show that it is feasible to capture traces with clear patterns for different executions at 15 m distance to the implementation of RP-masked AES. This verifies that the maximum measuring distance for far field EM traces is mainly depends on the wireless protocol applied to the embedded device [13], rather than on the cryptographic algorithm implemented on the digital blocks. Considering that it takes a very large number of traces to recover a subkey when traces are captured at 10 cm (see Sect. 7.2), to conduct a successful attack on implementations of RP masked AES at a long distance may require a huge a mount of traces, which is impractical at this stage. Further research may be necessary to improve the efficiency of attacks.

8 Conclusion

We demonstrate the first deep-learning far field EM attack on a Bluetooth device implementation of AES-128 with RP masking scheme. By comparing two different approaches of deep-learning based higher-order attacks, we show the multi-step profiling strategy outperforms the single-step strategy. Our neural network trained on traces from one Bluetooth device can recover the key from random-mask traces captured at 10 cm distance to another device.

The results in our experiments are preliminary and probably can be improved. Future works include mounting similar attacks to break the Bluetooth device at a longer distance and training models for breaking higher-order masked AES.

Funding Open access funding provided by Royal Institute of Technology. This work was supported in part by the research Grant 2018-04482 from the Swedish Research Council and by the Vinnova Competence Center for Trustworthy Edge Computing Systems and Applications at KTH Royal Institute of Technology.

Declarations

Conflict of interest There is no conflict of interest.

References

1. Kocher, P.C.: Timing attacks on implementations of Diffie–Hellman, RSA, DSS, and other systems. In: Annual International Cryptology Conference, pp. 104–113. Springer, Berlin (1996)
2. Kocher, P., Jaffe, J., Jun, B.: Differential power analysis. In: Annual International Cryptology Conference, pp. 388–397. Springer, Berlin (1999)
3. Cagli, E., Dumas, C., Prouff, E.: Convolutional neural networks with data augmentation against jitter-based countermeasures. In: International Conference on Cryptographic Hardware and Embedded Systems, pp. 45–68. Springer, Berlin (2017)
4. Martinasek, Z., Malina, L., Trasy, K.: Profiling power analysis attack based on multi-layer perceptron network. In: Computational Problems in Science and Engineering, pp. 317–339. Springer, Berlin (2015)
5. Maghrebi, H., Portigliatti, T., Prouff, E.: Breaking cryptographic implementations using deep learning techniques. In: International Conference on Security, Privacy, and Applied Cryptography Engineering, pp. 3–26. Springer, Berlin (2016)
6. Benadjila, R., Prouff, E., Strullu, R., Cagli, E., Dumas, C.: Deep learning for side-channel analysis and introduction to ascad database. J. Cryptogr. Eng. **6**, 66 (2020)
7. Picek, S., Samiotis, I.P., Kim, J., Heuser, A., Bhasin, S., Legay, A.: On the performance of convolutional neural networks for side-channel analysis. In: International Conference on Security, Privacy, and Applied Cryptography Engineering, pp. 157–176. Springer, Berlin (2018)
8. Jin, M., Zheng, M., Hu, H., Yu, N.: An enhanced convolutional neural network in side-channel attacks and its visualization, arXiv preprint arXiv:2009.08898 (2020)
9. Camurati, G., Poeplau, S., Muench, M., Hayes, T., Francillon, A.: Screaming channels: when electromagnetic side channels meet radio transceivers. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, pp. 163–177 (2018)
10. Camurati, G., Francillon, A., Standaert, F.-X.: Understanding screaming channels: from a detailed analysis to improved attacks. IACR Trans. Cryptogr. Hardw. Embed. Syst. **6**, 66 (2020)
11. Small portable AES 128/192/256 in c, Github (2013). https://github.com/kokke/tiny-AES-c/
12. Wang, R., Wang, H., Dubrova, E.: Far field EM side-channel attack on AES using deep learning. In: Proceedings of the 4th ACM Workshop on Attacks and Solutions in Hardware Security, pp. 35–44 (2020)
13. Wang, R., Wang, H., Dubrova, E., Brisfors, M.: Advanced far field EM side-channel attack on AES. In: Proceedings of the 7th ACM on Cyber-Physical System Security Workshop, pp. 29–39 (2021)
14. Wang, R., Ngo, K., Dubrova, E.: Side-channel analysis of Saber KEM using amplitude-modulated em emanations. Cryptol. ePrint Arch. **6**, 66 (2022)
15. Jekkea, D., et al.: Saber algorithm specifications and supporting documentation. Secur. Commun. Netw. **2022**, 66 (2022)

16. Wang, R., Ngo, K., Dubrova, E.: Making biased DL models work: message and key recovery attacks on Saber using amplitude-modulated EM emanations. Cryptol. ePrint Arch. **6**, 66 (2022)
17. Zhao, Z.: Far Field Electromagnetic Side Channel Analysis of AES, Master's thesis, School of Electrical Engineering and Computer Science (2020)
18. Wang, R.: Deep learning Based Side-Channel Analysis of AES Based on Far Field Electromagnetic Radiation, Master's thesis, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology (2020)
19. Liu, K.: Far Field EM Side-channel Attack Based on Deep Learning with Automated Hyperparameter Tuning, Master's thesis, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology (2021)
20. Chari, S., Jutla, C.S., Rao, J.R., Rohatgi, P.: Towards sound approaches to counteract power-analysis attacks. In: Annual International Cryptology Conference, pp. 398–412. Springer, Berlin (1999)
21. Rivain, M., Prouff, E.: Provably secure higher-order masking of AES. In: International Workshop on Cryptographic Hardware and Embedded Systems, pp. 413–427. Springer, Berlin (2010)
22. Prouff, E., Rivain, M.: Masking against side-channel attacks: a formal security proof. In: Advances in Cryptology—EUROCRYPT 2013: 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26–30, 2013. Proceedings 32.Springer, Berlin, pp. 142–159 (2013)
23. Duc, A., Dziembowski, S., Faust, S.: Unifying leakage models: from probing attacks to noisy leakage. In: Advances in Cryptology—EUROCRYPT 2014: 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11–15, 2014. Proceedings 33, pp. 423–440. Springer, Berlin (2014)
24. Ming, J., Li, H., Zhou, Y., Cheng, W., Qiao, Z.: Revealing the weakness of addition chain based masked sbox implementations. IACR Trans. Cryptogr. Hardw. Embed. Syst. **66**, 326–350 (2021)
25. Li, H., Ming, J., Zhou, Y.: Assessment of addition-chain-based masked s-box using deep-learning-based side-channel attacks. Secur. Commun. Netw. **2022**, 66 (2022)
26. Ming, J., Cheng, W., Zhou, Y., Li, H.: Apt: efficient side-channel analysis framework against inner product masking scheme. In: 2021 IEEE 39th International Conference on Computer Design (ICCD), pp. 575–582. IEEE, (2021)
27. Ishai, Y., Sahai, A., Wagner, D.: Private circuits: securing hardware against probing attacks. In: Annual International Cryptology Conference, pp. 463–481. Springer, Berlin (2003)
28. Joan, D., Vincent, R.: The design of Rijndael: AES-the advanced encryption standard. In: Information Security and Cryptography. Springer, Berlin (2002)
29. Hospodar, G., Gierlichs, B., De Mulder, E., Verbauwhede, I., Vandewalle, J.: Machine learning in side-channel analysis: a first study. J. Cryptogr. Eng. **1**(4), 293 (2011)
30. Secareanu, R.M., Warner, S., Seabridge, S., Burke, C., Watrobski, T.E., Morton, C., Staub, W., Teilier, T., Friendman, E.: Physical design to improve the noise immunity of digital circuits in a mixed-signal smart-power system. In: 2000 IEEE International Symposium on Circuits and Systems (ISCAS), vol. 4, pp. 277–280. IEEE (2000)
31. Bronckers, S., Van der Plas, G., Rolain, Y.: Substrate Noise Coupling in Analog/RF Circuits. Artech House (2010)
32. Juszczak, P., Tax, D.M.J., Duin, R.P.W.: Feature scaling in support vector data description. In: Proceedings of the Annual Conference Advanced School Computed Imaging, pp. 25–30 (2002)
33. Gilbert Goodwill, B.J., Jaffe, J., Rohatgi, P., et al.: A testing methodology for side-channel resistance validation. NIST Noninvasive Attack Test. Workshop **7**, 115–136 (2011)
34. Welch, B.L.: The generalization of 'Student's problem when several different population varlances are involved. Biometrika **34**(1–2), 28–35 (1947)
35. Sim, B.-Y., Kang, J., Han, D.-G.: Key bit-dependent side-channel attacks on protected binary scalar multiplication. Appl. Sci. **8**(11), 2168 (2018)
36. Elaabid, M.A., Meynard, O., Guilley, S., Danger, J.-L.: Combined side-channel attacks. In: International Workshop on Information Security Applications, pp. 175–190. Springer, Berlin (2010)
37. Gierlichs, B., Lemke-rust, K., Paar, C.: "C.: Templates vs. stochastic methods: a performance analysis for side channel cryptanalysis. In: CHES 2006, Lecture Notes in Computer Science. Citeseer (2008)
38. Pahlevanzadeh, H., Dofe, J., Yu, Q.: Assessing CPA resistance of AES with different fault tolerance mechanisms. In: 2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC), pp. 661–666 (2016)