Blind Side-Channel SIFA

Melissa Azouaoui[1,2], Kostas Papagiannopoulos[1,3], Dominik Zürner[1]

[1] NXP Semiconductors, Germany
[2] Université Catholique de Louvain, Belgium
[3] University of Amsterdam, The Netherlands

Abstract. Statistical Ineffective Fault Attacks (SIFA) have been recently proposed as very powerful key-recovery strategies on symmetric cryptographic primitives' implementations. Specifically, they have been shown to bypass many common countermeasures against faults such as redundancy or infection, and to remain applicable even when side-channel countermeasures are deployed. In this work, we investigate combined side-channel and fault attacks and show that a profiled, SIFA-like attack can be applied despite not having any direct ciphertext knowledge. The proposed attack exploits the ciphertext's side-channel and fault characteristics to mount successful key recoveries, even in the presence of masking and duplication countermeasures, at the cost of both side-channel and fault profiling. We analyze the attack using simulations, discuss its requirements, strengths and limitations, and compare different approaches to distinguish the correct key. Finally, we demonstrate its applicability on an ARM Cortex-M4 device, utilizing a combination of laser-based fault injection and microprobe-based EM side-channel analysis.

Keywords: Fault Injection Attacks, Side-Channel Attacks, Combined Attacks, Statistical Ineffective Fault Attacks.

1 Introduction

The recent years marked the widespread adoption of the Internet of Things (IoT). As a direct consequence, we have observed a tremendous increase in the number of embedded devices used. Such devices often provide critical or sensitive functions, therefore they necessitate robust and mathematically sound cryptographic algorithms to ensure security and privacy. However, implementations of cryptographic algorithms, particularly on embedded devices, remain vulnerable to hardware exploitation. Attacks such as Fault Injection (FI) or Side-Channel Analysis (SCA) can generate and use unintentional information leakages to recover secret data such as cryptographic keys.

Fault Injection attacks (FI) are *active* techniques that aim to inject errors into a target device, during the execution of a cipher. These injections can be accomplished by several tampering means and varying granularity. Clock/voltage glitches can induce fairly coarse faults [17, 25], while focused laser beams [27] can

produce fine-grained errors. Typically, a fault is injected at a specific point in time and then algebraic approaches or statistical distinguishers are used to derive the secret key. The first fault attack was published by Boneh et al. in 1997 [4]. Expanding, Biham et al. [3] introduced Differential Fault Attacks (DFA), collecting pairs of faulty and fault-free ciphertexts and using the combined knowledge of the fault induced and the difference between the ciphertexts to recover the key. Contrary to FI, SCA is a class of *passive* techniques that exploit the unintentional leakage that depends on secret data used during a cipher. SCA accomplishes this by observing physical quantities such as power consumption, electromagnetic (EM) emissions or execution time, and mainly relies on statistical analyses to recover the secret. Despite the apparent differences between active FI and passive SCA, most techniques typically rely on some knowledge and control over the algorithm's plaintext and/or ciphertext. However, so called blind side-channel [19] or blind fault [18] attacks, can be applied to recover secret keys, while requiring little or no knowledge about the plaintext or the ciphertext.

1.1 Related work

The FI literature presents several attack strategies that adapt to various cases of limited knowledge or control over the plaintext and ciphertext. To deal with such restrictions, Fuhr et al. introduced Statistical Fault Attacks (SFA), that only require access to faulty ciphertexts [13]. Towards similar goals, Korkikian, Pelissier and Naccache designed a blind fault attack that does not require the knowledge of either plaintext or ciphertext and is applicable to any round [18]. However, it relies on the possibility to encrypt the same unknown plaintext multiple times, on the observation of the number of faulty ciphertexts and on specific fault models. Should the designer opt for detection-based countermeasures that limit access to faulty ciphertexts, Clavier introduced Ineffective Fault Attacks (IFA) that use ineffective faults (induced faults that do not change the internal state of the cipher) to probe the intermediate values of an algorithm [6].

In addition, it is worth mentioning that the restrictions imposed to the FI adversary can often be bypassed via the usage of side-channel analysis, leading to combined attacks. For instance, Roche, Lomné and Khalfallah present an SCA-assisted DFA that utilizes pairs of valid and faulty ciphertexts and relies on the ability to repeat plaintexts and to observe through side-channel the leakage of the unmasked faulty ciphertexts, while the later are manipulated by the fault detection mechanism [22]. This attack can be prevented if the ciphertexts remain masked during the fault detection procedure. Similarly, Saha et al. propose a technique that requires side-channel leakage of the final comparison but no knowledge of the faulty ciphertexts [23]. It is based on the access to the bytewise HD between the correct and the faulty ciphertext for each fault injection and requires a precise multiple bit-reset or bit-set fault model. Following, they also present another profiled attack where ciphertexts are unknown and the adversary solely observes if a fault is detected or not [24]. Such a binary answer is particularly easy to obtain, simply by observing the high-level protocol which the cipher is a part of. However, it requires to repeat the same unknown

plaintext during the fault profiling phase and the attack phase, and being able to exploit very precise fault effects.

Recently, Statistical Ineffective Fault Attacks (SIFA), which are a combination of SFA [14] and IFA [7] were put forward by Dobraunig et al. [12]. SFA bypasses DFA's requirement for repeated plaintexts and IFA uses only ineffective faults to attack the intermediate values. Since ineffective faults cannot be detected by classic countermeasures, SIFA can exploit many practical implementations protected with common countermeasures against faults.

1.2 Contribution

SIFA is a strong attack due to its minimal requirements on the attacker's capabilities, as it only requires knowledge of the ciphertexts. In this work, we show that this knowledge is not always necessary. In addition, the previously mentioned attacks are prevented if the fault detection is correctly masked or if plaintexts cannot be repeated. Some additionally require ideal side-channel information, while in practice side-channel leakages are noisy. We show in the remainder of this paper that it is possible to bypass both fully masked fault detection mechanisms and the impossibility to repeat plaintexts, at the cost of both side-channel leakage and fault distribution profiling. This result is relevant for applications such as the session key derivation used in the EMV payment scheme [10] or when using any modes of operation that limit the adversary's access to and control of the plaintext and ciphertext.

We combine profiled fault and side-channel attacks to perform SIFA using only side-channel leakage of the correct ciphertexts. The proposed attack allows to bypass both securely implemented fault detection countermeasures and side-channel countermeasures with very limited information, at the cost of preliminary side-channel and fault profiling. **As opposed to the previously mentioned blind fault attacks, this attack does not require any repetition of the plaintext**, thus remains applicable if AES is used in conjunction with a mode of operation that does not allow repetitions. **We stress however that the attack relies on the fact that the correct ciphertext is unmasked after the fault detection countermeasure** since it is an ephemeral secret. We confirm the applied attack both in simulation and in practice. We discuss alternatives in the conclusion. We offer background in Section 2 and describe the new attack in Section 3. The attack is analyzed with simulations in Section 4 and is experimentally verified in Section 5 using a Langer micro-probe and laser-based FI on an ARM Cortex-M4.

2 Background

2.1 Notations

We use capital letters for random variables and lowercase letters for their realizations. Multiple realizations are denoted with indexed sets, e.g. $\{a_i\}_{0 \leq i \leq n}$

or simply $\{a_i\}$. We use sans serif font for functions, e.g. $\mathsf{HW}(\cdot)$ denotes the Hamming weight function. We denote the conditional probability of random variable A given B as $\Pr[A|B]$, we also use Pr to denote likelihoods/densities. We use calligraphic letters to denote distributions (e.g. distribution $\mathcal{D}$) and use $\mathcal{N}(\mu, \sigma^2)$ to denote the Gaussian distribution with mean μ and variance σ^2. We use $\sim$ to denote that a random variable follows certain distribution, e.g. $A \sim \mathcal{N}(\mu, \sigma^2)$. If $A \sim \mathcal{N}(\mu, \sigma^2)$, we state $\Pr[\mathrm{a}] = \mathcal{N}(\mathrm{a}; (\mu, \sigma^2))$ and if $A \sim \mathcal{D}$, we state $\Pr[a] = \mathcal{D}(a)$. We use apostrophes to denote faulted values, e.g. A' is the biased version of A.

2.2 Statistical ineffective fault attacks (SIFA)

Statistical Ineffective Fault Attacks (SIFA) [12] introduced by Dobraunig et al. allow an attacker to bypass many common DFA countermeasures such as redundant computation or infective countermeasures [26]. The authors demonstrate that a fault injected at the input of the MixColumns step of the 9th round introduces a bias on the AES state, which is otherwise uniformly distributed. Knowing the ciphertexts, and by making a hypothesis on 4 bytes of the last round key (2^{32} guesses), it is possible to backtrack to the biased state. Incorrect key hypotheses result in a uniformly distributed state, while for the correct hypothesis the state distribution is biased. Such bias is easily detected using the Squared Euclidean Imbalance (SEI) distinguisher which measures the distance between the obtained hypothetical distributions and the uniform distribution. SIFA is robust against noise introduced by failed fault inductions (for example in dummy rounds), but can be impeded if the probability of inducing an ineffective fault is negligible or if the resulting distribution of ineffective faults is uniform or close to uniform.

While the work of Dobraunig et al. [12] focuses on the penultimate round, they point out that SIFA can be performed on the last round as well, only requiring a hypothesis on a single key byte. However, in this case the distribution of the state is far from uniform for all key byte guesses and thus the 10th-round SIFA must acquire information on the bias. If not known a priori, such information can be learned through a process of fault profiling, typically on an open device that is identical to the device-under-attack. SIFA is also applicable to masked implementations as demonstrated in [11], where it is shown that faulting only one share during the SBox computation is enough to mount an attack.

3 Blind Side-Channel SIFA

This section describes the attack proposed in this paper, named blind side-channel SIFA. The attack focuses on the last (10th) AES round and targets a single key byte, utilizing both SCA and FI. It can be straightforwardly adapted for different symmetric ciphers, as well as SFA techniques. Note that while the attack is blind during the attack phase, it requires a preliminary fault and side-channel profiling.

3.1 Attack context and motivation

This work focuses on a protected cipher implementation (AES-128) that deploys both higher-order masking against SCA and duplication-based fault detection against FI. We assume that the fault detection procedure is implemented in a protected manner and unmasking is performed solely on correct ciphertexts, i.e. the combined attack of Roche et al. [22] is no longer applicable. Furthermore, we consider an attacker that has no access or control over the plaintext and ciphertext, both are only assumed to be random. Thus, straightforward high-order DPA [5, 21] or SIFA are not applicable, since they both rely on plaintext or ciphertext knowledge. Likewise, the attacks of Korkikian et al. [18], Saha et al. [23, 24] are not applicable since the plaintext cannot be repeated. Similarly to Saha et al. [24] though, we assume that the attacker can observe the detection of a fault, either when a detection-based countermeasure is triggered within the AES implementation or through an error message in a higher-layer protocol.

Such a restricted attack scenario can impose severe limitations on the adversary, since it disables several effective tools from his arsenal. The main attack strategy that appears viable against such an implementation is a blind high-order SCA. A profiled version of this attack would follow the lines of Hanley, Tunstall and Marnane [16] or analytical/algebraic attacks [28], i.e. profiling and combining all shares of at least two intermediate values. An unprofiled version would follow the lines of Linge, Dumas and Lambert-Lacroix [20], or Clavier, Reynaud and Wurcker [9], using the joint distribution of multiple intermediates' leakages. All high-order attacks however are impacted by the noise amplification and the exponential relation between attack traces and the masking order [5]. Intuitively, a blind attack alters the usual data complexity growing exponentially in the masking order d, to grow exponentially in $2 \cdot d$. Such a conundrum motivates us towards a combined attack that is not subject to the noise amplification, while still dealing with the unknown plaintext and ciphertext. However, our attack relies on the fact that only correct ciphertexts are unmasked and their leakage can be observed, and additionally comes at the cost of both side-channel and fault profiling, which we describe in the next sections.

3.2 Working principle

Blind side-channel SIFA consists of two phases. The first phase (profiling) requires access to an open training device supplied with random plaintexts, and knowledge of the ciphertexts and keys. Such device is used to profile the side-channel leakage of the ciphertext (Section 3.2.1) and subsequently profile the biased fault distribution of the ciphertext (Section 3.2.2). The second phase (actual attack) is performed on a closed device where the ciphertext is unknown. The attacker injects faults while observing the side-channel leakage of the ciphertext resulting only from ineffective faults and random unknown plaintexts. Subsequently a key recovery strategy is deployed (Section 3.2.3).

3.2.1 Side-channel profiling During the first phase, the attacker profiles the ciphertext leakage using side-channel measurements and a deterministic leakage function such as HW[1]. A ciphertext byte C's leakage L is modeled as: $L = \alpha \cdot \mathsf{HW}(C) + \beta + N$, where α and β are constant terms, $N \sim \mathcal{N}(0, \sigma^2)$ is Gaussian noise and $C \in \mathbb{F}_{2^8}$. Subsequently, the attacker uses pairs of known ciphertext bytes and corresponding leakages to estimate the Gaussian templates $\mathcal{N}(\mu_h, \sigma^2), \forall h \in \{0, \ldots, 8\}$. During the actual attack, these estimations will allow us to compute the likelihood $\Pr[\mathsf{HW}(c) = h|l] \propto \mathcal{N}(l; (\mu_h, \sigma^2))$ and recover a probability vector on $\mathsf{HW}(c)$.

3.2.2 Ineffective fault profiling Continuing, the attacker must characterize the ineffective fault distribution of the ciphertext. To do so, he injects faults during the 10th round of AES in order to bias the Sbox output and observes the resulting correct ciphertexts. In this round we denote the true key byte as k, the biased SBox output as s' and the corresponding ciphertext byte by c', i.e. when no fault is detected, $c' = k \oplus s'$. For brevity we omit the $\mathsf{ShiftRows}$ operation. The faults can be injected before or during the masked SBox operation, since targeting non linear operations is a necessary requirement of SIFA on masked implementations [11]. The attacker acquires multiple ciphertexts $\{c'_i\}$ (filtered by fault detection) and uses the knowledge of k during the profiling phase to backtrack to $\{s'_i\}$. Consequently he computes $\mathsf{HW}(s'_i \oplus k^*), \forall i, \forall k^* \in \mathbb{F}_{2^8}$ and builds using histograms the fault templates of the ciphertext's HW distribution for all candidates k^*. We denote these fault templates by $\mathcal{D}_0, \mathcal{D}_1, \ldots, \mathcal{D}_{255}$, which provide the probability $\Pr[h|k^*] = \mathcal{D}_{k^*}(h)$ of observing a ciphertext with Hamming weight h if the key is k^*.

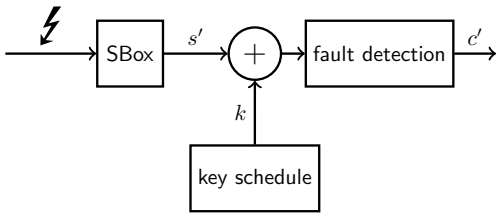

Fig. 1. Fault injection during the 10th round of AES.

3.2.3 Key-recovery strategies During the second phase of the attack, the adversary has *no access to or control* over the ciphertexts or plaintexts, thus he is limited to injecting faults and measuring the side-channel leakage of ineffectively faulty ciphertexts. In particular, he measures the leakages $\{l_i\}$ of n ciphertexts $\{c'_i\}$ with respective Hamming weights $\{h_i\}$, while faulting the 10th round SBox.

[1] HW leakage is commonly encountered in embedded devices and is widespread in SCA literature, since it provides a good approximation of side-channel behavior.

The attack's goal is to use the side-channel and fault profiling steps described in Sections 3.2.1 and 3.2.2 in order to recover the secret key. Following, we describe different key-recovery strategies.

Distribution Comparison. A straightforward key-recovery strategy consists of the following steps. First, the attacker simply classifies each leakage l_i into its mostly likely HW value i.e. he finds the Hamming weight h that maximizes the likelihood:

$$\Pr[h|l_i] \propto \Pr[l_i|h] \cdot \Pr[h] \tag{1}$$

where $\Pr[l_i|h] = \mathcal{N}(l_i; (\mu_h, \sigma^2))$ (estimated in 3.2.1) and $\Pr[h] = 2^{-8} \cdot \binom{8}{h}$ is the prior probability of weight h. Once all n ciphertext leakages have been classified, the attacker builds a histogram of the ciphertext's HW distribution that we denote by $\tilde{\mathcal{D}}_k$. Finally, a key candidate k^* is scored based on how similar its fault-based hypothetical distribution $\mathcal{D}_{k^*}$ (estimated in 3.2.2) is to $\tilde{\mathcal{D}}_k$. Different distinguishers can be used to compare distributions, such as SEI [14, 12], χ^2 and KL-divergence [19].

Improved Distribution Comparison. An issue that arises when comparing the hypothetical distributions $\mathcal{D}_{k^*}$ to the empirical distribution $\tilde{\mathcal{D}}_k$, is that while the hypothetical distributions correspond to true values, $\tilde{\mathcal{D}}_k$ is estimated from side-channel leakages and is impacted by the side-channel noise. In the profiled case we investigate, it is possible to improve the previous comparison by accurately simulating the impact of the noise on the hypothetical distributions, altering slightly the process in 3.2.2. First, we add to the ciphertext's Hamming weights (used to estimate the hypothetical distributions $\mathcal{D}_{k^*}$) noise samples, knowing the noise standard deviation σ, to simulate their side-channel leakage. Then each simulated leakage is classified into its most likely Hamming weight value based on the profiled leakage model of Section 3.2.1. This makes distributions $\mathcal{D}_{k^*}$ more comparable to the empirical distribution $\tilde{\mathcal{D}}_k$ that is estimated from the side-channel leakage. Another approach to perform this, is to estimate a confusion matrix on the HW classification and multiply it by the hypothetical ineffective fault distributions.

Maximum Likelihood. A more generic approach to this problem is to follow a maximum likelihood strategy that takes into account the full distribution of the leakage and not only the classified Hamming weights. In our case, we are interested in the likelihood of a key candidate k^*, having observed n leakages of the biased ciphertexts that we denote by $\{l_i\}_{1 \le i \le n}$. The likelihood $\Pr[k^*|\{l_i\}_{1 \le i < n}]$ is evaluated as follows:

$$\Pr[k^*|\{l_i\}_{1 \le i \le n}] = \prod_{i=1}^{n} \Pr[k^*|l_i] = \prod_{i=1}^{n} \frac{\Pr[l_i|k^*] \cdot \Pr[k^*]}{\Pr[l_i]} \tag{2}$$

$$\propto \prod_{i=1}^{n} \Pr[l_i|k^*] \propto \prod_{i=1}^{n} \sum_{h=0}^{8} \Pr[l_i|h] \cdot \Pr[h|k^*] \tag{3}$$

Equation 2 assumes the independence of the ciphertexts' leakages and then applies Bayes' rule. Equation 3 is simply deduced by ignoring the terms that do not help to distinguish the key candidates, in this case $\Pr[l_i]$, and terms constant for each key, in this case $\Pr[k^*] = \frac{1}{256}$. Then, Equation 3 applies the law of total probabilities. In the final form, we recognize the term $\Pr[l_i|h]$ which simply corresponds to $\mathcal{N}(l_i; (\mu_h, \sigma^2))$ and was estimated in 3.2.1. Similarly, we recognize the term $\Pr[h|k^*]$ which corresponds to the fault templates in 3.2.2, i.e. $\Pr[h|k^*] = \mathcal{D}_{k^*}(h)$. This maximum likelihood derivation is reminiscent of the one used by Clavier and Reynaud [8] for blind SCA using joint distributions of leakages.

4 Theoretical and Simulated Analysis

This section analyses the effectiveness of the proposed attack using simulations. Section 4.1 discusses the necessary conditions on the fault distribution and investigates the impact of the side-channel leakage function. Section 4.2 compares the previously described key-recovery strategies.

4.1 Impact of the leakage function

The conditions on SIFA apply to the proposed attack as well, which are a non-negligible ineffective fault probability and a non-uniform distribution of ineffective faults. For instance, the uniform bit-flip fault model does not fulfill such a condition [12]. Accordingly, if we consider that the adversary has access only to the ciphertexts' Hamming weights, then a uniformly distributed fault implies that Hamming weights are distributed according to $\mathcal{B}(8, \frac{1}{2})$ (with $\mathcal{B}$ denoting the binomial distribution) for all key candidates, and thus the attacker is unable to distinguish the key byte.

In the following, we show that despite obtaining a fault verifying the previous conditions, the success of blind side-channel SIFA does not only depend on the fault distribution but also on the deterministic part of the side-channel leakage. For the first part of this analysis, we leave aside the impact of the noise (which we investigate later) and mainly focus on the deterministic leakage function due to its inherent information compression: when a value is explicitly targeted by a side-channel adversary, only partial information can be recovered on it. This information can be represented by a surjective function (typically HW). A simple example that explicitly illustrates this point is a stuck-at-0 fault, combined with a device leaking the HW of manipulated values. This implies that the biased ciphertexts are equal to the key byte, since $c' = s' \oplus k = 0 \oplus k = k$. However, only the HW of the ciphertext (and thus of the key byte) can be observed and recovered through SCA[2].

While the stuck-at-0 case is simple to understand, it may be less trivial to notice the shortcomings of other induced biases in combination with the leakage

[2] The average number of keys to test after recovering the HW of its 16 bytes is $\approx 2^{90}$.

function. For instance, despite inducing an appropriate bias for SIFA, a random-and fault (bit-wise multiplication by random values), combined with the HW function is not able to recover the full value of the key byte as demonstrated by the simulated experiments shown on Figure 2. First, on the left side of Figure 2, we plot the hypothetical distributions $\mathcal{D}_{k^*}$ of ciphertexts' Hamming weights biased by ineffective random-and faults, for all possible key byte values $k^* \in \mathbb{F}_{2^8}$. We can observe that some of these distributions are identical and thus cannot be distinguished. On the right side of Figure 2 this observation is further confirmed. We performed the maximum likelihood attack described in Section 3.2.3 on HW leakages (HW, blue line). We compare it to the same attack performed on identity leakages (ID, orange line). We plot for both attacks the average rank of the correct key byte as function of the number of fault inductions. Note that the ineffectivity rate (the ratio between the number of ineffective faults and the total number of fault injections) of a random-and fault is approx. 10%. Naturally, since random-and faults are suitable for SIFA, it succeeds using less than 50 ineffective faults. However, an identical attack performed on HW is unable to achieve an average rank below 20 candidates, regardless of the number of fault injections. This is due to the previously mentioned information compression property of the side-channel leakage function.

The previous observations and experiments suggest that the fault induction (together with the fault detection mechanism) is required to induce a suitable bias when combined with the leakage function. Specifically, the attack's success depends on all hypothetical distributions (as shown on the left of Figure 2) and the possibility to distinguish every distribution out of the 256 other ones. Notably, this condition is different from the one required for SIFA on the penultimate round which mainly depends on the advantage of distinguishing the biased distribution from a uniform one. The suitability of an induced ineffective fault distribution on the target device can be easily confirmed by performing SIFA taking into account the leakage function, to evaluate the maximum information that can be recovered on the key byte, as shown for instance on the right side of Figure 2.

4.2 Comparison of key-recovery strategies

In this section we compare the key-recovery strategies described in Section 3.2.3: the distribution comparison, the improved distribution comparison and the maximum likelihood strategy. In the following, to compare distributions we use the Kullback-Leibler divergence D_{KL} as a distinguisher. The $D_{KL}(\mathcal{Q}_1||\mathcal{Q}_2)$ from distribution $\mathcal{Q}_1$ to distribution $\mathcal{Q}_2$ is defined as:

$$D_{KL}(\mathcal{Q}_1||\mathcal{Q}_2) = \sum_x \mathcal{Q}_1(x) \log \frac{\mathcal{Q}_1(x)}{\mathcal{Q}_2(x)}$$

We also use a simulated ineffective fault distribution and simulated HW leakages with noise standard deviation $\sigma \in \{0.7, 1.5\}$. The ineffective fault distribution is generated as a random highly non-uniform distribution on $\mathbb{F}_{2^8}$ as described by the Python code:

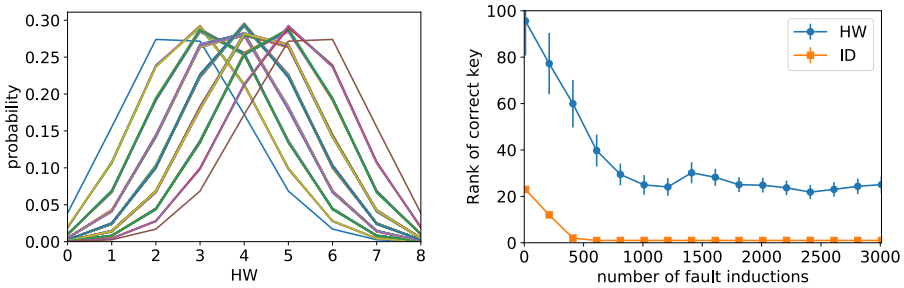

Fig. 2. Blind side-channel SIFA with random-and faults and noiseless HW leakage. Left: hypothetical distributions $\mathcal{D}_{k^*}$ for $k^* \in \mathbb{F}_{2^8}$. Right: comparison of classic SIFA and SIFA on HW for random-and faults.

```
fd = numpy.random.uniform(0.0,0.4,size=256)
fd = fd/numpy.sum(fd)
```

The distributions generated by this process usually allow to extract most of the information on the key byte. We performed all three attacks for both noise levels and plot on Figure 3 the average ranks they achieved as function of the number of ineffective faults. The left side corresponds to the low-noise case with $\sigma = 0.7$ and the right side to moderate-noise level with $\sigma = 1.5$. The X-axis corresponds to the number of ineffectively faulty ciphertexts and the Y-axis to the rank of the key byte. The vertical lines on every data point correspond to the standard error on the mean estimation of the rank.

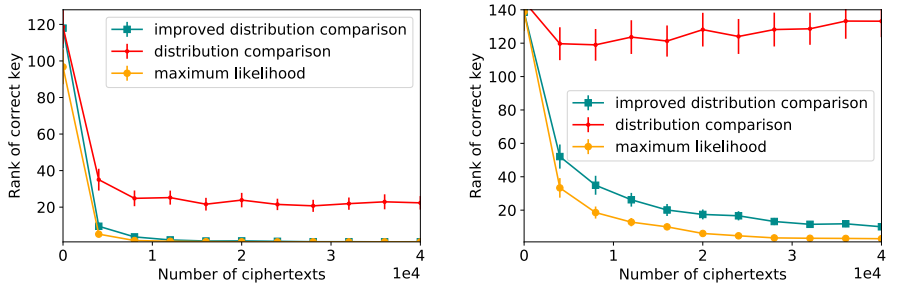

Fig. 3. Comparison of key-recovery strategies for blind side-channel SIFA. Left: $\sigma = 0.7$. Right: $\sigma = 1.5$.

On Figure 3, we observe that for low noise, the maximum likelihood approach and the improved distribution comparison achieve similar results, while the classical comparison is not able to decrease the ranks below 20 candidates. This is due to the fact that the empirical distribution $\tilde{\mathcal{D}}_k$ is distorted as a result of the

side-channel noise, leading to a poor comparison with the hypothetical distributions $\mathcal{D}_{k^*}$. This observation is confirmed in the moderate noise case where the classical distribution comparison strategy is not able to distinguish the correct key, while both the improved comparison and the maximum likelihood strategies are able to sufficiently lower the key rank. However, we observe that the maximum likelihood attack is more robust against noise and is the optimal key-recovery strategy. Based on these results, the following sections will focus on the maximum likelihood approach.

5 Experimental Validation

This section confirms the applicability of blind side-channel SIFA in an experimental setting, using a modern ARM micro-controller. We use a setup that injects faults while concurrently measuring the side-channel leakage. We use the same device for the profiling and the attack.

5.1 Target and setup description

The target device is an ARM Cortex-M4 based micro-controller running at a clock frequency of 32MHz. The target implementation is the ANSSI AES-128 hardened library [2] protected with affine masking [15] against SCA. The shuffling countermeasure is disabled. While this implementation is not protected against faults, we simply assume that a redundant computation (duplication) is implemented in a masked manner as a fault countermeasure and incorrect ciphertexts are neither unmasked nor returned by the device. Instead, the unmasking of the ciphertext is followed by a state copy to return an array of bytes to the encryption function. To perform FI, a diode laser system is used to generate laser pulses with a diode current of 200mA and a pulse width of 37ns. The laser beam is focused on the die surface with a lens. During a χ^2-square based preliminary parameters search, we found that a fault injection on the SRAM2 produces a significant bias on the output of the Sbox operation. For our experiments, we chose a fixed position within the SRAM2 block. A trigger and a delay generator were used to trigger the FI. Regarding side-channel measurements, a Langer microprobe was placed near the SRAM and the glue logic. The leakage traces of the target ciphertext byte were recorded with a Lecroy WR 625Zi oscilloscope at a sampling rate of 2.5Ghz.

5.2 Hybrid Attack: Real faults and simulated side-channel

Given the issues laid out in Section 4, we first aim to confirm that the ineffective fault distribution induced by the laser can indeed result in key recovery under a HW leakage function. Thus, we performed the maximum likelihood attack as described in Section 3.2.3, using the ineffective fault distribution produced by the FI and we simulated the HW side-channel leakage. This hybrid attack is carried out for different noise levels. It corresponds to ideal side-channel modeling, since

we assume the leakage to conform strictly to the HW model. In addition, we use the same distribution to both estimate the hypothetical fault distributions and to sample the ineffective faults for the attack. The results are plotted on Figure 4 where the X-axis corresponds to the number of ineffective faults and the Y-axis to the average rank of the key bytes. The results shown on Figure 4 demonstrate that the specific laser-induced faults, combined with noisy HW leakage of the ciphertexts can recover the correct value of the key byte in the context of SIFA-like attacks, without any input or output knowledge. It is also robust to the addition of side-channel noise and to the noise induced by failed (non-ineffective) fault injections during the experimental process.

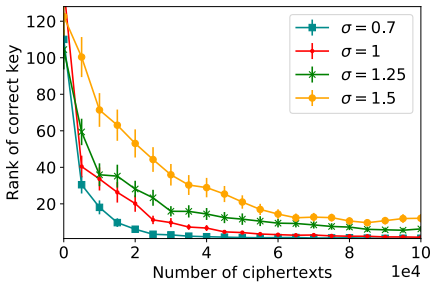

Fig. 4. Blind side-channel SIFA on ARM Cortex M4. The attack uses the estimated fault distributions after laser FI and simulated side-channel leakages.

5.3 Combined Attack: Real faults and real side-channel

Having established the stability of the attack we proceed with the results of the combined FI and SCA. Initially (profiling phase), the ineffective fault profiling was performed using random plaintexts and a random key as described in Section 3.2.2. Continuing (attack phase), the combined FI and side-channel measurements were performed using random plaintexts and a fixed key. We note that our ineffective fault profiles did not exhibit any significant differences compared to the ones observed in the attack phase. We also emphasize that the issue of either fault or side-channel portability on different devices is orthogonal to this paper. How this matter affects a combined attack such as performed in this work, however remains an interesting scope for further research.

Regarding the side-channel measurements, we obtained 150k traces corresponding to the ineffective faults after 550k fault injections (leading to a 30% ineffectivity rate). We processed the traces by selecting (during profiling) the time samples, also called Points-of-Interest (PoI), with the highest correlation to the HW of the ciphertext, then applied PCA [1] to compress the leakage further into a single dimension leading to an SNR ≈ 0.4. Due to the length of the experiment, we selected 40k traces from the middle of the experiment for profiling

and used the remaining traces to conduct the maximum likelihood attack. Evaluation results are plotted on Figure 5 as function of the number of ineffective faults.

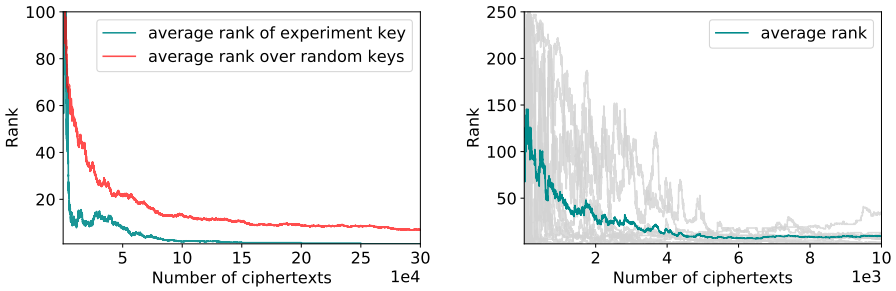

Fig. 5. Results of combined attack. (Right) Full combined experiment. (Left) Simulated side-channel with measurements' noise level.

First, on the right part of the figure we provide results of the full combined attack on a single key. The lowest average rank of 6.54 is achieved with less than 8k traces. To replicate the attack for a large number of keys, we use simulated side-channel attack with a noise level corresponding to the SNR observed on the device ($\sigma^2 = 5$). We show the results on the left part of Figure 5, where we plot the average rank across random keys and additionally the average rank for the key used in the combined experiment. We notice that while the simulated attack eventually leads to an average rank of 1, the attack on real traces reaches rapidly a very low rank of 6.54 but is left with only a few candidates to distinguish. Presumably, this effect can be due to the experimental setup and the interaction between the side-channel leakage and the laser fault injection. This however was not verified experimentally. The thorough investigation of these effects and how to correct them is a very interesting scope for advanced research in the context of combined attacks.

6 Conclusion

This work considered profiled and combined FI and SCA in order to apply a SIFA-like key-recovery without any knowledge of the ciphertext. Concretely, we show that fault inductions on an ARM Cortex-M4 micro-controller lead to very low key ranks, despite only observing the side-channel leakage of ineffectively faulty ciphertexts, and without any plaintext repetition. However, our attack relies on both side-channel and fault profiling and additionally assumes that the ciphertext is an ephemeral secret that is unmasked after the fault detection mechanism.

Future work could potentially investigate the possibility of applying blind side-channel based SIFA in a fully unprofiled fashion, i.e. without any prior fault

characterization or side-channel profiling, and also how this attack applies when infective countermeasures are deployed. Another direction for further research involves comparing the attack proposed in this work to blind high-order SCA, when the attacker has no control over or knowledge of the plaintext/ciphertext. Last but not least, we have observed that performing an experiment with a combined FI and SCA setup is not always simple. Future research can work towards dealing with experimental instabilities with setup improvements or using signal processing.

Acknowledgment

The authors would like to thank Martin Butkus and Vincent Verneuil for the very valuable discussions and contributions.

References

1. Cédric Archambeau, Eric Peeters, François-Xavier Standaert, and Jean-Jacques Quisquater. Template attacks in principal subspaces. In Louis Goubin and Mitsuru Matsui, editors, *Cryptographic Hardware and Embedded Systems - CHES 2006*. Springer, 2006.
2. Ryad Benadjila, Louiza Khati, Emmanuel Prouff, and Adrian Thillard. Hardened library for aes-128 encryption/decryption on arm cortex m4 architecture. https://github.com/ANSSI-FR/SecAESSTM32.
3. Eli Biham and Adi Shamir. Differential fault analysis of secret key cryptosystems. In Burton S. Kaliski Jr., editor, *Advances in Cryptology - CRYPTO '97*. Springer, 1997.
4. Dan Boneh, Richard A. DeMillo, and Richard J. Lipton. On the importance of eliminating errors in cryptographic computations. *J. Cryptology*, 14(2), 2001.
5. Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards sound approaches to counteract power-analysis attacks. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO '99*. Springer, 1999.
6. Christophe Clavier. Secret external encodings do not prevent transient fault analysis. In Pascal Paillier and Ingrid Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings*, volume 4727 of *Lecture Notes in Computer Science*, pages 181–194. Springer, 2007.
7. Christophe Clavier. Secret external encodings do not prevent transient fault analysis. In Pascal Paillier and Ingrid Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems - CHES 2007*. Springer, 2007.
8. Christophe Clavier and Léo Reynaud. Improved blind side-channel analysis by exploitation of joint distributions of leakages. In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems - CHES 2017*. Springer, 2017.
9. Christophe Clavier, Léo Reynaud, and Antoine Wurcker. Quadrivariate improved blind side-channel analysis on boolean masked AES. In Junfeng Fan and Benedikt Gierlichs, editors, *Constructive Side-Channel Analysis and Secure Design - COSADE 2018*. Springer, 2018.

10. EMV Co. Emv integrated circuit card specifications for payment systems, security and key management, november 2011. version 4.3.
11. Christoph Dobraunig, Maria Eichlseder, Hannes Groß, Stefan Mangard, Florian Mendel, and Robert Primas. Statistical ineffective fault attacks on masked AES with fault countermeasures. In Thomas Peyrin and Steven D. Galbraith, editors, *Advances in Cryptology - ASIACRYPT 2018*. Springer, 2018.
12. Christoph Dobraunig, Maria Eichlseder, Thomas Korak, Stefan Mangard, Florian Mendel, and Robert Primas. SIFA: exploiting ineffective fault inductions on symmetric cryptography. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018.
13. Thomas Fuhr, Éliane Jaulmes, Victor Lomné, and Adrian Thillard. Fault attacks on AES with faulty ciphertexts only. In Wieland Fischer and Jörn-Marc Schmidt, editors, *2013 Workshop on Fault Diagnosis and Tolerance in Cryptography, Los Alamitos, CA, USA, August 20, 2013*, pages 108–118. IEEE Computer Society, 2013.
14. Thomas Fuhr, Éliane Jaulmes, Victor Lomné, and Adrian Thillard. Fault attacks on AES with faulty ciphertexts only. In Wieland Fischer and Jörn-Marc Schmidt, editors, *2013 Workshop on Fault Diagnosis and Tolerance in Cryptography*. IEEE Computer Society, 2013.
15. Guillaume Fumaroli, Ange Martinelli, Emmanuel Prouff, and Matthieu Rivain. Affine masking against higher-order side channel analysis. In Alex Biryukov, Guang Gong, and Douglas R. Stinson, editors, *Selected Areas in Cryptography - SAC 2010*. Springer, 2010.
16. Neil Hanley, Michael Tunstall, and William P. Marnane. Unknown plaintext template attacks. In Heung Youl Youm and Moti Yung, editors, *Information Security Applications, WISA 2009*. Springer, 2009.
17. Oliver Kömmerling and Markus G. Kuhn. Design principles for tamper-resistant smartcard processors. In Scott B. Guthery and Peter Honeyman, editors, *Proceedings of the 1st Workshop on Smartcard Technology, Smartcard 1999, Chicago, Illinois, USA, May 10-11, 1999*. USENIX Association, 1999.
18. Roman Korkikian, Sylvain Pelissier, and David Naccache. Blind fault attack against SPN ciphers. In Assia Tria and Dooho Choi, editors, *2014 Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC 2014*. IEEE Computer Society, 2014.
19. Yanis Linge, Cécile Dumas, and Sophie Lambert-Lacroix. Using the joint distributions of a cryptographic function in side channel analysis. In Emmanuel Prouff, editor, *Constructive Side-Channel Analysis and Secure Design - COSADE 2014*. Springer, 2014.
20. Yanis Linge, Cécile Dumas, and Sophie Lambert-Lacroix. Using the joint distributions of a cryptographic function in side channel analysis. In Emmanuel Prouff, editor, *Constructive Side-Channel Analysis and Secure Design - 5th International Workshop, COSADE 2014, Paris, France, April 13-15, 2014. Revised Selected Papers*, volume 8622 of *Lecture Notes in Computer Science*, pages 199–213. Springer, 2014.
21. Thomas S. Messerges. Using second-order power analysis to attack DPA resistant software. In Çetin Kaya Koç and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2000, Second International Workshop, Worcester, MA, USA, August 17-18, 2000, Proceedings*, volume 1965 of *Lecture Notes in Computer Science*, pages 238–251. Springer, 2000.
22. Thomas Roche, Victor Lomné, and Karim Khalfallah. Combined fault and side-channel attack on protected implementations of AES. In Emmanuel Prouff, editor,

Smart Card Research and Advanced Applications - 10th IFIP WG 8.8/11.2 International Conference, CARDIS 2011, Leuven, Belgium, September 14-16, 2011, Revised Selected Papers, volume 7079 of *Lecture Notes in Computer Science*, pages 65–83. Springer, 2011.

23. Sayandeep Saha, Dirmanto Jap, Jakub Breier, Shivam Bhasin, Debdeep Mukhopadhyay, and Pallab Dasgupta. Breaking redundancy-based countermeasures with random faults and power side channel. In *2018 Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC 2018, Amsterdam, The Netherlands, September 13, 2018*, pages 15–22. IEEE Computer Society, 2018.
24. Sayandeep Saha, Debapriya Basu Roy, Arnab Bag, Sikhar Patranabis, and Debdeep Mukhopadhyay. Breach the gate: Exploiting observability for fault template attacks on block ciphers. *IACR Cryptology ePrint Archive*, 2019, 2019.
25. Sergei Skorobogatov. Fault attacks on secure chips.
26. Harshal Tupsamudre, Shikha Bisht, and Debdeep Mukhopadhyay. Destroying fault invariant with randomization - A countermeasure for AES against differential fault attacks. In Lejla Batina and Matthew Robshaw, editors, *Cryptographic Hardware and Embedded Systems - CHES 2014*. Springer, 2014.
27. Jasper G. J. van Woudenberg, Marc F. Witteman, and Federico Menarini. Practical optical fault injection on secure microcontrollers. In Luca Breveglieri, Sylvain Guilley, Israel Koren, David Naccache, and Junko Takahashi, editors, *2011 Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC 2011*. IEEE Computer Society, 2011.
28. Nicolas Veyrat-Charvillon, Benoît Gérard, and François-Xavier Standaert. Soft analytical side-channel attacks. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I*, volume 8873 of *Lecture Notes in Computer Science*, pages 282–296. Springer, 2014.