Side-channel Analysis of Lightweight Ciphers: Does Lightweight Equal Easy? [⋆]

Annelie Heuser[1], Stjepan Picek[2], Sylvain Guilley[3], and Nele Mentens[2]

[1] IRISA/CNRS, Rennes, France
[2] KU Leuven, ESAT/COSIC and iMinds, Leuven-Heverlee, Belgium
[3] TELECOM-ParisTech, Paris, France and Secure-IC S.A.S., Rennes, France

Abstract. Side-channel attacks represent a powerful category of attacks against cryptographic devices. Still, side-channel analysis for lightweight ciphers is much less investigated than for instance for AES. Although intuition may lead to the conclusion that lightweight ciphers are weaker in terms of side-channel resistance, that remains to be confirmed and quantified. In this paper, we consider various side-channel analysis metrics which should provide an insight on the resistance of lightweight ciphers against side-channel attacks. In particular, for the non-profiled scenario we use the theoretical confusion coefficient and empirical correlation power analysis. Furthermore, we conduct a profiled side-channel analysis using various machine learning attacks on PRESENT and AES. Our results show that the difference between AES and lightweight ciphers is smaller than one would expect. Interestingly, we observe that the studied 4-bit S-boxes have a different side-channel resilience, while the difference in the 8-bit ones is only theoretically present.

Keywords: Lightweight cryptography, Machine learning, Comparison, Confusion coefficient, CPA

1 Introduction

With the advent of the Internet of Things, we are surrounded with smart objects (aka things) that have the ability to communicate with each other and with centralized resources. The two most common and widely noticed artifacts are RFID and Wireless Sensor Networks which are used in supply-chain management, logistics, home automation, surveillance, traffic control, medical monitoring, and many more. Most of these applications have the need for cryptographic secure components which inspired research on cryptographic algorithms for constrained devices. Accordingly, lightweight cryptography has been an active research area over the last 10 years. A number of innovative ciphers have been proposed in order to optimize various performance criteria and have been subject to many

[⋆] This work has been supported in part by Croatian Science Foundation under the project IP-2014-09-4882. In addition, this work was supported in part by the Research Council KU Leuven (C16/15/058) and IOF project EDA-DSE (HB/13/020).

comparisons. Lately, the resistance against side-channel attacks has been considered as an additional decision factor.

Side-channel attacks analyze physical leakage that is unintentionally emitted during cryptographic operations in a device (e.g., power consumption [1], electromagnetic emanation [2]). This side-channel leakage is statistically dependent on intermediate processed values involving the secret key, which makes it possible to retrieve the secret from the measured data. So-called profiled side-channel distinguishers assume that the attacker is able to possess an additional device to the one he wants to attack, and on which he has the freedom of nearly full control. In this advanced setting, Machine learning (ML) techniques have shown to be effective in various scenarios (e.g., [3, 4]).

Side-channel analysis for lightweight ciphers is of particular interest not only because of the apparent lack of research so far, but also because of the interesting properties of S-boxes. Since the nonlinearity property for S-boxes usually used in lightweight ciphers (i.e., 4×4) can be maximally equal to 4, the difference between the input and the output of an S-box is much smaller than for instance for AES [5]. Therefore, one could conclude that from that aspect, SCA for lightweight ciphers must be more difficult. However, the number of possible classes (e.g., Hamming weight (HW) or key classes) is significantly lower, which may indicate that (profiled) SCA must be easier than for standard ciphers. Besides the difference in the number of classes and consequently probabilities of correct classification, there is also a huge time and space complexity advantage (for the attacker) when dealing with lightweight ciphers.

Our Contributions In this paper we give a detailed study of lightweight ciphers in terms of side-channel resistance, in particular for software implementations. As a point of exploitation we concentrate on the non-linear operation (S-box) during the first round. Our comparison includes SPN ciphers with 4-bit S-boxes such as KLEIN [6], PRESENT [7], PRIDE [8], RECTANGLE [9], Mysterion [10] as well as ciphers with 8-bit S-boxes: AES, Zorro [11], Robin [12].

In the non-profiled scenario we investigate first the relationship between different key hypotheses with the confusion coefficient [13, 14]. Using specific properties of the confusion coefficient (like the minimum value and the variance) we give a preliminary classification regarding the side-channel resistance. Furthermore, using simulated data for various signal-to-noise ratios (SNR) we present empirical results for Correlation Power Analysis (CPA) [15] and discuss the difference between attacking 4-bit and 8-bit S-boxes. Finally, we compare several supervised (i.e., profiled) machine learning techniques for PRESENT and AES.

Road Map This paper is organized as follows. Section 2 gives basic information on the ciphers and exploitations we investigate. Next, in Section 3 we discuss Correlation Power Analysis (CPA), confusion coefficient, and profiled side-channel analysis. Section 4 concludes and offers directions for future work.

2 Ciphers & Exploitations

2.1 Investigated Ciphers

AES [5] The Advanced Encryption Standard (AES) has been standardized by NIST in 2001 [16]. It has an SPN structure with an internal fixed block size of 128-bits represented as a 4×4 byte matrix. At the beginning, the plaintext state is `xor-ed` with the secret key. Subsequently, each encryption round consists of the application of `SubBytes`, `ShiftRows`, `MixColumns`, and `AddRoundKey`; in the last round, `MixColumns` is omitted.

KLEIN [6] KLEIN is an AES-like lightweight block cipher. The substitution stage uses 16 similar involutive 4-bit S-boxes. Similar to AES, each encryption round consists of `AddRoundKey`, `SubNibbles`, `RotNibbles`, and `MixNibbles`, followed by a final key addition.

PRESENT [7] PRESENT has a 64-bit block size with a bit oriented permutation layer. The non-linear layer is based on a single 4-bit S-box which was designed to be optimal in hardware. An encryption round consists of `AddRoundKey`, a substitution (`sBoxLayer`), and a permutation layer (`pLayer`). A final key addition is performed after the encryption rounds.

PRIDE [8] PRIDE has been optimized for 8-bit microcontrollers with a special focus on the linear layer of the cipher. It is designed in a bit-sliced fashion to minimize the number of instructions necessary to evaluate it. The 4-bit S-box is an involution.

RECTANGLE [9] The state of RECTANGLE is represented as a 4×16 matrix. The non-linear layer consists of the parallel application of a 4-bit S-box on the columns of the state and the linear layer is a fixed rotation over a different amount of steps in each row.

Robin [12] Robin is one instance of the so-called LS-design, in which the internal state of the cipher is a matrix of $s \times L$ bits. The non-linear layer consists of the parallel applications of a $s \times s$ bits ($s = 8$) permutation on each column, which is chosen to be efficiently implemented in a bit-sliced fashion. The linear layer consists of the application of a linear $L \times L$ bit ($L = 16$) permutation on each row of the matrix.

Mysterion [10] The Mysterion cipher is also based on the LS-design principles. The internal state of the block cipher is organized into a 4×32 bit matrix for Mysterion-128, which is further subdivided into 4 4×8 blocks. A round contains the following operations: `S-box layer`, `L-Box layer` and `ShiftColumns`. The `S-box layer` is a 4-bit S-box called “Class 13”, as introduced in [17], that is applied in parallel to each column of the internal state.

Zorro [11] Zorro is a modified version of AES with a variant of the S-box that is easier to mask. Fewer S-box calls are performed and the number of multiplications has been minimized. Besides, the execution is split into “steps” of 4 rounds and the key (simply the master key) is added only at the end of each step.

2.2 Exploitations

In this paper, our main targets are the weaknesses arising in software implementations on serial microprocessors. In these applications, the Hamming weight (HW) and the Hamming distance (HD) leakage model are most commonly found in practice. More precisely, the loading and storing of data in memory (e.g., `S-box` calls) is usually causing HW leakage, whereas the register updating (e.g., writing of intermediate round states) is causing HD leakage. Typically the latter is less significant than the former, which is why we concentrate on a specific memory operation.

We focus on side-channel attacks targeting the key processed within the first round using a divide and conquer strategy. The main common operation all previous described ciphers share, is first the addition (`xor`) of the roundkey/masterkey followed by (at least one) S-box call. Our study therefore concentrates on leakage measurements X arising from an S-box lookup operation within the first round, i.e.,

$$X = \mathrm{HW}(\mathtt{Sbox}[T \oplus k^*]) + N, \tag{1}$$

where N is independent additive noise, k^* one chunk of the secret key (round key or master key), and T a plaintext chunk (byte or nibble).

Note that our study does not include leakages from all operations in the specific ciphers, nor (in case the cipher uses a key scheduling algorithm) the complexity to go from a round key to the master key, which may be an interesting next step for future work.

3 (Empirical) Side-Channel Evaluation

3.1 CPA & Confusion Coefficient

Correlation Power Analysis (CPA) [15] is one of the most common non-profiled side-channel distinguishers that is also integrated in common criteria evaluations. For CPA in order to reveal the secret key k^*, the attacker makes hypothetical predictions depending on a key guess k on the deterministic part of the leakage. More precisely, let n denote the number of bits of one key chunk (n bits), for each key hypothesis $k \in \mathbb{F}_2^n$ one has:

$$Y(k) = \mathrm{HW}(\mathtt{Sbox}[T \oplus k]). \tag{2}$$

Given a set of Q leakage measurements $X_1, \ldots, X_Q$ corresponding to $T_1, \ldots, T_Q$ plaintexts, the attacker computes the correlation between the measurements and the hypothetical model $Y(k)$ for all key hypotheses. Finally, the key $\hat{k}$ that maximizes the correlation is selected, i.e.:

$$\hat{k} = \arg\max_{k} \rho(X, Y(k)) \tag{3}$$

with ρ being the Pearson correlation coefficient [18].

Before presenting the results from the empirical evaluation of CPA, we first want to further analyze the predictions in Eq. (2) for different ciphers. Interestingly, the predictions for different keys, $Y(0), \ldots, Y(2^n - 1)$, are not independent. Considering the model in Eq. (2), the relationships depend on the choice of the S-box and can be described by the so-called confusion coefficient [13, 14]

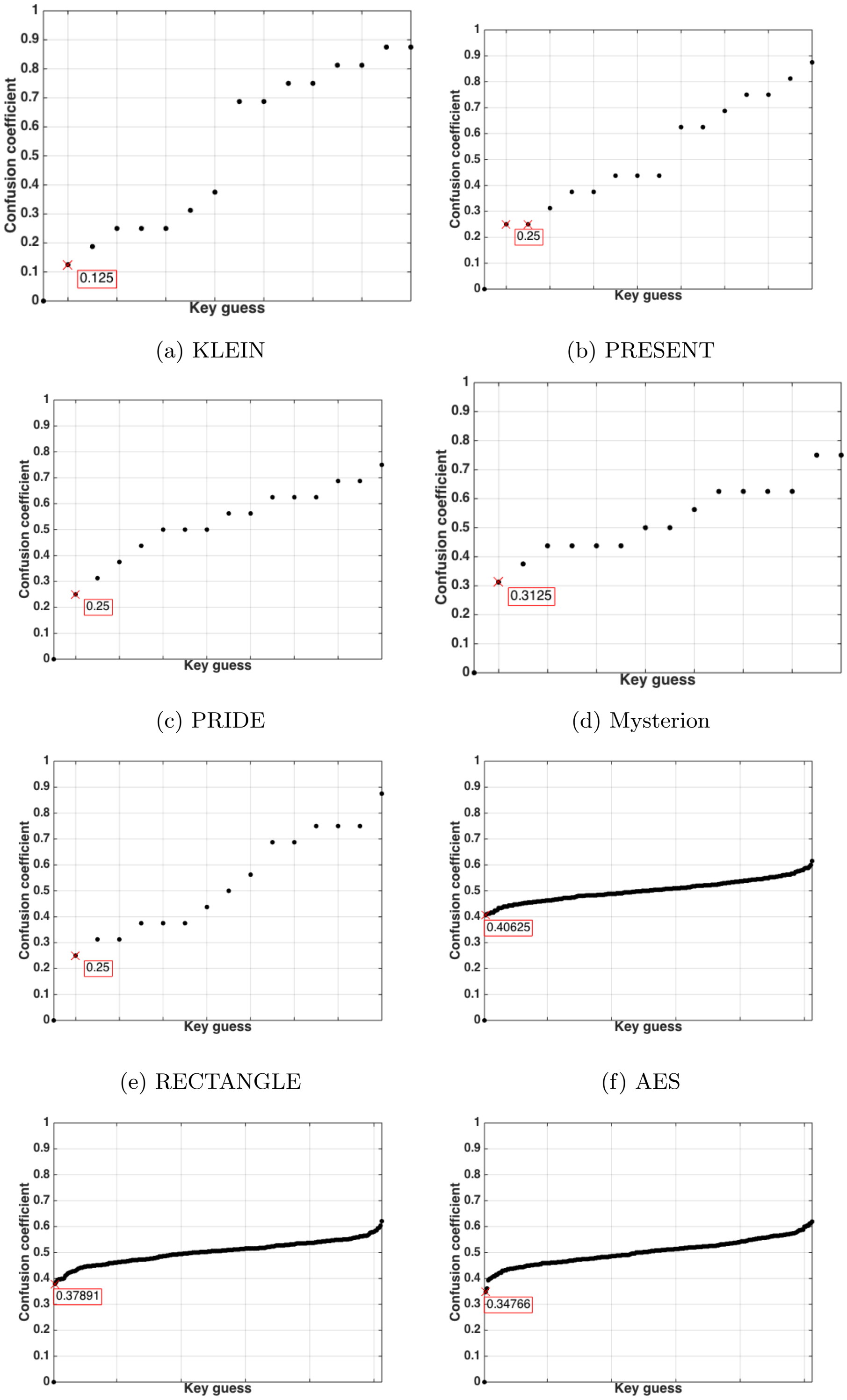
$$\kappa(k^*, k) = \mathbb{E}\Big\{\Big(\frac{Y(k^*) - Y(k)}{2}\Big)^2\Big\}, \tag{4}$$

where the expectation is taken over T.

Figures 1a to 1e show the confusion coefficient for 4-bit S-boxes and Figures 1f to 1h for 8-bit S-boxes. Note that, the distribution of $\kappa(k^*, k)$ is independent on the particular choice of k^* (in the case there are no weak keys) and the values are only permuted. For our experiments we choose $k^* = 0$ and furthermore order $\kappa(k^*, k)$ in an increasing order of magnitude. One can observe that the distribution is indeed different for the investigated ciphers. But how to judge what is easier and harder to attack from a side-channel point of view?

Recent works [13, 14] showed that the theoretical success rate of CPA can be divided into three factors: confusion coefficient, signal to noise ratio (SNR), and the number of measurements, but without further investigating the confusion coefficient in particular. The authors in [19] give a first-order approximation of the success rate of CPA (for a low SNR) which only depends on the minimum value of $\kappa(k^*, k)$, where the higher the minimum, the lower the side-channel security. Another approach has been taken in [20] using $var(\kappa(k^*, k))$ as a criterion, where smaller values indicate lower side-channel security. All values for 4-bit S-boxes are given in Table 1 on the left, where both criteria show the same trend, in particular, Mysterion should be the easiest to attack and KLEIN the most difficult. Note that PRESENT, PRIDE, and RECTANGLE have the same minimum value but different variances. Interestingly, the values given for 8-bit S-boxes in Table 1 on the right indicate that the side-channel resistance of the investigated 8-bit S-boxes is lower than for the ones with 4-bit S-boxes. Recall that the confusion coefficient measures the relationship between different key hypotheses. Now, as for 8-bits we have 256 possible values for $T \in \mathbb{F}_2^8$ and $Y(k) \in [0, 1, \ldots, 8]$ it is easier to distinguish than for 4-bit S-boxes with $T \in \mathbb{F}_2^4$ and $Y(k) \in [0, 1, \ldots, 4]$.

However, in practice we cannot straightforwardly conclude that due to the properties of the confusion coefficient, 4-bit S-boxes are harder to attack than 8-bit S-boxes. One reason is that the confusion coefficient is theoretical (i.e., holding for $Q \to \infty$). But, especially for low noise scenarios Q might be small (below 100). So, naturally the 4-bit variant with only 16 inputs should converge faster than with 256 inputs. Or in other words, considering $Q = 100$, one can



(a) KLEIN

(b) PRESENT

(c) PRIDE

(d) Mysterion

(e) RECTANGLE

(f) AES

(g) Zorro

(h) Robin

Fig. 1: Confusion coefficients

Table 1: Properties of $\kappa(k^*, k)$

	4-bit					8-bit		
	KLEIN	PRESENT	PRIDE	Mysterion	RECT.	AES	Zorro	Robin
$\mathrm{var}(\kappa(k^*, k))$	0.071	0.038	0.018	0.015	0.035	0.0017	0.0019	0.0023
$\min_k \kappa(k^*, k)$	0.117	0.234	0.234	0.292	0.234	0.4046	0.3774	0.3462

observe each plaintext for 4-bit S-boxes approximately 6.25 times, whereas for the 8-bit case more than the half has not been observed yet. Another reason is that the variance of the signal is not equivalent. In particular, as the HW follows a binomial distribution, we have $Var(\mathrm{HW}(\texttt{Sbox}[T \oplus k]))$ with $T, k \in \mathbb{F}_2^4$ equal to 1 for 4-bit S-boxes and equal to 2 for 8-bit S-boxes. Accordingly, given the same amount of independent additional noise, the SNR using 8-bit S-boxes is twice as high as for 4-bit S-boxes.

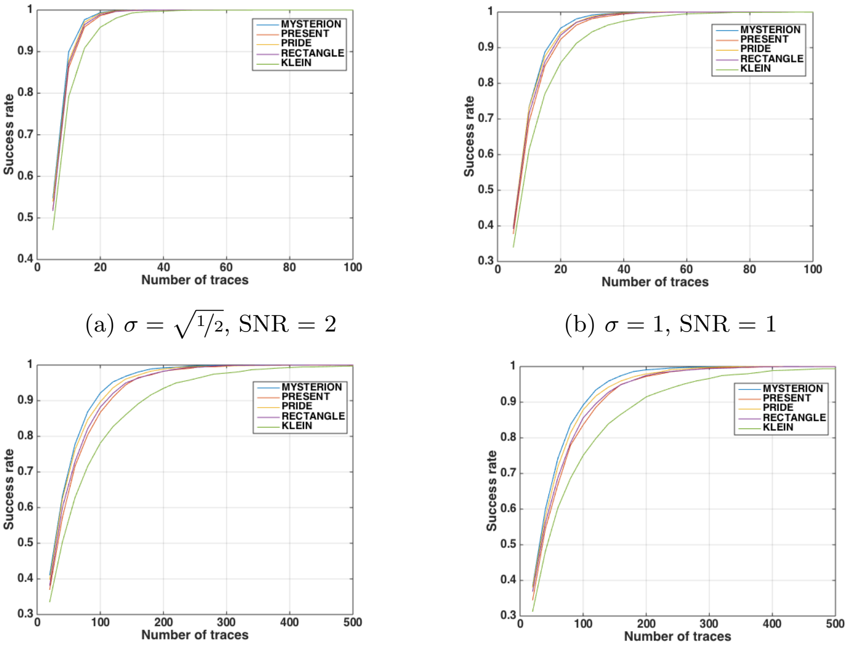
Figures 2 and 3 give the success rate for CPA for various levels of noise, where we simulated the traces as in Eq. (1) with $N \sim \mathcal{N}(0, \sigma^2)$. To be reliable, we use 5 000 independent experiments with randomly chosen T. For 4-bit S-boxes, Figure 2 confirms the ranking given by the confusion coefficient: Mysterion is the easiest to attack and KLEIN the hardest, which is independent of the noise level. Figure 3 shows that all three ciphers behave similarly even for different levels of noise. Accordingly, the (small) differences in the confusion coefficients in Table 1 do not influence the side-channel resistance in practice.

There are two ways to compare the success rates for 4-bit and 8-bit S-boxes in Figures 2 and 3, either having the same additional independent noise (environmental noise) σ or the same SNR. Using the same amount of σ (Figures 2b vs. 3a and 2d vs. 3c), we can observe that AES, Zorro, and Robin perform better than KLEIN and similar to or slightly worse than the others. On the other hand, when comparing the SNR, we observe that AES, Zorro, and Robin behave in a similar way as KLEIN.

3.2 Profiled Side-channel Analysis

Machine learning (ML) is a term encompassing a number of methods that can be used for clustering, classification, regression, feature selection, and other knowledge discovering methods [21]. In supervised machine learning, the algorithm is provided with a set of data instances (i.e., measurements) and data classes (i.e., values of $Y(k^*)$) in a training phase. The goal of this phase is to "learn" the relationship between the instances and the classes in order to be able to reliably map new instances to the classes in the testing phase.

For our study, we use one algorithm per ML family based on the form in which the output function is represented. In particular, we use Naive Bayes as the simplest algorithm that does not have any parameters to tune. Next, from the decision tree family we use the C4.5 algorithm, which is an algorithm considered to be robust to noise. From the perceptron family, we use the Multi



(a) $\sigma = \sqrt{1/2}$, SNR = 2

(b) $\sigma = 1$, SNR = 1

(c) $\sigma = \sqrt{8}$, SNR = $1/8$

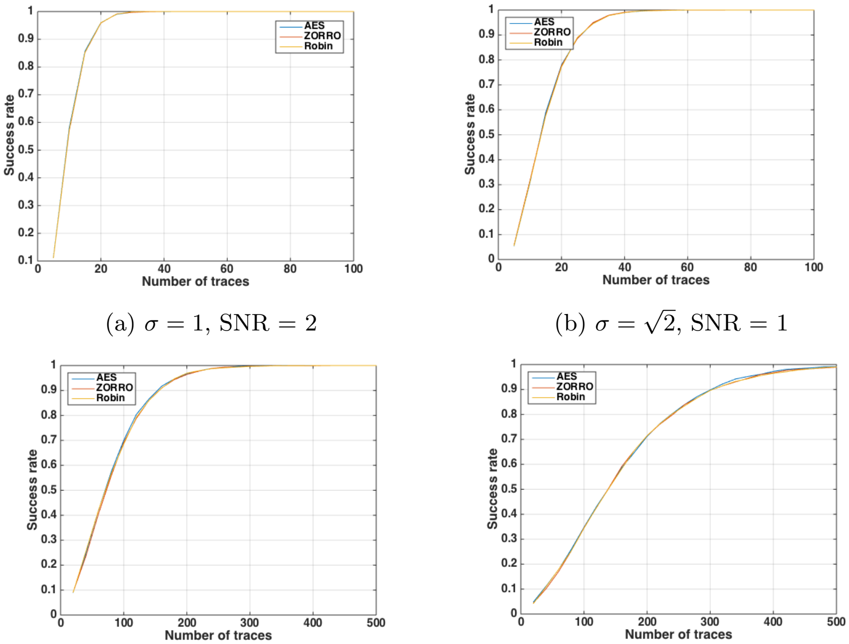
(d) $\sigma = 4$, SNR = $1/16$

Fig. 2: Success rates (ciphers with 4-bit S-boxes)

Layer Perceptron (MLP) algorithm, which represents an advance over the simple perceptron algorithm.

Our experiments are divided in two phases: training and testing (i.e., attacking) with datasets containing 10 000, 30 000, and 50 000 instances. As common for ML techniques we use $2/3$ of the instances for training and $1/3$ for testing (e.g., results for 10 000 instances are obtained with 6 650 training instances and 3 350 instances in the testing phase). On the training set we conduct a 10-fold cross-validation with all the considered parameters. Note that the training phase contains a tuning phase in which we select the best parameters for each algorithm. Due to the lack of space, we do not present results from the training phase but we mention the best obtained parameters that are then used in the testing phase. We also conducted the same set of experiments with more advanced ML techniques – Rotation Forest and Support Vector Machines, but the results did not differ significantly from those presented here.

Note that our simulated measurements only contain one feature (time instance), which is commonly accepted for simulated data, but not usual when using ML techniques or profiled SCA (at least before dimension reduction). If one has at his disposal a sufficient number of measurements with many features and the level of noise is low, previous results confirm that such a scenario is easy for profiled attack. However, if the level of noise is high or the number of mea-



(a) $\sigma = 1$, SNR $= 2$

(b) $\sigma = \sqrt{2}$, SNR $= 1$

(c) $\sigma = 4$, SNR $= 1/8$

(d) $\sigma = \sqrt{32}$, SNR $= 1/16$

Fig. 3: Success rates (ciphers with 8-bit S-boxes)

surements is too low, then the process becomes more cumbersome. Our study shows that even if only a single feature is available (with sufficient information), the attack can be very powerful. Moreover, with the increase in the number of features, the "curse of dimensionality" can appear: as the number of features grow, the classification effort grows exponentially. Common ways to overcome this problem in SCA are dimension reduction techniques like PCA and LDA. Finally, we note that working with only a single feature also makes theoretical analysis, such as probably approximately correct (PAC) learning, easier; we leave this for future work.

Naive Bayes (NB) classifier is a method based on the Bayesian rule (similar to template attacks [22]). Naive Bayes works under the simplifying assumption that the predictor attributes (measurements) are mutually independent among the features given the target class. The existence of highly correlated attributes in a dataset can thus influence the learning process and reduce the number of successful predictions. Additionally, Naive Bayes assumes a normal distribution for predictor attributes and outputs posterior probabilities.

The space complexity for the Naive Bayes algorithm for both the training and the testing phase equals $\mathrm{O}\big(|\mathcal{Y}|Dv\big)$, where $|\mathcal{Y}|$ is the number of classes, D is the number of features, and v is the average number of values for a feature. On the other hand, for the training phase, the time complexity equals $\mathrm{O}\big(QD\big)$

and for the testing phase $\mathrm{O}\big(|\mathcal{Y}|D\big)$, where Q is the number of training examples. Further information about the Naive Bayes algorithm can be found in [23].

C4.5 is the landmark decision tree algorithm [24]. It is a divide-and-conquer algorithm that splits features at tree nodes using the information-based gain ratio criterion. The node splits in further branches if more information is gained (as measured by the gain ratio) by the split than by keeping all the instances at the node. The runtime of the algorithm is $\mathrm{O}\big(D \times Q \times \log Q\big)$, where D is the number of features and Q is the number of instances [25]. The trees are first grown to full length and pruned afterwards in order to avoid data overfitting.

With the C4.5 algorithm we investigate the influence of the confidence factor parameter that is used for pruning, where smaller values relate to more pruning. We tested that parameter in the range $[0.05, 0.4]$ with a step of 0.05. We conducted a separate tuning phase for each noise level and selected a confidence factor of 0.1 for $\sigma = 1$, 0.2 for $\sigma = 3$, and 0.05 for $\sigma = 5$.

Multi Layer Perceptron (MLP) is a feedforward neural network that maps sets of inputs onto sets of appropriate outputs. Multi layer perceptron consists of multiple layers of nodes in a directed graph, where each layer is fully connected to the next one. To train the network, the backpropagation algorithm is used, which is a generalization of the least mean squares algorithm in the linear perceptron. A perceptron is a linear binary classifier applied to the feature vector. Each vector component has an associated weight w_i. Furthermore, each perceptron has a threshold value θ. The output of a perceptron is "1" if the direct sum between the feature vector and the weight vector is larger than zero and "-1" otherwise. A perceptron classifier works only for data that are linearly separable, i.e., if there is some hyperplane that separates all the positive points from all the negative points [21].

MLP must consist of 3 or more layers (since input and output represent two layers) of nonlinearly-activating nodes [26]. We investigate a learning rate parameter in range $[0.05, 0.4]$ with a step of 0.05, a momentum with values $[0.1, 0.2, 0.3, 0.4]$, a training time with values $[400, 500, 600]$, and a validation threshold with values $[10, 20, 30]$. In our experiments we set the number of hidden layers to be equal to $(number\_of\_classes + number\_of\_attributes)/2$, the learning rate is set to 0.1, the momentum applied to the weights during the update is set to 0.2, the training time is set to 500, and the validation threshold to 20.

4-bit vs. 8-bit We highlight with a gray cell if the the Area Under Curve (AUC) [18] is close to 0.5 which means that the algorithm is closer to random guessing. Note that in our study we use PRESENT and AES. However, the results (in particular the accuracy) are not specific to these ciphers but rather to the fact of using 4-bit/8-bit S-boxes, the intermediate states and the binomial distribution of the HW.

In addition to the previous scenario of attacking the HW of the output of the S-box, we first perform classifications on key chunks, directly resulting in 16 and 256 classes. The results are presented in Table 2, showing that the accuracy (given in percentages) for PRESENT is higher than for AES for all levels of noise, which seems natural since PRESENT has a significantly smaller number of classes than AES. However, when comparing the best values directly, one can observe that the difference is rather small (e.g., for $\sigma = 1$: 41.55 vs. 38.33). What is interesting to observe, is that the level of noise has much less impact when comparing $\sigma = 3$ and $\sigma = 5$ than when comparing $\sigma = 1$ and $\sigma = 3$. Finally, we observe that the number of measurements does not play a significant role in this case.

Table 2: Testing results for classifying a key chunk (nibble or byte)

PRESENT: 16 classes

	10,000			30,000			50,000		
Algorithm	$\sigma = 1$	$\sigma = 3$	$\sigma = 5$	$\sigma = 1$	$\sigma = 3$	$\sigma = 5$	$\sigma = 1$	$\sigma = 3$	$\sigma = 5$
NB	41.55	19.94	12.06	42.62	18.68	13.86	41.72	18.53	14.04
C4.5	40.73	14.85	11.79	41.88	15.79	12.05	41.9	16.08	12.76
MLP	40.67	19.3	11.15	41.4	18.3	14.15	40.82	18.24	13.85

AES: 256 classes

	10,000			30,000			50,000		
Algorithm	$\sigma = 1$	$\sigma = 3$	$\sigma = 5$	$\sigma = 1$	$\sigma = 3$	$\sigma = 5$	$\sigma = 1$	$\sigma = 3$	$\sigma = 5$
NB	38.33	12.67	7.42	37.43	13.04	8.23	38.84	13.29	8.47
C4.5	34.88	9.67	7.69	35.71	10.94	7.18	36.25	10.98	7.04
MLP	35.21	10.94	7.11	37.27	13	7.85	38.67	13.2	8.05

Table 4 gives the results for attacking the HW output of the S-box. Again, we observe that the accuracy is higher for PRESENT than for AES, but we notice that for AES the algorithm is rather "randomly" guessing than predicting meaningful classes. This is mainly due to the imbalance of the HWs since they follow a binomial distribution (see Table 3). In particular, for AES with randomly distributed inputs, the HW value 4 is occurring in 27.34% of all events, which is rather high. Therefore, the classifier mainly outputs class 4, giving an accuracy between 27% and 28%. For PRESENT we can see that HW class 2 is occurring in 37.5% of all cases. However, as there are fewer classes in total, the algorithm seems to try to find a reasonable classification.

Table 3: Occurrences of Hamming weights in %

HW	0	1	2	3	4	5	6	7	8
4-bit	6.25	25	37.5	25	6.25	–	–	–	–
8-bit	0.39	3.12	10.93	21.87	27.34	21.87	10.93	3.12	0.39

We additionally investigate the scenario of chosen plaintexts during the profiling phase. Table 5 presents the results for both PRESENT and AES with ex-

Table 4: Testing results for classifying the HW of the S-box output

PRESENT: 5 classes

	10,000			30,000			50,000		
Algorithm	$\sigma = 1$	$\sigma = 3$	$\sigma = 5$	$\sigma = 1$	$\sigma = 3$	$\sigma = 5$	$\sigma = 1$	$\sigma = 3$	$\sigma = 5$
NB	51.27	38.55	37.12	51.17	38.57	37.1	51.04	38.92	37.81
C4.5	50.06	38.82	37.03	51.05	38.16	37.19	50.72	38.73	37.59
MLP	51.27	39.12	37.03	51.07	38.47	37.31	50.57	39	38

AES: 9 classes

	10,000			30,000			50,000		
Algorithm	$\sigma = 1$	$\sigma = 3$	$\sigma = 5$	$\sigma = 1$	$\sigma = 3$	$\sigma = 5$	$\sigma = 1$	$\sigma = 3$	$\sigma = 5$
NB	27.67	27.63	28.18	27.07	27.04	27.52	27.94	27.93	28.04
C4.5	27.76	26.91	27.64	27.07	26.77	27.26	27.94	27.94	28.15
MLP	27.64	27.64	27.21	27.03	27.03	27.47	27.93	27.93	28.33

actly 1 000 measurements for each class, i.e., the total number of measurements equals 5 000 for PRESENT and 9 000 for AES. We can see that the problem of predicting only a subset of classes is not present and again we observe that classifying PRESENT is more accurate than AES.

Table 5: Results with 1 000 measurements per class, HW model

	PRESENT (5 classes)			AES (9 classes)		
Algorithm	$\sigma = 1$	$\sigma = 3$	$\sigma = 5$	$\sigma = 1$	$\sigma = 3$	$\sigma = 5$
NB	49.7	30.55	24.97	45.32	21.85	19.19
C4.5	50.73	30.79	24.06	43.67	21.26	19.36
MLP	50.12	29.7	24.18	44.14	21.82	19.02

4 Conclusions

In this paper, we investigate whether side-channel analysis is easier for lightweight ciphers than e.g. for AES. We cover both profiled and non-profiled techniques. In the case of non-profiled attacks, we evaluate a number of S-boxes appearing in lightweight ciphers using the confusion coefficient and empirical simulations. Interestingly, we see that the 8-bit S-boxes from AES, Zorro, and Robin perform similarly, whereas for 4-bit S-boxes we have a clear ranking, with the S-box of Mysterion being the weakest to attack and the S-box of KLEIN the hardest. Further, we cannot conclude that the 4-bit S-boxes are generally significantly less resistant than the investigated 8-bit S-boxes. For profiled attacks, we analyze several machine learning techniques for PRESENT and AES. Note that in this scenario our results are applicable to all 4-bit and 8-bit S-boxes. Our results show that attacking PRESENT is somewhat easier than attacking AES, with the difference mainly stemming from the varying number of classes in one or the other scenario. Still, that difference is not so apparent as one could imagine. Since we work with only a single feature and yet obtain a good accuracy in a number of test scenarios, we are confident (as our experiments also confirm) that adding more features will render classification algorithms even more powerful, which will result in an even higher accuracy.

Finally, we did not consider any countermeasures for the considered lightweight algorithms, since the capacity for adding countermeasures is highly dependent on the environment (which we assume to be much more constrained than in the case of AES). However, our results show that a smart selection of S-boxes results in an inherent resilience (especially for 4-bit S-boxes).

References

1. Kocher, P.C., Jaffe, J., Jun, B.: Differential Power Analysis. In: Proceedings of CRYPTO'99. Volume 1666 of LNCS., Springer-Verlag (1999) 388–397
2. Gandolfi, K., Mourtel, C., Olivier, F.: Electromagnetic analysis: Concrete results. In: Proceedings of the Third International Workshop on Cryptographic Hardware and Embedded Systems. CHES '01, London, UK, UK, Springer-Verlag (2001) 251–261
3. Hospodar, G., Gierlichs, B., De Mulder, E., Verbauwhede, I., Vandewalle, J.: Machine learning in side-channel analysis: a first study. Journal of Cryptographic Engineering **1** (2011) 293–302 10.1007/s13389-011-0023-x.
4. Lerman, L., Bontempi, G., Markowitch, O.: A machine learning approach against a masked AES - Reaching the limit of side-channel attacks with a learning model. J. Cryptographic Engineering **5**(2) (2015) 123–139
5. Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Springer (2002)
6. Gong, Z., Nikova, S., Law, Y.W. In: KLEIN: A New Family of Lightweight Block Ciphers. Springer Berlin Heidelberg, Berlin, Heidelberg (2012) 1–18
7. Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin, Y., Vikkelsoe, C.: PRESENT: An Ultra-Lightweight Block Cipher. In: CHES. Volume 4727 of LNCS., Springer (September 10-13 2007) 450–466 Vienna, Austria.
8. Albrecht, M.R., Driessen, B., Kavun, E.B., Leander, G., Paar, C., Yalçın, T. In: Block Ciphers – Focus on the Linear Layer (feat. PRIDE). Springer Berlin Heidelberg, Berlin, Heidelberg (2014) 57–76
9. Zhang, W., Bao, Z., Lin, D., Rijmen, V., Yang, B., Verbauwhede, I.: Rectangle: a bit-slice lightweight block cipher suitable for multiple platforms. Science China Information Sciences **58**(12) (2015) 1–15
10. Journault, A., Standaert, F.X., Varici, K.: Improving the security and efficiency of block ciphers based on ls-designs. Designs, Codes and Cryptography (2016) 1–15
11. Gérard, B., Grosso, V., Naya-Plasencia, M., Standaert, F.: Block Ciphers That Are Easier to Mask: How Far Can We Go? [27] 383–399
12. Grosso, V., Leurent, G., Standaert, F.X., Varıcı, K. In: LS-Designs: Bitslice Encryption for Efficient Masked Software Implementations. Springer Berlin Heidelberg, Berlin, Heidelberg (2015) 18–37

13. Fei, Y., Luo, Q., Ding, A.A.: A Statistical Model for DPA with Novel Algorithmic Confusion Analysis. In Prouff, E., Schaumont, P., eds.: CHES. Volume 7428 of LNCS., Springer (2012) 233–250
14. Thillard, A., Prouff, E., Roche, T.: Success through Confidence: Evaluating the Effectiveness of a Side-Channel Attack. [27] 21–36
15. Brier, É., Clavier, C., Olivier, F.: Correlation Power Analysis with a Leakage Model. In: CHES. Volume 3156 of LNCS., Springer (August 11–13 2004) 16–29 Cambridge, MA, USA.
16. NIST/ITL/CSD: Advanced Encryption Standard (AES). FIPS PUB 197 (Nov 2001) `http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf`.
17. Ullrich, M., De Cannière, C., Indesteege, S., Küçük, Ö., Mouha, N., Preneel, B.: Finding Optimal Bitsliced Implementations of 4 4-bit S-Boxes. SKEW 2011 Symmetric Key Encryption Workshop (February 2011)
18. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer Series in Statistics. Springer New York Inc., New York, NY, USA (2001)
19. Guilley, S., Heuser, A., Rioul, O.: A Key to Success - Success Exponents for Side-Channel Distinguishers. In Biryukov, A., Goyal, V., eds.: Progress in Cryptology - INDOCRYPT 2015 - 16th International Conference on Cryptology in India, Bangalore, India, December 6-9, 2015, Proceedings. Volume 9462 of Lecture Notes in Computer Science., Springer (2015) 270–290
20. Picek, S., Papagiannopoulos, K., Ege, B., Batina, L., Jakobovic, D.: Confused by confusion: Systematic evaluation of dpa resistance of various s-boxes. In Meier, W., Mukhopadhyay, D., eds.: Progress in Cryptology – INDOCRYPT 2014: 15th International Conference on Cryptology in India, New Delhi, India, December 14-17, 2014, Proceedings, Cham, Springer International Publishing (2014) 374–390
21. Mitchell, T.M.: Machine Learning. 1 edn. McGraw-Hill, Inc., New York, NY, USA (1997)
22. Chari, S., Rao, J.R., Rohatgi, P.: Template Attacks. In: CHES. Volume 2523 of LNCS., Springer (August 2002) 13–28 San Francisco Bay (Redwood City), USA.
23. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian Network Classifiers. Machine Learning **29**(2) (1997) 131–163
24. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1993)
25. Frank, E., Witten, I.H.: Generating Accurate Rule Sets Without Global Optimization. In Shavlik, J., ed.: Fifteenth International Conference on Machine Learning, Morgan Kaufmann (1998) 144–151
26. Collobert, R., Bengio, S.: Links Between Perceptrons, MLPs and SVMs. In: Proceedings of the Twenty-first International Conference on Machine Learning. ICML '04, New York, NY, USA, ACM (2004) 23–
27. Bertoni, G., Coron, J.S., eds.: Cryptographic Hardware and Embedded Systems - CHES 2013 - 15th International Workshop, Santa Barbara, CA, USA, August 20-23, 2013. Proceedings. In Bertoni, G., Coron, J.S., eds.: CHES. Volume 8086 of Lecture Notes in Computer Science., Springer (2013)