Side-channel Analysis of Lightweight Ciphers: Current Status and Future Directions

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, Kasteelpark Arenberg 10, bus 2452, B-3001 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 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 different side-channel resiliences, where 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 managements, logistics, home automation, surveillance, traffic control, medical monitoring, and many more. Most of these applications have the need for cryptographic secure components which inspired the research on cryptographic algorithms for constraint devices. Lightweight cryptography has been an active research area over the last 10 years where many innovative ciphers have been proposed in order to optimize various performance criteria and have been subject for many comparisons. Lately, the resistance against side-channel attacks has been considered as an additional decisive factor.

Side-channel attacks analyze physical leakage that is unintentionally emitted during cryptographic operations in a device (e.g., power consumption, electromagnetic emanation). 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 by maximizing some statistical distinguisher. So-called profiled side-channel distinguishers assume that the attacker is able to possess an additional device to the one he want to attack, 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.

Side-channel analysis (SCA) for lightweight ciphers is of particular interest not only because of apparent lack of research done up to now, but also because of the interesting properties of S-boxes. Since the nonlinearity property for S-boxes usually used in lightweight ciphers (4×4) can be maximally equal to 4, that means that the difference between the input and the output of an S-box is much smaller than for instance in the case of AES. Therefore, one could conclude that from that aspect, SCA for lightweight ciphers must be more difficult than for standard ciphers. However, from the other side, if we consider profiled attacks where the distinguisher "learns" about the leakage model, the number of possible classes is significantly lower, which indicates that profiled SCA must be easier than for standard ciphers. Indeed, for 4×4 S-boxes, there are only 5 classes if considering the Hamming weight (HW) or 16 classes if considering one key nibble directly. On the other hand, for 8×8 S-box, there are 9 HW classes and 256 subkey classes. Besides the huge 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 Substitution-permutation Network (SPN) ciphers with 4-bit S-boxes such as KLEIN [1], PRESENT [2], PRIDE [3], RECTANGLE [4], Mysterion [5] as well as ciphers with 8-bit S-boxes: AES [6], ZORRO [7], Robin [8].

In the non-profiled scenario we first investigate the relationship between different key hypotheses with the confusion coefficient, which has been introduced in [9] for a 1-bit model and extended to multi-models in [10]. Using specific properties (like the minimum value and the variance) we can give a preliminary classification regarding the side-channel resistance. Furthermore, using simulated data for various signal-to-noise ratios (SNR) we give empirical results for Correlation Power Analysis (CPA) [11] and discuss the difference between attacking 4-bit and 8-bit S-boxes. Moreover, we compare several machine learning (i.e., profiled) techniques for PRESENT and AES ciphers where we show that attacking PRESENT is somewhat easier than attacking AES. Naturally, our results should be considered only as a first step in the side-channel analysis of lightweight ciphers.

Road Map This paper is organized as follows. Section 2 gives basic information on ciphers and exploitations we investigate. Next, in Section 3 we discuss Correlation Power Analysis (CPA), confusion coefficient, and profiled side-channel analysis. Since we consider traces consisting of only one feature, our machine learning attack – Naive Bayes also corresponds to template attack, which is a well-known profiled attack and one that is also the most powerful from the information theoretic point of view [12]. Following that, we discuss several possible research avenues and finally, in Section 4 we conclude with brief remarks.

2 Ciphers & Exploitations

2.1 Investigated Ciphers

Next, we briefly introduce several ciphers we investigate in this paper where the first three ciphers have 8-bit S-boxes while the rest have 4-bit S-boxes.

AES [6] The Advanced Encryption Standard (AES) has been standardized by NIST in 2001 [13]. It has an SPN structure with an internal fixed block size of 128-bits represented as 4×4 byte matrix and a variable key size of 128, 192 or 256 bits. 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` where in the last round the `MixColumns` is omitted.

Zorro [7] Zorro is a modified version of the AES with a special focus on the S-box that is intended to be easy to mask. Therefore, fewer calls to the S-box are made during each round and the S-box has been modified to minimize the number of multiplications necessary to compute it. Besides, the execution is split into "steps" of 4 rounds and the key (simply the master key) is added only at the end of a step. Like AES each round is made of 4 operations: `SB*` which is a variant of the SubBytes operation where the S-box is only applied to one row of the 4×4 bytes internal state, `AC` is a round constant addition similar to the one used in the LED cipher [14], `SR`, and `MC` which are identical to to `ShiftRows` and `MixColumns`.

Robin [8] Robin is one instance of so-called "LS-designs" where the internal state of the cipher is a matrix of $s \times L$ bits. The non-linear layer consists in the parallel applications of a $s \times s$ bits ($s = 8$) permutation on each column of the matrix and they are chosen so as to be efficiently implemented in a bit-sliced fashion. The linear layer consists in the application of a linear $L \times L$ bits ($L = 16$) permutation on each line of the matrix. Robin is a 128-bits block cipher and intended to ease the application of masking countermeasures and thus to help thwart side-channel attacks when this cipher is implemented on a micro-controller. We note that Robin uses involutions for both L-Box and its S-box.

KLEIN [1] KLEIN is an AES-like lightweight block cipher, designed for 64, 80, and 98-bit variations. The substitution stage uses 16 similar involutive (i.e., $S(x) = y$, $S(y) = x$ and $S(S(x)) = x$) S-boxes. The advantage of using an involutive s-box is the reduction of extra cost of inverse implementation which leads to an efficient serialization. Similar to AES, each encryption round consists of `AddRoundKey`, `SubNibbles`, `RotNibbles`, and `MixNibbles`, followed by a final key addition after the encryption rounds.

PRESENT [2] 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 the key addition `AddRoundKey`, followed by the substitution (`sBoxLayer`) and permutation layer (`pLayer`). The permutation layer is designed to match the effects of the combination between ShiftRows and MixColumns of AES. A final key addition is performed after the encryption rounds.

PRIDE [3] PRIDE has been optimized for 8-bit micro-controllers with a special focus on the linear layer of the cipher. The S-box is an involution to further limit the overhead implied by the implementation of both encryption and decryption. Furthermore, PRIDE is designed in a bit-sliced fashion to minimize the number of instructions necessary to evaluate it.

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

Mysterion [5] The cipher is based on the LS-design principles introduced by the designers of Fantomas and Robin and combines it with an AES-like structure to increase the security level. The internal state of the block cipher is organized into a 4×32 bit matrix for Mysterion-128, which are further subdivided into 4 4×8 blocks. A round consists in the following operations: `S-box layer`, and `L-Box layer`, and `ShiftColumns`. The `S-box layer` is a 4-bit S-box called "Class 13" introduced in [15] that is applied in parallel to each column of the internal state.

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 a 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 involved 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.,

$$X = \mathrm{HW}(\texttt{S-box}[T \oplus k^*]) + N, \tag{1}$$

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

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.

3 (Empirical) Side-Channel Evaluation

3.1 CPA & Confusion Coefficient

Correlation Power Analysis (CPA) [11] 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, for each key hypothesis $k \in \mathbb{F}_2^n$ one has:

$$Y(k) = \mathrm{HW}(\texttt{S-box}[T \oplus k]). \tag{2}$$

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. Finally, he decides for the key $\hat{k}$ maximizing the correlation, i.e.:

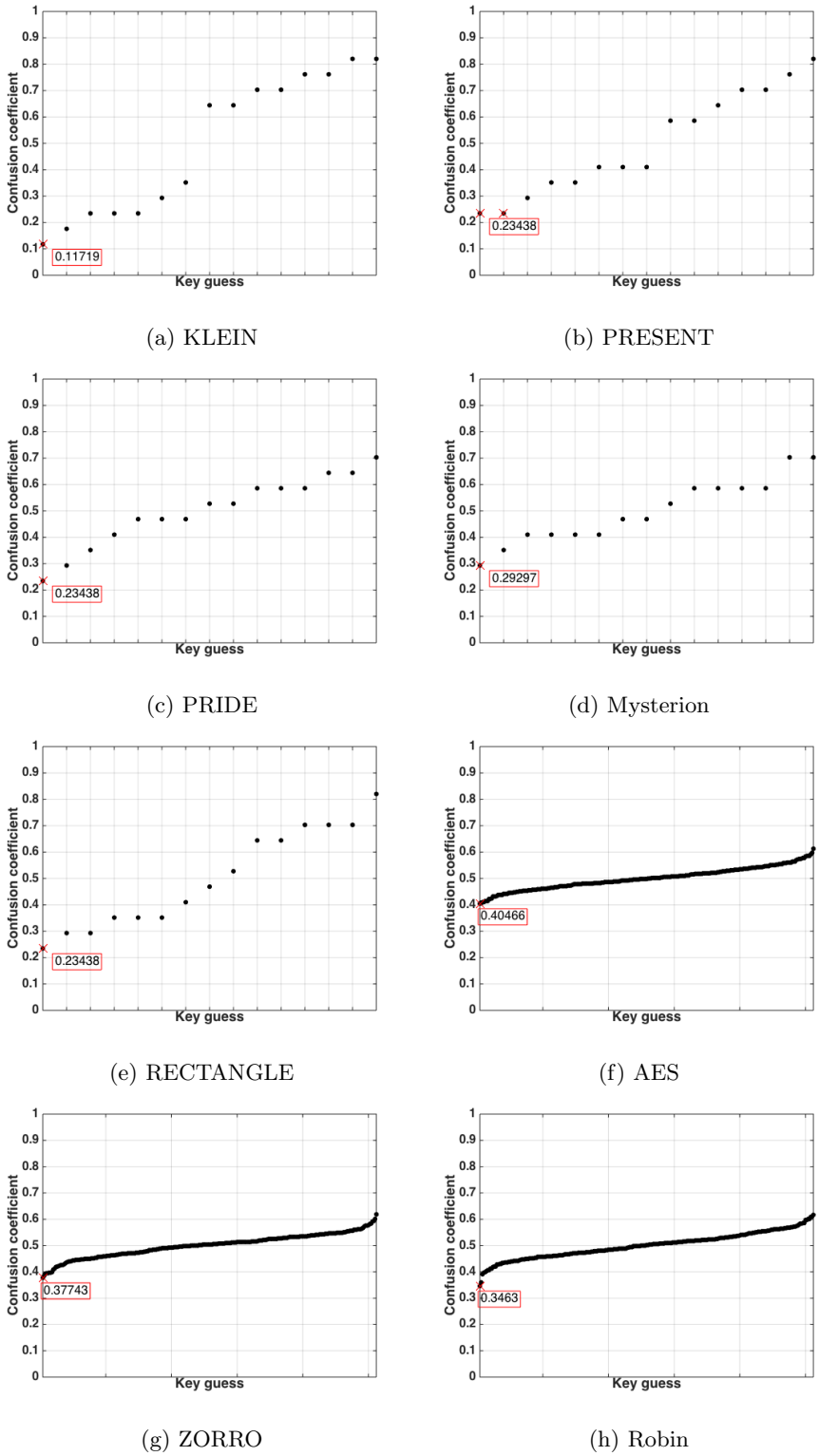
$$\hat{k} = \arg\max_k \rho(X, Y(k)) \tag{3}$$

with ρ being the Pearson correlation coefficient [16].

Before presenting 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 [9, 10]

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

where the expectation is taken over T.



(a) KLEIN (b) PRESENT

(c) PRIDE (d) Mysterion

(e) RECTANGLE (f) AES

(g) ZORRO (h) Robin

Fig. 1: Confusion coefficients

Figures 1a to 1e show the confusion coefficient for the investigated ciphers using 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 them in an increasing order of magnitude. One can observe that the distribution of confusion coefficient 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 [9,10] showed that the theoretical success rate of CPA can be divided into three factors: confusion coefficient, signal to noise ratio (SNR), and number of measurements, but without further investigating the confusion coefficient in particular. The authors in [17] 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 [18] using $var(\kappa(k^*, k))$ as a criteria, where smaller values indicate lower side-channel security. All values for 4-bit are given in Table 1, showing that 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 2 indicate that the side-channel resistance of the investigated 8-bit S-boxes is lower than for ones with 4-bit. Recall that the confusion coefficient measure the relationship between different key hypotheses. Now, as for 8-bits we have also 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 $Y(k) \in [0, 1, \ldots, 4]$.

Table 1: Properties of $\kappa(k^*, k)$ (4-bit S-boxes)

	KLEIN	PRESENT	PRIDE	Mysterion	RECTANGLE
$\mathrm{var}(\kappa(k^*, k))$	0.071	0.038	0.018	0.015	0.035
$\min_k \kappa(k^*, k)$	0.117	0.234	0.234	0.292	0.234

Table 2: Properties of $\kappa(k^*, k)$ (8-bit S-boxes)

	AES	ZORRO	Robin
$\mathrm{var}(\kappa(k^*, k))$	0.0017	0.0019	0.0023
$\min_k \kappa(k^*, k)$	0.4046	0.3774	0.3462

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 variance of the signal is not equivalent. In particular, as the HW follows a binomial distribution, we have $Var(\mathrm{HW}(\texttt{S-box}[T \oplus k]))$ with $T, k \in \mathbb{F}_2^4$ and 4-bit S-boxes is 1, whereas we have a variance $= 2$ for the counterpart with 8-bit. Accordingly, given the same amount of independent additional noise, the SNR using 8-bit S-boxes is twice as much as for 4-bit. Another

reason is that the confusion coefficient is theoretical (i.e., holding for $Q \to \infty$). However, 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 observe each plaintext for 4-bit approximately 6.25 times, whereas for the 8-bit case more than the half has not been observed yet.

Figures 2 and 3 give the success rate for CPA for various levels of noise, where we simulated the traces as in Eq. (1). 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 holds independently on the level of noise. 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 2 do not influence the side-channel resistance in practice.

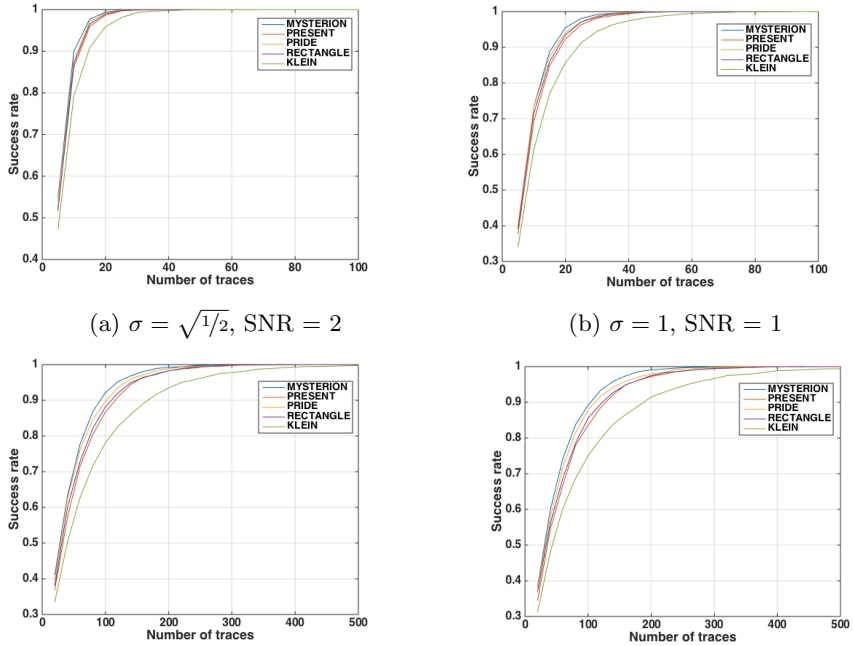
There are two ways to compare the success rates for 4-bit and 8-bit S-boxes in Figures 2 and 3, either having the same additional independent noise (environmental noise) σ or the same SNR. Using the same amount of σ (Figures 2b vs 3a and 2d vs 3c) we can observe that AES, ZORRO, and Robin perform better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

3.2 Profiled Side-channel Analysis

In profiled SCA, we investigate several machine learning techniques where Naive Bayes also corresponds to template attack (since we work with only a single feature). 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 [19]. All those methods can be classified in several ways; a common classification method divides the techniques into supervised and unsupervised machine learning methods. In supervised machine learning, the algorithm is provided with a set of data instances and data classes 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. On the contrary, in unsupervised machine learning, an algorithm does not know the classes a priori and needs to infer the relationships between the data and the unlabeled classes.

In the ML algorithm selection process, we use one algorithm per ML family based on the representation of the output function. Accordingly, we use Naive Bayes as the simplest algorithm that does not have any parameters to tune and corresponds to template attack since there is only one feature. Next, from the decision tree family we use C4.5 algorithm, which is an algorithm considered to be

[4] Note that our study does not include the complexity to go from a round key to the master key. Furthermore, in the case of ZORRO and Robin less S-box calls are made giving less points of exploitations.



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

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

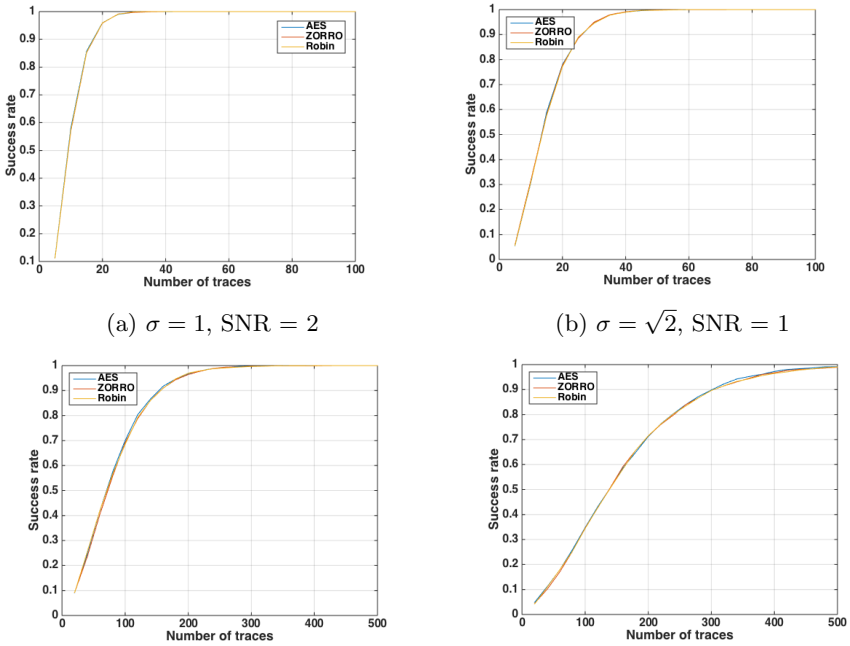
(c) $\sigma = \sqrt{8}$, SNR = $1/8$

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

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

robust to noise. From the perceptron family, we use the Multi Layer Perceptron (MLP) algorithm, which has an advance over the simple perceptron algorithm. Note that MLP can be considered as deep learning algorithm if it has large number of hidden layers.

In all our experiments, we conduct investigation in two phases: training and testing. The sizes of datasets are 10 000, 30 000, and 50 000. We divide our datasets in a ratio of 2:1 where we take the bigger set as the training set (2/3 of the data) and the smaller set for testing (1/3 of the data). On the training set we conduct a 10-fold cross-validation with all the considered parameters. In the 10-fold cross-validation, the original sample is first randomly partitioned into 10 equal sized subsets. Then, a single subsample is selected to validate the data while the remaining 9 subsets are used for training. The cross-validation process is repeated 10 times where each of the 10 subsamples is used once for validation. The obtained results are then averaged to produce an estimation. Note that alongside training phase, we also conduct a tuning where 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 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.



(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)

Naive Bayes (NB) classifier is a method based on the Bayesian rule (similar to template attack) but Naive Bayes works under a simplifying assumption that the predictor attributes (measurements) are mutually independent among the D features given the target class. 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 normal distribution for predictor attributes. A Naive Bayes classifier outputs posterior probabilities as a result of the classification procedure.

Bayes' formula is used to compute the posterior probability of each class value y given the vector of N observed attribute values x:

$$p(Y = y|\boldsymbol{X} = \boldsymbol{x}) = \frac{p(Y = y)p(\boldsymbol{X} = \boldsymbol{x}|Y = y)}{p(\boldsymbol{X} = \boldsymbol{x})}, \tag{5}$$

where $\boldsymbol{X} = \boldsymbol{x}$ represents the event that $\boldsymbol{X}_1 = \boldsymbol{x}_1 \wedge \boldsymbol{X}_2 = \boldsymbol{x}2 \wedge \ldots \wedge \boldsymbol{X}_N = \boldsymbol{x}_N$. Because this event is a conjunction of conditionally D independent events, individual probabilities can be multiplied. Moreover, as $p(\boldsymbol{X} = \boldsymbol{x})$ does not depend on the class y, the Naive Bayes classifier reduces to:

$$p(Y = y|X = x) = p(Y = y) \prod_{i=1}^{D} p(X_i = x_i|Y = y). \tag{6}$$

Space complexity for Naive Bayes algorithm for both training and testing phase equals $\mathrm{O}\big(|\mathcal{Y}|Dv\big)$, where $|\mathcal{Y}|$ is the number of classes, D is the number of attributes, and v is the average number of values for an attribute. On the other hand, time complexity for the training phase equals $\mathrm{O}\big(ND\big)$ and for the testing phase is $\mathrm{O}\big(|\mathcal{Y}|D\big)$ where N is the number of training examples. Further information about Naive Bayes algorithm can be found in [20].

C4.5 is the landmark decision tree algorithm developed by R. Quinlan [21]. It is a divide-and-conquer algorithm that splits attributes at tree nodes using the information-based gain ratio criterion. The node splits on further branches if more information is gained (as measured by gain ratio) by the split than by keeping all the instances at the node. The runtime of the algorithm is $\mathrm{O}\big(f\big) \times q \times \log q$ where f is the number of features and q is the number of instances [22]. 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 confidence factor parameter that is used for pruning where the 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 separate tuning phase for each noise level and select confidence factor of 0.1 for σ_1, 0.2 for σ_3, and 0.05 for σ_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 a multiple layers of nodes in a directed graph, where each layer is fully connected to the next one. To train the network, 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. For each vector component there is a weight w_i associated to it. Furthermore, each perceptron has a threshold θ. The output of a perceptron is "1" if the direct sum between the feature vector and weight vector is larger than zero and "-1" otherwise. A perceptron classifier works only for data that is linearly separable, i.e., if there is some hyperplane that separates all the positive points from all the negative points [19].

Differing from standard linear perceptrons, MLPs can distinguish data that are not linearly separable. MLP must consist of 3 or more layers (since input and output represent two layers) of nonlinearly-activating nodes [23]. We investigated learning rate parameter in range $[0.05, 0.4]$ with step 0.05, momentum with values $[0.1, 0.2, 0.3, 0.4]$, training time with values $[400, 500, 600]$, and 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, learning rate is set to 0.1, momentum applied to weights during update to 0.2, training time is set to 500, and validation threshold to 20.

4-bit vs 8-bit In Tables 3 and 4 we give results for PRESENT and AES where the results represent the accuracy in percentage. The results where the Area Under an ROC Curve (AUC) measure equals to 0.5 are given in cells with gray

background color. AUC close to 1 represents a good test, while value close to 0.5 represents a random guessing.

The results obtained show that attacking the PRESENT cipher is easier than attacking the AES in all the investigated scenarios. Those results are to be expected since PRESENT has significantly smaller number of classes than the AES while the probability of a successful class guess is inversely proportional with the number of classes. Furthermore, we observe that Naive Bayes is the most successful technique in the prevailing number of investigated scenarios. In Table 3 we see that C4.5 does not perform well with the highest level of noise (i.e., for σ_5 case).

What is interesting to observe is that the level of noise has much less impact when comparing σ_3 and σ_5 cases than when comparing σ_1 and σ_3. Besides that, the differences among the tested algorithms are in general too small to have a statistical significance so it is not possible to determine the best performing algorithm. When considering all 16 classes for one key nibble (going into one S-box), we see that the success rate is still sufficiently high to consider all tested algorithms successful. Finally, we observe that the number of measurements does not play a significant role for this case.

Table 3: Testing results for PRESENT

5 classes

Algorithm	10k			30k			50k		
	σ_1	σ_3	σ_5	σ_1	σ_3	σ_5	σ_1	σ_3	σ_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

16 classes

Algorithm	10k			30k			50k		
	σ_1	σ_3	σ_5	σ_1	σ_3	σ_5	σ_1	σ_3	σ_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

In Table 4 we give results for the AES scenario. Here, we can observe that when considering HW classes, all considered techniques encounter difficulties, which is due to a huge imbalance in the number of measurements belonging for instance to HW 0 or 8 and HW 4. This is especially apparent in several cases where none of the tested algorithms is able to give dependable results but we see that the increase in the number of measurements is helping alleviate this problem. We note that we also observed an interesting behavior where adding noise actually even helps in better classification since it gives more diversity in data. When considering 256 classes of a key byte directly, we have more

reliable measurements due to a better distribution of instances, but naturally the classification accuracy is lower than in the PRESENT case.

Finally, we note that we also conducted a number of experiments designed to give us indication of the linear separability of classes [24] where the results confirm that all our scenarios have classes that are not linearly separable, i.e, that are not easy to classify.

Table 4: Testing results for AES

9 classes

Algorithm	10k			30k			50k		
	σ_1	σ_3	σ_5	σ_1	σ_3	σ_5	σ_1	σ_3	σ_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

256 classes

Algorithm	10k			30k			50k		
	σ_1	σ_3	σ_5	σ_1	σ_3	σ_5	σ_1	σ_3	σ_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

When considering SCA and more precisely profiled attacks as the most powerful SCA attacks, there is a number of conditions that can render attacks more difficult, such as noise, large number of classes, the imbalance between the number of measurements belonging to each class or insufficient number of measurements in the profiling phase. However, we must be aware that such drawbacks are also present in other domains but often it seems those domains handle such drawbacks more successfully. Indeed, there is a number of techniques one can employ to deal with any of the aforesaid problems (as well as many other) but such techniques at least for now elude the SCA community. Therefore, we believe that meticulous investigations can advance the field of SCA significantly. We mention here only a number of concepts we deem to have significant potential: fuzzy classification, feature reduction/construction techniques, imbalanced learning, semi-supervised learning, etc. Finally, only by further advancing the attacks, we will be able to design more successful countermeasures.

4 Conclusions

In this paper, we investigate whether side-channel analysis of lightweight ciphers is easier than for instance when considering AES using 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 saw that the 8-bit S-boxes from AES, ZORRO, and Robin performed similar and we further cannot conclude that the 4-bit S-boxes are generally easier to attack than the 8-bit S-boxes. Furthermore for profiling, we analyze several machine learning techniques for PRESENT and AES. Our results show that attacking PRESENT is somewhat easier than attacking AES, the difference mainly stemming from the varying number of classes in one or other scenario. Still, that difference is not so apparent as one could imagine. This leaves us with a conclusion that attacking lightweight ciphers is not easy, but care should be taken if we consider attackers as powerful as for instance for the AES case. Since we work here with only a single feature scenario and yet in a number of test cases we obtain good accuracy we are confident that adding more features will render classification algorithms even more powerful which will result in an even higher accuracy.

Finally, in this paper we did not consider any countermeasures since that is highly dependent on the environment (which we must assume to be much more constrained than when considering for instance AES). However, our results show that even considering the inherent resilience of S-boxes one can make the cipher more or less resistant against SCA, which can be considered as a countermeasure that comes for free, i.e., with only a smart selection of the S-boxes.

References

1. Gong, Z., Nikova, S., Law, Y.W.: Klein: A new family of lightweight block ciphers. In Juels, A., Paar, C., eds.: RFID. Security and Privacy: 7th International Workshop, RFIDSec 2011, Amherst, USA, June 26-28, 2011, Revised Selected Papers, Berlin, Heidelberg, Springer Berlin Heidelberg (2012) 1–18
2. 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.
3. Albrecht, M.R., Driessen, B., Kavun, E.B., Leander, G., Paar, C., Yalçın, T.: Block ciphers – focus on the linear layer (feat. pride). In Garay, J.A., Gennaro, R., eds.: Advances in Cryptology – CRYPTO 2014: 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I, Berlin, Heidelberg, Springer Berlin Heidelberg (2014) 57–76
4. 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
5. 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
6. Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Springer (2002)
7. Gérard, B., Grosso, V., Naya-Plasencia, M., Standaert, F.: Block Ciphers That Are Easier to Mask: How Far Can We Go? [25] 383–399
8. Grosso, V., Leurent, G., Standaert, F.X., Varıcı, K.: Ls-designs: Bitslice encryption for efficient masked software implementations. In Cid, C., Rechberger, C., eds.: Fast Software Encryption: 21st International Workshop, FSE 2014, London, UK, March 3-5, 2014. Revised Selected Papers, Springer Berlin Heidelberg (2015) 18–37

9. 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
10. Thillard, A., Prouff, E., Roche, T.: Success through Confidence: Evaluating the Effectiveness of a Side-Channel Attack. [25] 21–36
11. 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.
12. 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.
13. NIST/ITL/CSD: Advanced Encryption Standard (AES). FIPS PUB 197 (Nov 2001) `http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf`.
14. Guo, J., Peyrin, T., Poschmann, A., Robshaw, M.J.B.: The LED Block Cipher. In Preneel, B., Takagi, T., eds.: CHES. Volume 6917 of LNCS., Springer (2011) 326–341
15. Ullrich, M., De Cannire, C., Indesteege, S., Mouha, N., Preneel, B.: Finding Optimal Bitsliced Implementations of 4 4-bit S-Boxes. SKEW 2011 Symmetric Key Encryption Workshop (February 2011)
16. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer Series in Statistics. Springer New York Inc., New York, NY, USA (2001)
17. 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
18. 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
19. Mitchell, T.M.: Machine Learning. 1 edn. McGraw-Hill, Inc., New York, NY, USA (1997)
20. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian Network Classifiers. Machine Learning **29**(2) (1997) 131–163
21. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1993)
22. 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
23. 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–
24. Ho, T.K., Basu, M.: Complexity measures of supervised classification problems. IEEE Transactions on Pattern Analysis and Machine Intelligence **24**(3) (Mar 2002) 289–300
25. 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)