Confused by Confusion: Systematic Evaluation of DPA Resistance of Various S-boxes

No Author Given

No Institute Given

Abstract. When studying the DPA resistance of S-boxes, the research community is divided in their opinions on what properties should be considered. So far, there exist only a few properties that aim at expressing the resilience of S-boxes to side-channel attacks. Recently, the confusion coefficient property was defined with the intention to characterize the resistance of an S-box. However, there exist no experimental results or methods for creating S-boxes with a "good" confusion coefficient property. In this paper, we employ a novel heuristic technique to generate S-boxes with "better" values of the confusion coefficient in terms of improving their side-channel resistance. We conduct extensive side-channel analysis and detect S-boxes that exhibit previously unseen behavior. For the 4×4 size we find S-boxes that belong to optimal classes, but they behave like linear elements when running a CPA attack, therefore keeping an attacker from achieving 100% success rate on recovering the key.

1 Introduction

The security of block-ciphers is a very important area in modern cryptography. Rather than more traditional linear [1] and differential cryptanalysis [2], the most practical attacks today belong to side-channel analysis (SCA), targeting actual implementations of block ciphers in software or hardware. Side-channel analysis relies on the physical leakages from the actual implementation and its efficiency is much greater than the one of linear or differential cryptanalysis [3]. To improve the algorithm resiliency to SCA, there exist many possible countermeasures such as various hiding and masking schemes [4]. However, all countermeasures come with a substantial increase in cost due to larger memory and area requirements and the decrease in performance of the algorithm implemented.

Block ciphers of today are typically designed either as a Feistel network or as a Substitution-Permutation network (SPN). In both design principles, S-boxes (or Substitution boxes), are usually the only nonlinear part. Therefore, S-boxes have a fundamental role in the security of most modern block ciphers [5] and their "good" cryptographic properties are of utmost importance for the security of encryption schemes in numerous applications. Although there exist a plethora of cryptographic properties defined for S-boxes in the literature, there are only a handful properties related to the SCA resistance. Currently, the properties related with SCA are SNR (DPA) [6], transparency order [7], criterion for the S-box resilience against CPA attacks [8] and as the newest measure, confusion

coefficient [9, 10]. Considering the transparency order, which was heavily investigated so far, results from different groups are somewhat conflicting [11–13]. Yet in all previous works the transparency order seems to have a certain influence on DPA resistance. For example, in the 4×4 S-boxes case (as used in e.g. PRESENT [20]), it is shown that one can obtain S-boxes that have better DPA resistance, while retaining properties of optimal S-boxes [14].

Nevertheless, numerous inconclusive results for different ciphers, platforms and leakage models have led to an attempt to redefine the transparency order measure [15]. This new approach also remains to be convincing in practical results.

When considering 8×8 S-boxes, previous results on transparency order suffer from two major drawbacks. The first drawback stems from the fact that an improved S-box (in regards to the transparency order property) may result in deterioration of many properties related with linear and differential resistance of the algorithm (e.g. nonlinearity and δ-uniformity). The second major drawback is the necessity to implement such improved S-boxes as lookup tables. For instance, an improved AES S-box (e.g. derived from heuristic search) loses the algebraic properties that are important for compact implementations [16, 17]. Still, there are possible settings where the improvement in DPA resistance makes up for the aforementioned drawbacks. In contrast to this, when considering 4×4 S-boxes, the situation is improved since both implementation options, as a lookup table and as a Boolean function in hardware, are viable.

In this paper, we generate S-boxes with an improved confusion coefficient and we show that this also improves DPA resistance. In order to confirm that, we conduct simulated and practical side-channel analysis on those improved S-boxes.

1.1 Related Work

In 2004, Guilley presents SNR (DPA) measure which is, to our best knowledge, the first property related with DPA resistance [6]. One year later, Prouff presents the transparency order property which is the first DPA-related property for the multi-bit case [7]. The usefulness of the result is somewhat questionable. On one hand, it is important to consider the issue but on the other hand considering this as a countermeasure seems to be doomed, when following the construction outlined in the paper. However the idea is valuable, as in 2012 several papers revisit the topic [11–14].
Apart from the transparency-related efforts, a new line of research by Yunsi Fei et al. [9,10,18,19] attempts to actually model the behavior of a cryptographic implementation with respect to side-channel resistance. Starting from DPA-related models [9, 18] they expand the concept to CPA attacks [10] and masking [19], while offering a probabilistic model for side-channel analysis.

1.2 Our Contribution

It is evident that using side-channel leakages is a powerful means of cryptanalysis. Therefore, it is important to find effective and efficient countermeasures (or combinations of them) that can prevent this type of cryptanalysis. This paper investigates the option of using heuristically-created S-boxes to increase the resistance to implementation attacks. More specifically, we are the first to use the confusion coefficient as a cipher design parameter. With the assistance of genetic algorithms, we create 4×4 and 8×8 S-boxes that obtain improved confusion coefficient property. For the 4×4 case, we create S-boxes that have increased resistance in the form of "ghost peaks" [4] (defined here as "Phantom" S-boxes), while remaining in optimal classes [5]. For the 8×8 case, we obtain increased resistance, albeit at the cost of classical cryptanalytic properties like nonlinearity and δ-uniformity. We evaluate the newly generated S-boxes in a real world scenario: we implement variants of PRESENT [20] and AES [21] ciphers that employ the new S-boxes and we perform side-channel analysis on them.

The remainder of this paper is organized as follows. We present necessary information about relevant cryptographic properties of S-boxes in Sect. 2. In Sect. 3 we give explanations of the algorithms we use and the analysis of several S-boxes with improved confusion coefficient property. We also compare the properties of our new S-boxes with the ones obtained from random search as well as with the original S-boxes. The side-channel resistance of the newly proposed S-boxes is presented in Sect. 4, both with simulations and also with experiments on a real target. We conclude the paper in Section 5.

2 Preliminaries

In this section we present some background information about side-channel analysis and cryptographic properties of S-boxes that are of interest.

2.1 Side-channel Analysis

Cryptographic devices, such as smart cards, RFID tags etc. have become pervasive in our lives as they are used in numerous everyday applications. However, a great deal of care should be taken as it is known that these devices can have physical channels, which leak the information that they process. These leakages can be exploited by an adversary monitoring side channels such as timing [22], power consumption [22, 23], electromagnetic emanation [24] or sound [25]. This kind of attacks enables the attacker to obtain otherwise unknown information on the workings of the underlying algorithm, therefore leading to practical attacks on even real-world cryptosystems. As these attacks are the most practical ones among many cryptanalysis efforts, this area attracted quite some interest in the literature in the past decade. There are recent publications in the literature that focus on modeling the physical leakage of an algorithm with the assumption of

a certain leakage model [7,18]. This line of research is aimed to provide a way to evaluate the side-channel resistance of an algorithm at the design phase to help cryptographers improve the overall security of a cryptosystem.

2.2 Cryptographic Properties of S-boxes

As mentioned previously, there exist several properties of S-boxes where each property relates to a certain cryptographic attack. However, here we concentrate only on several basic properties such as bijectivity, linearity and δ-uniformity (as given in [5]) and of course the new measure, confusion coefficient.

The addition modulo 2 is denoted as " $\oplus$ ". The inner product of vectors $\bar{a}$ and $\bar{b}$ is denoted as $\bar{a} \cdot \bar{b}$ and equals $\bar{a} \cdot \bar{b} = \oplus_{i=1}^{n} a_i b_i$.

Function F, called S-box or vectorial Boolean function, of size (n, m) is defined as any mapping F from $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$ [7]. When m equals 1 the function is called Boolean function. Boolean functions f_i, where $i \in \{1, ..., m\}$, are coordinate functions of F, where every Boolean function has n variables.

Hamming weight HW of a vector $\bar{a}$, where $\bar{a} \in \mathbb{F}_2^n$, is the number of non-zero positions in the vector.

An (n, m)-function is called **balanced** if it takes every value of $\mathbb{F}_2^m$ the same number 2^{n-m} of times [26].

Linearity L_f can be defined as [27]

$$L_f = max_{\substack{\bar{a} \in \mathbb{F}_2^n \\ \bar{v} \in \mathbb{F}_2^{m*}}} |W_F(\bar{a}, \bar{v})|. \tag{1}$$

where $W_F(\bar{a}, \bar{v})$ is Walsh transform of F [7].

$$W_F(\bar{a}, \bar{v}) = \sum_{\bar{x} \in \mathbb{F}_2^n} (-1)^{\bar{v} \cdot F(\bar{x}) \oplus \bar{a} \cdot \bar{x}}. \tag{2}$$

Nonlinearity N_F of an (n, m)-function F is equal to the minimum nonlinearity of all non-zero linear combinations $\bar{b} \cdot F$, where $\bar{b} \neq 0$, of its coordinate functions f_i [3].

$$N_F = 2^{n-1} - \frac{1}{2} max_{\substack{\bar{a} \in \mathbb{F}_2^n \\ \bar{v} \in \mathbb{F}_2^{m*}}} |W_F(\bar{a}, \bar{v})|, \tag{3}$$

Differential delta uniformity δ represents the largest value in the difference distribution table without counting the value 2^n in the first row and first column position [2, 26, 28].

Recently, Fei et al. introduced a new property that relates with the DPA resistance of S-boxes - **confusion coefficient** [9,10,18,19]. They give a probabilistic model that encompasses the three core parameters of a side-channel attack: the target device, the number of traces and the algorithm under examination. That model manages to separate these three elements and grants us the freedom to explore the cipher design space by solely focusing on the cipher algorithm.

$$\kappa(k_i, k_j) = Pr[(\psi|k_c) \neq (\psi|k_g)] \tag{4}$$

$$\widetilde{\kappa}(k_c, k_{g_i}, k_{g_j}) = Pr[(\psi|k_{g_i}) = (\psi|k_{g_j}), (\psi|k_{g_i}) \neq (\psi|k_c)] \tag{5}$$

$$\mathbf{K} : (N_k - 1) \times (N_k - 1),\ \mathbf{K_{ij}} = \begin{cases} \kappa(k_c, k_g), & \text{if } i = j \\ \widetilde{\kappa}(k_c, k_{g_i}, k_{g_j}), & \text{if } i \neq j \end{cases} \tag{6}$$

$$\mu_{\bar{Y}} = 2 \times \epsilon \times \bar{\kappa} \tag{7}$$

where $\bar{\kappa}$ is the diagonal vector of $\mathbf{K}$.

$$\mathbf{\Sigma_Y} = 16 * \sigma_W / N_m \times \mathbf{K} + 4 \times \epsilon^2 / N_m \times (\mathbf{K} - \bar{\kappa} \times \bar{\kappa}^T) \tag{8}$$

$$SR_{DPA} = \Phi_{N_k - 1}(\sqrt{N_m} \mathbf{\Sigma_Y}^{-1/2} \mu_{\bar{Y}}) \tag{9}$$

Equation (9) gives the success rate of a DPA attack (SR_{DPA}). It is computed over the cumulative distribution function (Φ_{N_k}) of a multivariate Gaussian distribution, with dimension (N_k) equal to key dimensionality (e.g. 256 for AES if the selection function partitions into 8-bit targets). The number of traces is directly represented in the formulas via N_m (number of measurements). The target device is characterized from the signal to noise ratio ($SNR = \epsilon / \sigma_w$) and the parameters ϵ and σ_w can be computed from side-channel measurements.
Cipher algorithm is isolated by defining and constructing the confusion coefficient κ as given in Eq. (4) and (5). The confusion matrix $\mathbf{K}$ that is subsequently constructed is given in Eq. (6). The matrix elements capture the behavior of the both the cipher and the selection function with respect to a specific key (k_c denotes the correct key and k_g the key guesses that stem from the key space).
The confusion coefficient with respect to a specific S-box was initially defined as the probability that 2 different keys will lead to a different S-box output as given in Eq. (4). Intuitively, a high confusion coefficient indicates that the S-box output (or any other intermediate value ψ targeted by a side-channel attack) is very distinctive. Thus, the S-box output is a good candidate for data leakage. Low confusion coefficient values (also referred to as high collision values) make side-channel attacks harder, i.e. they may require an increase in number of traces or SNR to yield the correct key candidate.

Early work from Fei et al. suggest that the confusion coefficient matrix captures the algorithmic behavior of the cipher [9, 18]. However, this matrix incorporates all possible confusion coefficients with respect to a given key, making the whole analysis key-dependent. In addition, we consider beneficial to move towards CPA-related models instead of DPA. Thus, we use more recent findings from Fei et al., namely the confusion coefficient for CPA, the confusion coefficient vector and its frequency distribution [10]. We compute the confusion coefficients for a given CPA selection functions as shown below.

Having computed all possible confusion coefficient values κ w.r.t. CPA attack and Hamming weight (HW) power model we compute the confusion coefficient vector. This vector contains all possible coefficient values for every key combination and its frequency distribution is deemed by the Fei et al. to be possible characterizer of side-channel behavior. The natural question that arises is what features of the frequency distribution of the confusion coefficient vector denotes side-channel resistance. We observe that the mean value of the distribution is

Algorithm 1 CPA selection function

for all key pairs k_a, k_b, $k_a \neq k_b$ **do**
  for all possible inputs in **do**
    $\kappa(k_a, k_b) = E[(HW(Sbox(in \oplus k_a)) - HW(Sbox(in \oplus k_b))^2]$
  end for
end for

directly related to the choice of the selection function, i.e. it solely depends on the divide-and-conquer approach that we use in our attack. Moreover, Heuser et al. demonstrate the link between nonlinearity and the distribution of the vector [29]. Specifically, highly nonlinear elements lead to a distribution with low variance. Therefore, we need to find S-boxes that demonstrate a high variance in the confusion coefficient vector distribution. Note that our S-boxes are generated under the Hamming weight leakage assumption – depending on the device this assumptions does not always hold.

Two S-boxes S_1 and S_2 are **affine equivalent** only if the following equation holds:

$$S_2(x) = B(S_1(A(x) + a)) + b, \tag{10}$$

where A and B are invertible 4×4 matrices and $a, b \in \mathbb{F}_2^4$ are constant values.

Resistance of S-boxes against most of the attacks remains unchanged if affine transformation is applied before and after S-box [5].

2.3 Optimal S-boxes

When considering 4×4 S-boxes, there exist in total 16! bijective 4×4 S-boxes which is approximately 2^{44} options to search from. Leander and Poschmann define optimal S-boxes as those that are bijective, have linearity equal to 8 and δ-uniformity equal to 4 [5]. Since the linearity of 8 is the same as nonlinearity 4, we continue talking about nonlinearity property instead of linearity.

Furthermore, they found that all optimal S-boxes belong to 16 classes, i.e. all optimal S-boxes are affine equivalent to one of those classes [5].

3 Experimental Setup and Results

When generating S-boxes with good properties, we use genetic algorithms approach as they produced good results in previous works [13, 14]. Additionally, we use random search as a baseline search strategy and affine transformations to check whether confusion coefficient property is affine invariant.

Random Search. In this setting we use Monte Carlo search method to find S-boxes that have good values of confusion coefficient. With this search method we cannot influence the value of any of the S-box properties and we consider it as a baseline search strategy. Here, S-boxes are created uniformly at random.

Genetic Algorithms. In this technique we evolve S-boxes that have good values not only for DPA-related properties, but also for other cryptography relevant properties.

Affine Transformation. As shown in [14] affine transformation affects transparency order values. We employ several different transformations and investigate their influence on confusion coefficient.

Further details about genetic algorithms and affine transformation are given in the following sections.

3.1 Genetic Algorithm

Genetic algorithms (GAs) are a subclass of evolutionary algorithms where the elements of the search space S are arrays of elementary types [30]. Genetic algorithms belong to evolutionary techniques that have been successfully applied to various optimization problems. To be able to evolve new individuals (solutions) GA uses variation operators where the usual ones are mutation and crossover (recombination) operators. Mutation operators are operators that use one parent to create one child by applying randomized changes to parent. Mutation depends on the mutation rate p_m which determines the probability that a change will occur within individual. Recombination operators work on two or more parents to create offspring from the information contained within parent solutions. Recombination is usually applied probabilistically according to a crossover rate p_c. Besides variation operators, it is necessary to decide about selection method. Today, the k-tournament selection method is widely used for this purpose [30]. In this selection k solutions are selected at random and the worst among them is replaced by the offspring of the remaining solutions. Further information about genetic algorithms can be found in [31, 32].

Representation and Fitness Functions. There are several possibilities how to represent S-boxes (e.g. truth tables or lookup tables). We decide to use permutation encoding since in this way the bijectivity property is automatically preserved. In this representation, $n \times m$ S-box is defined with an array of 2^m integer numbers with values between 0 and $2^m - 1$. Each of these values occurs exactly once in an array and represents one entry for the S-box lookup table, where inputs are in lexicographical order.

For a permutation representation, a mutation operator is selected uniformly at random between insert and inversion mutation [31]. For crossover operators we use partially mapped crossover (PMX) [31], position based crossover (PBX) [33] and order crossover (OX) [31] where the operator is selected uniformly at random. All variation operators are described in Appendix C.

Maximization of the value of a fitness function is the objective in all evolutionary experiments. A fitness function represents a definition of the problem to solve with genetic algorithm. For fitness function we use a combination of properties as presented in Section 2.

For the 8×8 case, fitness function equals the sum of nonlinearity (N_F) and confusion coefficient variance (κ) properties values as follows.

$$fitness = N_F + \kappa \tag{11}$$

When investigating 4×4 case, we add to the fitness function δ-uniformity property. In this way we ensure that evolved S-boxes belong to the one of optimal S-boxes classes.

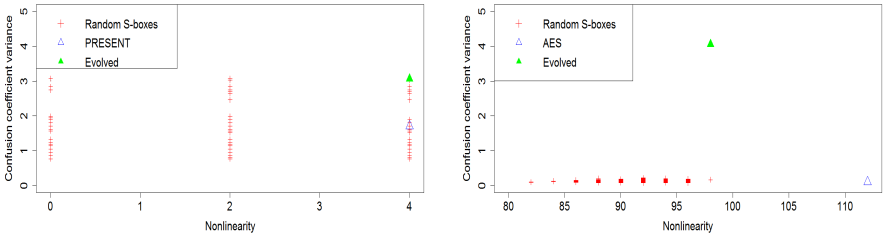
$$fitness = N_F + \kappa + (2^m - \delta) \tag{12}$$

We subtract delta uniformity value from the maximum obtainable value since we represent the problem as a maximization problem and δ property should be as small as possible. Both fitness function can be easily extended to contain more properties that are of relevance to the evolutionary experiments.

Here we emphasize that our approach is not only easily adaptable when adding additional properties, but if we want to change e.g. the leakage model it would only affect one term in the fitness function.

Common Parameters. For every fitness function we run 30 independent runs and population size is equal to 50. Mutation probability is set to 0.3 per individual. The parameters above are the result of a combination of a small number of preliminary experiments and our experience with similar problems; no thorough parameter tuning has been performed. Tournament selection parameter k is set to 3. Evolution process lasts until the stopping criterion is fulfilled, here the criterion is 50 generations without improvement.

In Figures 1(a) and 1(b) we present results for random, evolved and original S-boxes (AES and PRESENT) for sizes 4×4 and 8×8 respectively.



(a) 4×4 size (b) 8×8 size

Fig. 1. Nonlinearity vs. confusion coefficient variance

We see that for 4×4 S-box size we obtain maximum confusion coefficient variance of 3.07 while staying in optimal classes. For the 8×8 size, the best confusion coefficient variance we found is 4.057. However, this value comes at a cost of nonlinearity of 98 and δ-uniformity of 12 (AES has nonlinearity 112 and δ-uniformity 4).

3.2 Affine Equivalence

Recall that resistance of an S-box against most of attacks stays the same if affine transformation is applied before and after S-box. Therefore, it is useful to check whether that is true for confusion coefficient property.

As shown before, transparency order property changes under certain affine transformations [14]. Next, we check what happens with confusion coefficient variance property under affine transformation. We apply transformations as listed in Table 1 to AES S-box as well as to representatives to 16 optimal classes for 4×4 size and PRESENT S-box.

Table 1. Affine transformations.

Number	Transformation
1	$S(x) + c$
2	$S(B(x) + c)$
3	$(A(S(B(x) + c)) + d$
4	$(A(S(B(x) + c) + d)$

In this table $c, d \in \mathbb{F}_2^4$ are constants, $+$ represents XOR operation and A and B are invertible matrices.

Following conclusions apply both for 4×4 and 8×8 S-box sizes.

To change confusion coefficient property, changes 2, 3 and 4 are applicable. Here we note that our experiments show that the transformations 3 and 4 change confusion coefficient more significantly. For instance, the PRESENT S-box has a confusion coefficient variance of 1.709, when applying transformation 2 we succeed in obtaining maximal confusion coefficient variance of 1.803. However, when applying transformations 3 or 4, we obtain maximal confusion coefficient of variance 3.07.

Since affine transformation emerges as a good choice for generating S-boxes with good DPA properties we present result when applying transformation 3 to AES S-box and lexicographical representatives of 16 optimal classes. We opted for transformation 3 since it is one of two transformations that is capable of significantly changing confusion coefficient and we did not observe any significant difference between transformations 3 and 4. For all experiments the procedure consists of applying 25 000 random affine transformations and recording the best obtained results. The best results are presented in Table 2.

We can observe than in the case of 4×4 size, affine transformation reaches same maximum values (although different S-boxes) as genetic algorithms for 8 out of 16 optimal classes. Furthermore, division between classes that reach 3.07 and 3.02 is the same as in the case of optimal S-boxes and PRINCE suitable S-boxes [34]. Classes that reach values 3.07 are those that are not suitable for usage in the PRINCE algorithm. For 8×8 size affine transformation improves confusion coefficient variance only slightly.

Table 2. Results for affine transformation 3

S-box	κ variance
PRESENT	1.709
PRESENT transformation	3.07
G3, G4, G5, G6, G7, G11, G12, G13	3.02
G0, G1, G2, G8, G9, G10, G14, G15	3.07
AES	0.111
AES transformation	0.149

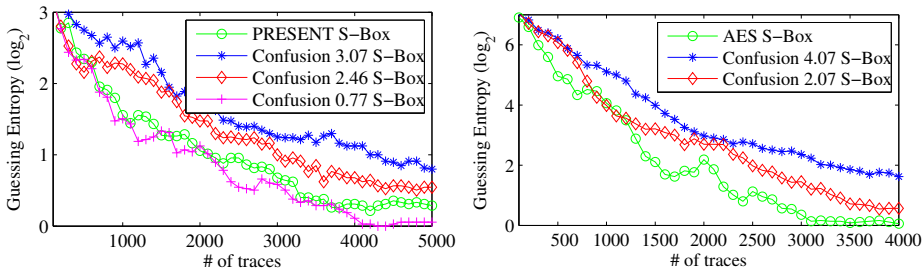
4 Side-Channel Experiments

Although the method explained in the previous sections results in the generation of S-boxes with various values for the confusion coefficient, these S-boxes require a thorough practical analysis. This is required to quantify how much of a change in variance in the confusion coefficient will result in a certain gain in side-channel resistance in terms of the number of measurements required to recover the key.

First, we performed simulations to see how the newly generated S-boxes behave under the Hamming weight model when a certain amount of Gaussian noise is added to the measurements. For the simulated experiments, we used 3 newly generated S-boxes and the PRESENT S-box as the baseline case. One of the 3 newly generated S-boxes is the so called "*Phantom*" S-box that leads to having two key candidates with correlation values equal in magnitude, hence making it more difficult for an attacker who has no knowledge of the exact leakage model of a device to deduce the correct key with 100% accuracy. The "*Phantom*" S-box can be shortly defined as an S-box leading to ghost peaks in the correlation trace after running the attacks. This happens since the S-box outputs have either the same or complementary Hamming weight values for inputs with a particular XOR difference.

Figure 2 presents the logarithm ($\log_2$) of the guessing entropy [35] with respect to the number of traces processed for the attacks we run on the simulated traces we produced with the inclusion of Gaussian noise with mean 0 and standard deviation 16. Important point to note about Figure 2 is that the PRESENT S-box has a confusion coefficient variance of 1.709. Therefore, it can be clearly seen that Figure 2(a) shows a clear distinction in guessing entropy with respect to the variance of the confusion coefficient. Similarly one should note that AES has confusion coefficient variance of 0.11, and Figure 2(b) shows a good distinction in guessing entropy w.r.t. the confusion coefficient variance.

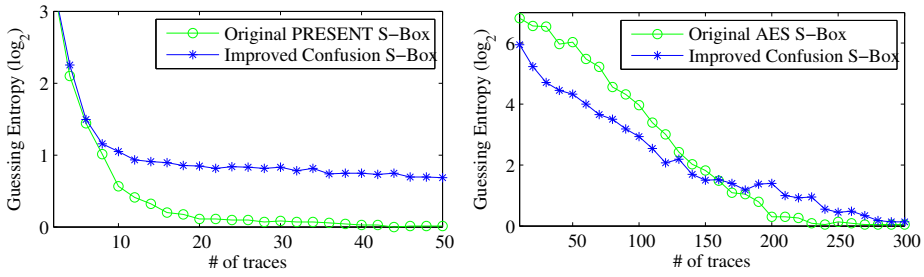
For the practical experiments, we used an ATmega163 microcontroller embedded in a smart card and we collected many measurements using a modified card reader enabling us to produce a trigger signal the oscilloscope and a LeCroy oscilloscope. To be able to make a fair assessment of the side-channel security of different S-boxes, we collected the information from 50 separate attacks and com-



(a) (b)

Fig. 2. Guessing entropy of the simulated S-boxes (4×4 in (a), 8×8 in (b)) with respect to the number of traces processed.

bined them in terms of guessing entropy in Figure 3. Again it is clear that when the attack is applied using the Hamming weight model, the S-box having the better confusion coefficient value shows better resistance against side-channel attacks. Here an important fact to note here is that the "*Phantom*" S-box exhibits this property only when the Hamming weight model is used. The reason for this behavior is that "*Phantom*" S-boxes lead to having either the same Hamming weight, or the exact opposite Hamming weight in the outputs when the inputs have a certain XOR difference in between. Therefore, when one of the bits is taken into account rather than the Hamming weight of the whole S-box output for mounting the attack, this "*Phantom*" behavior may not necessarily persist. However, if the target leaks the Hamming weight of intermediate values, then the attacker would be forced to use a weaker selection function (bit model) for that particular device, therefore leading to an attack requiring the acquisition of more power traces.



(a) (b)

Fig. 3. Guessing entropy of the S-boxes (4×4 in (a), 8×8 in (b)) with respect to the number of traces processed on the AVR microcontroller.

It should be noted that the results presented in this paper assume a particular leakage model, namely the Hamming weight model. We have computed the confusion coefficient with this assumption in mind but if the leakage of a device is known, it is straightforward to integrate that leakage model in our genetic algorithms and produce an S-box which will resists to side-channel attacks better in a device leaking in that particular leakage model.

Although we observe that improving the confusion coefficient results in designs which have better side-channel resistance, we do recognize that this cannot be counted as a countermeasure. We believe it is interesting to investigate how an improved S-box interacts with other countermeasures and especially with masking. Since in this work we focus on the practicality of the confusion coefficient metric, it remains as an interesting open question to see whether the S-box improvements are persistent after masking or not.

5 Conclusion

In this work we consider the DPA resistance properties of S-boxes of various sizes. We show it is possible to evolve S-boxes that have better confusion coefficient variance values. Using genetic algorithms we are able to produce both 4×4 and 8×8 size S-boxes that exhibit improved DPA resistance.

Next, we show that an affine transformation changes the confusion coefficient variance property. This fact can be important not only from the theoretical perspective, but also from the practical one. We reiterate that with the genetic algorithms approach change in the leakage model leads only to the change in one fitness function factor. Therefore, we can easily adapt the procedure to other more generic scenarios.

In further work we will concentrate on the interaction between the improved S-boxes and masking countermeasure.

References

1. M. Matsui and A. Yamagishi, "A new method for known plaintext attack of FEAL cipher," in *Proceedings of*, ser. EUROCRYPT'92. Berlin, Heidelberg: Springer-Verlag, 1993, pp. 81–91.
2. E. Biham and A. Shamir, "Differential Cryptanalysis of DES-like Cryptosystems," in *Proceedings of*, ser. CRYPTO '90. London, UK, UK: Springer-Verlag, 1991, pp. 2–21.
3. C. Carlet, "On highly nonlinear S-boxes and their inability to thwart DPA attacks," in *Proceedings of*, ser. INDOCRYPT'05. Berlin, Heidelberg: Springer-Verlag, 2005, pp. 49–62.
4. S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards (Advances in Information Security)*. Secaucus, NJ, US1A: Springer-Verlag New York, Inc., 2007.
5. G. Leander and A. Poschmann, "On the Classification of 4 Bit S-Boxes," in *Arithmetic of Finite Fields*, ser. Lecture Notes in Computer Science, C. Carlet and B. Sunar, Eds. Springer Berlin Heidelberg, 2007, vol. 4547, pp. 159–176.

6. S. Guilley and R. Pacalet, "Differential Power Analysis Model and Some Results," in *In proceedings of CARDIS 2004*. Kluwer Academic Publishers, 2004, pp. 127–142.
7. E. Prouff, "DPA Attacks and S-Boxes," in *12th International Workshop, FSE 2005, Paris, France*, ser. Lecture Notes in Computer Science, vol. 3557. Springer, 2005, pp. 424–441.
8. S. Guilley, P. Hoogvorst, R. Pacalet, and J. Schmidt, "Improving Side-Channel Attacks by Exploiting Substitution Boxes Properties." in *International Workshop on Boolean Functions : Cryptography and Applications*, ser. BFCA '14, 2007, pp. 1–25.
9. Y. Fei, Q. Luo, and A. A. Ding, "A statistical model for dpa with novel algorithmic confusion analysis," in *CHES*, 2012, pp. 233–250.
10. Y. Fei, A. A. Ding, J. Lao, and L. Zhang, "A statistics-based fundamental model for side-channel attack analysis," *IACR Cryptology ePrint Archive*, vol. 2014, p. 152, 2014.
11. B. Mazumdar, D. Mukhopadhyay, and I. Sengupta, "Design and implementation of rotation symmetric S-boxes with high nonlinearity and high DPA resilience," in *Hardware-Oriented Security and Trust (HOST), 2013 IEEE International Symposium on*, 2013, pp. 87–92.
12. B. Mazumdar, D. Mukhopadhay, and I. Sengupta, "Constrained Search for a Class of Good Bijective S-Boxes with Improved DPA Resistivity," *Information Forensics and Security, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2013.
13. S. Picek, B. Ege, L. Batina, D. Jakobovic, L. Chmielewski, and M. Golub, "On Using Genetic Algorithms for Intrinsic Side-channel Resistance: The Case of AES S-box," in *Proceedings of the First Workshop on Cryptography and Security in Computing Systems*, ser. CS2 '14. New York, NY, USA: ACM, 2014, pp. 13–18.
14. S. Picek, B. Ege, L. Batina, D. Jakobovic, and K. Papagiannopoulos, "Optimality and beyond: The case of 44 s-boxes," in *Hardware-Oriented Security and Trust (HOST), 2014 IEEE International Symposium on*, 2014, to appear.
15. K. Chakraborty, S. Maitra, S. Sarkar, B. Mazumdar, and D. Mukhopadhyay, "Redefining the transparency order," Cryptology ePrint Archive, Report 2014/367, 2014, http://eprint.iacr.org/.
16. D. Canright, "A very compact s-box for aes," in *CHES*, 2005, pp. 441–455.
17. D. Canright and L. Batina, "A very compact "perfectly masked" s-box for aes," in *ACNS*, 2008, pp. 446–459.
18. Q. Luo and Y. Fei, "Algorithmic collision analysis for evaluating cryptographic systems and side-channel attacks," in *HOST*, 2011, pp. 75–80.
19. A. A. Ding, L. Zhang, Y. Fei, and P. Luo, "A statistical model for higher order dpa on masked devices," *IACR Cryptology ePrint Archive*, vol. 2014, p. 433, 2014.
20. A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. Robshaw, Y. Seurin, and C. Vikkelsoe, "PRESENT: An Ultra-Lightweight Block Cipher," in *Proceedings of*, ser. CHES '07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 450–466.
21. J. Daemen and V. Rijmen, *The Design of Rijndael*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2002.
22. P. Kocher, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS and other systems," in *Advances in Cryptology - CRYPTO'96*, ser. Lecture Notes in Computer Science, N. Koblitz, Ed., no. 1109. Springer-Verlag, 1996, pp. 104–113.
23. P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," in *Advances in Cryptology - CRYPTO'99*, ser. Lecture Notes in Computer Science, M. Wiener, Ed., no. 1666. Springer-Verlag, 1999, pp. 388–397.

24. J.-J. Quisquater and D. Samyde, "ElectroMagnetic Analysis (EMA): Measures and Counter-Measures for Smard Cards," in *Smart Card Programming and Security (E-smart 2001)*, ser. Lecture Notes in Computer Science, I. Attali and T. P. Jensen, Eds., vol. 2140. Springer-Verlag, 2001, pp. 200–210.
25. D. Genkin, A. Shamir, and E. Tromer, "Rsa key extraction via low-bandwidth acoustic cryptanalysis," Cryptology ePrint Archive, Report 2013/857, 2013, http://eprint.iacr.org/.
26. Y. Crama and P. L. Hammer, *Boolean Models and Methods in Mathematics, Computer Science, and Engineering*, 1st ed. New York, NY, USA: Cambridge University Press, 2010.
27. A. Braeken, "Cryptographic Properties of Boolean Functions and S-Boxes," Ph.D. dissertation, Katholieke Universiteit Leuven, 2006.
28. K. Nyberg, "Perfect Nonlinear S-Boxes," in *Advances in Cryptology - EUROCRYPT '91, Workshop on the Theory and Application of of Cryptographic Techniques, Brighton, UK, April 8-11, 1991, Proceedings*, ser. Lecture Notes in Computer Science, vol. 547. Springer, 1991, pp. 378–386.
29. A. Heuser, S. Guilley, and O. Rioul, "A theoretical study of kolmogorov-smirnov distinguishers: Side-channel analysis vs. differential cryptanalysis," *IACR Cryptology ePrint Archive*, vol. 2014, p. 8, 2014.
30. T. Weise, *Global Optimization Algorithms Theory and Application*, 2009.
31. A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*. Springer-Verlag, Berlin Heidelberg New York, USA, 2003.
32. Z. Michalewicz, *Genetic algorithms + data structures = evolution programs (3rd ed.)*. London, UK, UK: Springer-Verlag, 1996.
33. G. Syswerda, "Schedule optimization using genetic algorithms," in *Handbook of Genetic Algorithms*, 1991, pp. 332–349.
34. J. Borghoff, A. Canteaut, T. Gneysu, E. Kavun, M. Knezevic, L. Knudsen, G. Leander, V. Nikov, C. Paar, C. Rechberger, P. Rombouts, S. Thomsen, and T. Yaln, "PRINCE : A Low-Latency Block Cipher for Pervasive Computing Applications," in *Advances in Cryptology: ASIACRYPT 2012*, ser. Lecture Notes in Computer Science, X. Wang and K. Sako, Eds. Springer Berlin Heidelberg, 2012, vol. 7658, pp. 208–225.
35. F.-X. Standaert, T. G. Malkin, and M. Yung, "A unified framework for the analysis of side-channel key recovery attacks," in *Proceedings of the 28th Annual International Conference on Advances in Cryptology: The Theory and Applications of Cryptographic Techniques*, ser. EUROCRYPT '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 443–461.
36. Z. Gong, S. Nikova, and Y. Law, "Klein: A new family of lightweight block ciphers," in *RFID. Security and Privacy*, ser. Lecture Notes in Computer Science, A. Juels and C. Paar, Eds. Springer Berlin Heidelberg, 2012, vol. 7055, pp. 1–18.
37. C. Canniere, H. Sato, and D. Watanabe, "Hash function Luffa: Specification 2.0.1." Submission to NIST (Round 2), 2009, http://www.sdl.hitachi.co.jp/crypto/luffa/.
38. J. Daemen, M. Peeters, G. V. Assche, and V. Rijmen, "Nessie proposal: the block cipher NOEKEON," Nessie submission, 2000, http://gro.noekeon.org/.

A Comparison of S-Boxes Used in Lightweight Block Ciphers

We are interested only in S-box properties and not the strength or the quality of the algorithm as a whole. Therefore, we select a set of examples that we believe

are interesting for comparison. Here we compare PRESENT [20], PRINCE [34], Klein [36], Luffa [37] and NOEKEON [38] S-boxes. Table 3 presents results for two important criteria for an S-box to be optimal, as well as two properties related to DPA resistance.

Table 3. S-box Properties of Some Well-known 4×4 Algorithms

Algorithm	NL	δ	T_F	$SNR(DPA)$	κ variance
PRINCE	4	4	3.4	2.129	1.709
PRESENT	4	4	3.533	2.129	1.709
NOEKEON	4	4	3.533	2.187	1.615
Klein	4	4	3.467	1.691	2.742
Luffa	4	4	3.733	2.53	1.191

B Evolved S-boxes

An example of optimal S-box of size 4×4 and confusion coefficient variance is:

S-box = (0x6, 0x4, 0x7, 0x8, 0x0, 0x5, 0x2, 0xA, 0xE, 0x3, 0xD, 0x1, 0xC, 0xF, 0x9, 0xB)

Evolved 8×8 S-box with variance of 4.057 and nonlinearity 98 is given below.

S-box = (0xb1, 0x23, 0x98, 0x27, 0x4b, 0x14, 0x9, 0x5c, 0x55, 0xa, 0x4a, 0x4c, 0x1b, 0x3a, 0xa2, 0x53, 0xd6, 0xfb, 0x9f, 0x5e, 0xae, 0xde, 0xe7, 0 x9e, 0x4f, 0x97, 0xf7, 0x2d, 0x2e, 0xbe, 0xab, 0x2b, 0x91, 0x87, 0x36 , 0x1c, 0x81, 0x9d, 0xe5, 0x1a, 0xac, 0x1e, 0x5b, 0x86, 0x8c, 0x74, 0x6a, 0x8a, 0x5f, 0x65, 0xd5, 0x3f, 0xfe, 0xd9, 0xf, 0x37, 0xdd, 0x7d , 0xf2, 0xec, 0xf6, 0xe2, 0xb3, 0xaf, 0x77, 0x99, 0xca, 0xb9, 0xbb, 0xd0, 0x6c, 0xa7, 0x3d, 0xcb, 0x17, 0x75, 0x76, 0x4d, 0xad, 0xcf, 0x5 0, 0x68, 0x16, 0x2, 0x12, 0x78, 0x56, 0x1, 0xb0, 0x71, 0x5a, 0x29, 0 x6, 0x69, 0x58, 0x88, 0x8b, 0x6b, 0xe9, 0x8e, 0xc1, 0xc7, 0x6e, 0x63, 0x13, 0xbc, 0x2f, 0x38, 0x96, 0xbd, 0xdc, 0x62, 0xa8, 0x82, 0x24, 0 xa1, 0xb8, 0x0, 0x80, 0x61, 0xcc, 0x83, 0x22, 0x2c, 0xc2, 0xc0, 0xa0, 0x90, 0xf0, 0xdf, 0xdb, 0xba, 0xe8, 0xf9, 0xbf, 0x7c, 0x59, 0x7b, 0 xeb, 0xd8, 0xa3, 0xff, 0xf3, 0xf8, 0xc8, 0x5, 0x64, 0x66, 0xaa, 0xa9, 0xe, 0xb2, 0xd2, 0x19, 0x10, 0x70, 0x45, 0xc, 0x2a, 0x79, 0x3e, 0x5 d, 0x6d, 0xfa, 0xed, 0xda, 0xe1, 0x9a, 0x7f, 0x4e, 0x8d, 0xf5, 0xfc, 0x7a, 0x57, 0xfd, 0xd, 0xe4, 0x95, 0x18, 0xb4, 0xb5, 0x1d, 0x26, 0x4 8, 0x93, 0x67, 0x7, 0x51, 0xd4, 0x34, 0x43, 0x84, 0x9b, 0x92, 0x60, 0x28, 0x49, 0xc6, 0xc4, 0x8, 0x54, 0xa5, 0x41, 0x40, 0xea, 0xa4, 0x44 , 0x35, 0x15, 0x3b, 0xce, 0xf4, 0xd3, 0x33, 0xb6, 0x8f, 0xcd, 0x25, 0xef, 0xb7, 0x3c, 0x46, 0xee, 0x85, 0x32, 0x3, 0xc3, 0x31, 0xb, 0x30, 0x72, 0xd1, 0x20, 0x4, 0xa6, 0xc9, 0x21, 0x89, 0x47, 0x52, 0x7e, 0x 6f, 0x11, 0xc5, 0xf1, 0xd7, 0x39, 0x94, 0x1f, 0xe3, 0x9c, 0xe0, 0x73, 0xe6, 0x42)

Affine transformation of AES S-box with improved confusion coefficient variance of 0.149357 is given next.

S-box = (0x92, 0x21, 0xd1, 0x6c, 0x5c, 0xf2, 0xf5, 0x86, 0xdd, 0x43, 0x8a, 0x2 8, 0xb8, 0xa3, 0x8b, 0xcf, 0x12, 0xca, 0x23, 0x37, 0xa6, 0xb7, 0x3b, 0xc0, 0x20, 0x4b, 0x5b, 0x22, 0xa4, 0x8, 0x96, 0xff, 0xb2, 0x56, 0xe 9, 0xcd, 0x17, 0x13, 0x57, 0x76, 0x19, 0x18, 0x1d, 0x25, 0xa0, 0x70, 0xec, 0x26, 0xef, 0x1e, 0x8e, 0x29, 0x39, 0x78, 0x6b, 0x4d, 0x60, 0x 95, 0x44, 0xfa, 0xab, 0xcb, 0xc8, 0xe, 0xae, 0xf4, 0x79, 0x46, 0xc, 0x85, 0x7c, 0xbf, 0x40, 0x81, 0xd7, 0x3a, 0xf3, 0xbd, 0x2b, 0x27, 0x5 5, 0x90, 0x61, 0x10, 0xde, 0x82, 0xb6, 0xe0, 0x72, 0x4e, 0x35, 0xea, 0x8c, 0xac, 0x77, 0x52, 0xd5, 0x88, 0xdf, 0x64, 0xc1, 0x65, 0x42, 0x 9c, 0x16, 0x15, 0x33, 0x0, 0x7d, 0x4f, 0x98, 0x9f, 0x45, 0xa2, 0x67, 0x69, 0xd6, 0xcc, 0xd0, 0x5e, 0xbb, 0x73, 0x87, 0x6d, 0x74, 0x3f, 0x ad, 0x7, 0x50, 0x7f, 0x4a, 0x1b, 0x68, 0x71, 0xe2, 0x2, 0x3d, 0xc2, 0x38, 0xba, 0xd9, 0xa, 0x32, 0x31, 0x97, 0xda, 0x99, 0x8f, 0xd8, 0x9d , 0xd3, 0x2e, 0x2d, 0x5d, 0xc4, 0x54, 0x58, 0x91, 0x6, 0x6e, 0x51, 0 x3c, 0x6f, 0xfd, 0xf6, 0xf, 0x48, 0x34, 0x4, 0xf7, 0xc6, 0xa7, 0xf1, 0xaa, 0x47, 0x5a, 0x3e, 0x66, 0xdc, 0x6a, 0x3, 0xb3, 0x63, 0xfc, 0x1 a, 0x49, 0x2c, 0xf0, 0xce, 0x36, 0x7e, 0xe7, 0xe5, 0xb5, 0xa1, 0x7a, 0xc9, 0xee, 0x1c, 0xa5, 0x7b, 0xd4, 0x9b, 0x41, 0xd, 0xa8, 0x5, 0x84 , 0xb1, 0x93, 0x2f, 0xbe, 0xc5, 0xb, 0xeb, 0xe1, 0xaf, 0x9a, 0x80, 0 x8d, 0x4c, 0xe4, 0xfb, 0x9e, 0x89, 0x24, 0x2a, 0x83, 0x9, 0x94, 0x53, 0xbc, 0x5f, 0xa9, 0xc7, 0x75, 0xb0, 0x30, 0x1f, 0xb4, 0xdb, 0xf8, 0 xc3, 0xb9, 0xd2, 0xfe, 0x11, 0xed, 0x59, 0xf9, 0xe8, 0x1, 0xe3, 0xe6, 0x62, 0x14)

C Variation Operators

PMX Crossover. First, two crossover positions are chosen randomly, and the segment between them from the first parent is copied to the offspring. Then, starting from the first crossover position check elements in that segment of second parent that have not been copied. For each of those elements i, check the offspring to see what elements j has been copied in its place from first parent.Place those values i into the positions occupied j in parent 2. If the place occupied by j in parent 2 has already been occupied in the offspring by an element k, put i in the position occupied by k in parent 2. After all the elements in crossover segment are finished, the rest of the offspring is filled from parent 2 [31].

PBX Crossover. In this operator first the values in random positions from the first parent are copied to the same positions in the offspring. Next, values from the second parent that are not present in the offspring are copied to it starting from the beginning of the offspring [31].

OX Crossover. Two crossover positions are chosen at random, and the segment between those positions is copied from the first parent to the offspring. Starting from the second crossover point in the second parent, copy unused values to the offspring in the order they appear in the second parent, wrapping around at the end of the list [31].

Inversion Mutation. In this operator, first two positions are chosen at random. Then, the segment between those 2 values are written in reverse order [31].

Insert Mutation. In this operator two positions are selected at random and then the value from one of those position is moved to be next to the other position. Values in the segment between are shuffled to make room for value to be moved [31].