On Using Genetic Algorithms for Intrinsic Side-Channel Resistance: The Case of AES S-Box

Stjepan Picek
Digital Security Group - ICIS
Radboud University Nijmegen
The Netherlands
stjepan@computer.org

Barış Ege
Digital Security Group - ICIS
Radboud University Nijmegen
The Netherlands
B.Ege@cs.ru.nl

Lejla Batina
Digital Security Group - ICIS
Radboud University Nijmegen
The Netherlands
lejla@cs.ru.nl

Domagoj Jakobovic
Faculty of Electrical
Engineering and Computing
University of Zagreb
Croatia
domagoj.jakobovic@fer.hr

Łukasz Chmielewski
Riscure BV
Delft, The Netherlands
Chmielewski@riscure.com

Marin Golub
Faculty of Electrical
Engineering and Computing
University of Zagreb
Croatia
marin.golub@fer.hr

ABSTRACT

Finding balanced S-boxes with high nonlinearity and low transparency order is a difficult problem. The property of transparency order is important since it specifies the resilience of an S-box against differential power analysis. Better values for transparency order and hence improved side-channel security often imply less in terms of nonlinearity. Therefore, it is impossible to find an S-box with all optimal values. Currently, there are no algebraic procedures that can give the preferred and complete set of properties for an S-box. In this paper, we employ evolutionary algorithms to find S-boxes with desired cryptographic properties. Specifically, we conduct experiments for the 8×8 S-box case as used in the AES standard. The results of our experiments proved the feasibility of finding S-boxes with the desired properties in the case of AES. In addition, we show preliminary results of side-channel experiments on different versions of "improved" S-boxes.

Categories and Subject Descriptors

D.4 [**Operating Systems**]: Miscellaneous; D.4.6 [**Software Engineering**]: Security and Protection—*Cryptographic controls*

General Terms

Security, Experimentation

Keywords

Block Ciphers, S-Box, Transparency Order, Genetic Algorithms, Side-channel Analysis

CS2, January 20 2014, Vienna, Austria
http://dx.doi.org/10.1145/2556315.2556319

1. INTRODUCTION

Block cipher algorithms are vulnerable to various kinds of cryptanalysis. Besides more traditional linear [1] and differential cryptanalysis [2], the most popular attacks today belong to side-channel analysis (SCA) targeting actual implementations of cryptography in software or hardware. SCA relies on the physical leakages from the actual implementation and its efficiency is much greater than the one of linear or differential cryptanalysis [3]. Through various countermeasures such as numerous masking and hiding schemes [4] it is possible to make the algorithm more resilient to SCA. However, this comes with a substantial cost increase due to the increase of memory requirements and the decrease of performance of the algorithm implemented.

In the design process of block ciphers one usually follows principles of diffusion and confusion as introduced by Shannon [5]. The amount of confusion in an algorithm is measured with the nonlinearity property of nonlinear parts of algorithm e.g. Boolean functions and S-boxes. Considering side-channel security, Prouff [6] defines the transparency order property that characterizes the resistance of S-boxes to the SCA or more precisely to differential power analysis (DPA) [7]. However, there are still no algebraic methods available to design S-boxes that have good transparency order and adequately high nonlinearity. Since the worst transparency order is obtained in the case when bent functions are used (as bent functions obtain maximal nonlinearity [8]) it implies the fact that nonlinearity and transparency order are conflicting criteria.

When random generation is used for S-boxes, the resulting S-boxes have reduced nonlinearity when compared to many specially constructed S-boxes. Therefore, random generation does not present a viable choice in the generation of S-boxes with good transparency order.

In this paper we follow this intuition but we also do better in terms of making good trade-offs among all the properties. We use evolutionary computation techniques to evolve S-boxes with good transparency order and acceptable nonlinearity values. More precisely, this work makes the first step in using this powerful method on a very practical cryptographic problem. We aim at finding better alternatives for S-boxes, as used in block ciphers or other symmetric cryp-

tographic primitives, in terms of improving their resistance against side-channel analysis without too much deteriorating the security of S-boxes. Our general goal is to come up with an evolutionary computation framework for finding "proper" S-boxes that is both, effective and efficient. Naturally, a design method that does not favour special methods (e.g. algebraic based) also has several downsides. The most obvious bottleneck is the inability to store S-box in a format different from a lookup table. From that perspective it is unlikely that the new S-boxes can be used in every environment. However, on platforms with sufficient area for the lookup tables and where resilience against SCA is of great importance, we are confident that our method can be a viable alternative. In this case, as table lookups are also susceptible to cache attacks, a cache-timing resistant lookup table could be used e.g. as presented by Bernstein [9].

1.1 Related Work

We divide relevant work in two categories: first one concerning previous usages of evolutionary computation in evolving S-boxes and second one covering results concerning transparency order property.

Evolving S-boxes. There are many successful applications of evolutionary computation when evolving Boolean functions or S-boxes. Relevant works include the papers of Millan et al. [10] and Jacob et al. [11].

Transparency order. Mazumdar et al. construct rotation symmetric S-boxes with high nonlinearity and DPA resistance [14]. Furthermore, they employ those S-boxes in several hardware implementations and show that their S-boxes have better DPA resistance than the AES S-box. Same authors also use constrained random search to find S-boxes with better than AES transparency order [15].

Our work is the first one to use the techniques of evolutionary computation in finding cryptographically strong S-boxes that feature also improved side-channel resilience. More details on our contributions are given below.

1.2 Our Contribution

When using evolutionary computation techniques a special caution is required as evolutionary algorithms are not magic-solvers for any kind of problem. They can help in finding viable solutions but to have something feasible or in this case suitable for real-life applications, all the conditions have to be taken into account and treated specifically. Wolpert and Macready introduce the "No Free Lunch" theorem and prove that there is no single best algorithm for every problem [16]. Of course, this theorem is only applicable when we possess no knowledge about the problem at hand. With a careful choice of evolutionary computation technique and with adequate settings, evolutionary computation can be used to solve various real-world problems.

Here we need to reiterate that evolutionary computation should not be regarded as the best possible method for solving this problem (or any problem).

In this paper, we use evolutionary computation technique, specifically genetic algorithm, to evolve S-boxes with low transparency order and relatively high nonlinearity values. To be able to do that, we experiment with several versions of evolutionary computation techniques to find the best one. Also we present simple, yet effective fitness function we use to find new S-boxes. The experiments prove that evolutionary algorithms are a viable option in evolving S-boxes with low transparency order and high nonlinearity. In addition, we show the results of practical experiments that confirm our findings. For this purpose, we use power consumption traces derived from a programmable smart card on which our new improved S-boxes are implemented. More precisely, we conduct two different types of the experiments. The first type are experiments to evolve S-boxes, and the second type are the experiments to evaluate the resistance of evolved S-boxes to DPA attacks. To avoid the confusion, for the former experiments we use the name evolutionary experiments and for the latter side-channel experiments.

The remainder of this paper is organized as follows: In Sect. 2 we survey necessary information about evolutionary computation and cryptographic properties of S-boxes. In Sect. 3 our evolutionary computation experimental setup and the results are presented. Sect. 4 contains a discussion about the implementation of evolved S-boxes and our first results from side-channel analysis. Finally, in Sect. 5 we conclude the paper.

2. PRELIMINARIES

Here we give necessary information about side-channel analysis, cryptographic properties of S-boxes and evolutionary computation.

2.1 Side-channel Analysis and DPA

Small cryptographic devices, such as smart cards, RFID tags etc. have become pervasive in our lives and lots of our security and privacy-sensitive data is stored on those constrained platforms. These devices provide unintentional output channels, often called side channels. Sometimes, these types of information leakages may be linked either to the types of operations that the cryptographic algorithm is performing, or to the data, i.e., the keys being processed. This makes the leakages explorable by the adversary trying to extract the secret key as she is always looking for shortcuts in cryptanalysis. Considering the physical information explored there are several side channels possible. The best known and most commonly used side-channel is power consumption. as introduced in the first academic publications by Kocher et al. [7, 17]. Different sources of side-channel data, such as electromagnetic emanation [18], timing [17], sound, and temperature have been used for successful side-channel attacks (for a general overview see e.g. [4]).

2.2 Cryptographic Properties of S-boxes

Here we present the properties that are used in evaluation of S-boxes by evolutionary algorithms. Other relevant cryptographic properties are calculated a posteriori and presented in Sect. 3.3.

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 $\bar{a} \cdot \bar{b} = \oplus_{i=1}^{n} a_i b_i$.

An (n, m)-function is any mapping F from $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$ [6]. Such a function F is called S-box or vectorial Boolean function. If m equals 1 then the function is called Boolean function. Boolean functions f_i, where $i \in \{1, ..., m\}$ are coordinate functions of F and 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 [19]. Balanced (n, n)-functions are permutations on $\mathbb{F}_2^n$.

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

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

Here, $W_F(\bar{a}, \bar{v})$ represents Walsh transform of F [6].

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

In 2005, Prouff introduced a new cryptographic property of S-boxes: transparency order [6] which can be defined for a (n,m)-function as follows.

$$T_F = max_{\bar{\beta} \in \mathbb{F}_2^m} (|m - 2HW(\bar{\beta})| - \frac{1}{2^{2n} - 2^n} \sum_{\bar{a} \in \mathbb{F}_2^{n*}} | \sum_{\substack{\bar{v} \in \mathbb{F}_2^m \\ HW(\bar{v}) = 1}} (-1)^{\bar{v} \cdot \bar{\beta}} W_{D_a F}(\bar{0}, \bar{v})|). \quad (3)$$

Here, $W_{D_a F}$ represents Walsh transform of the derivative of F with respect to a vector $a \in \mathbb{F}_2^n$.

This property is unlike the ones known up to that time (with the exception of SNR (DPA) (F) property [20]) since it is related with the resistance of the S-boxes to the DPA attacks. According to Prouff, transparency order has an upper bound of m for an (n,m)-function. This bound is achieved if every coordinate function f_i is bent function. In the case F is an affine function, then the transparency order is zero. The higher the transparency order value is, the lower is the S-box resistance to the DPA attacks [6]. Since bent functions have maximum nonlinearity, we can see that high nonlinearity and low transparency order are conflicting criteria. Carlet also showed that some S-boxes with very high nonlinearity have very bad transparency orders [3].

2.3 Genetic Algorithms

Genetic algorithms (GAs) are a subclass of evolutionary algorithms where the elements of the search space S are arrays of elementary types [21]. Today, genetic algorithms represent evolutionary technique that has been successfully applied to various optimization problems. To be able to produce 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 randomised 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 [21].

3. EXPERIMENTAL SETTINGS AND RESULTS

In all our evolutionary experiments we use the genetic algorithm as presented in [22]. For evolutionary algorithms test suite we use the Evolutionary Computation Framework (ECF) [**?**]. ECF is a C++ framework intended for the application of any type of the evolutionary computation, developed at the University of Zagreb.

The goal is to evolve balanced bijective S-boxes with high nonlinearity and low transparency order. We experiment with the 8×8 size S-box as this is the size of AES S-box which represents the standard for block ciphers.

3.1 Fitness Function and Representation

Maximization of the value of a fitness function is the objective in all evolutionary experiments. Fitness function represents definition of the problem to solve with evolutionary algorithm. For fitness function we use a combination of balancedness, nonlinearity and transparency order properties. Since we require that the solutions are balanced, we do not add balancedness to the fitness function. Rather, we set it as a constraint that needs to be fulfilled to evaluate the fitness value of an individual.

Our fitness function equals the sum of nonlinearity (N_F) and transparency order (T_F) properties values. Since the transparency order value should be as low as possible, we subtract the value obtained from the upper bound value for transparency order.

This fitness function can be easily extended to contain more properties that are of relevance to the evolutionary experiments.

$$fitness = N_F + (m - T_F) \quad (4)$$

We also experimented with the weighted fitness formula but the results were similar. This is due to the fact that algorithm finds some nonlinearity level and while remaining at the same level looks for the best transparency order value. If we add more weight to the transparency order than it artificially adds the importance of transparency order for lower values of nonlinearity but still achieves same transparency order values for each nonlinearity level.

In the genetic algorithm we use permutation representation of solutions. In the permutation representation, 8×8 S-box is defined with an array of 256 integer numbers with values between 0 and 255 (256 distinct values). Each of those values occurs exactly once in an array and represents one entry for the S-box lookup table, where inputs are in lexicographical order. The optimization problem is finding an adequate ordering of those values to achieve the desired properties. When using permutation representation, the problem of finding good S-boxes can be informally treated as a special instance of traveling salesman problem (TSP) [23, 24]. In TSP the objective is to find the optimal path between all the cities in the map (or more generally, objective is to decide on the order of values). Here, we wanted to find the optimal path between values in S-box lookup tables. When regarded as a TSP, we can conclude the problem is hard since there is $256! - 2$ possible solutions (we neglect solutions where the output of the lookup table is the same as the input and where AES S-box is a solution).

3.2 Evolutionary Process and Parameters

Once the parameters of GA are set, we can start with the generation of the initial population. Solutions in initial population are created by randomly setting each value from 0 to 255 as outputs of a lookup table. When the initial population is generated, genetic algorithm starts with

the evolution process. In each iteration it randomly chooses k possible solutions (the tournament of size k) and selects the worst solution among those (this selection method also ensures elitism i.e. the best solutions are always propagated to the next generation). The remaining solutions are used as parents which create one offspring via variation operators. The offspring (new solution) then replaces the worst solution in the tournament.

For variation operators we use 3 mutation operators and 3 crossover operators (we chose the operators that are among the most common ones in use today). We use insert mutation [22], inversion mutation [22] and swap mutation [**?**]. For crossover operators we use partially mapped crossover (PMX) [**?**], position based crossover (PBX) [**?**] and order crossover (OX) [**?**]. For each offspring, an operator is selected uniformly at random between all operators within a class of operators (mutation and crossover). Further informations about variation operators can be found in [22].

The evolution process repeats until the stopping criterion is met; here the stopping criterion is a certain number of generations without improvement of the best solution.

Parameters for the evolutionary algorithm are following: the size of (n, m)-function is 8×8, number of independent runs for each evolutionary experiment is 30 and the population size is 100. Tournament size in steady-state tournament selection is equal to 3. Mutation probability is set to 0.3 per individual. This mutation rate is chosen on a basis of a small set of tuning experiments where it showed the best results on average.

3.3 Genetic Algorithm Results

Several examples of S-boxes are given in Table 1. First two S-boxes should be regarded as benchmarks, since first one is the AES S-box and second one is a randomly created S-box.

S-boxes 1 to 5 are examples of evolved S-boxes. Additionally, we give values for the following cryptographic properties: DPA signal-to-noise ratio (SNR) [20], global avalanche criterion (GAC) - absolute indicator (Δ_F) and sum-of-square indicator (σ_F) [25, 26] and differential δ-uniformity (δ - uniformity) [19, 27]. Here, GAC and δ-uniformity represent the properties related to the resistance of algorithm to the linear and differential cryptanalysis, and DPA (SNR) relates with the DPA resistance of S-boxes.

Table 1: Cryptographic Properties of S-boxes

S-box	N_F	T_F	SNR	δ-unif.	Δ_F	σ_F
AES S-box	112	7.86	9.599	4	32	133120
Random S-box	92	7.805	10.001	12	96	257152
S-box 1	100	7.717	8.686	10	104	245632
S-box 2	98	7.358	5.825	12	104	341248
S-box 3	98	7.41	6.034	14	112	370816
S-box 4	100	7.53	5.44	12	104	298624
S-box 5	98	7.50	6.547	14	112	356224

All the S-boxes enumerated in the table are balanced so we did not write that property in the table. Also, all the S-boxes have algebraic degree equal to 7. We omitted correlation immunity property from the table since it must be 0 as evident by Siegenthaler's inequality [8]. Further, none of the S-boxes satisfy SAC property so we also omitted it from the table [8].

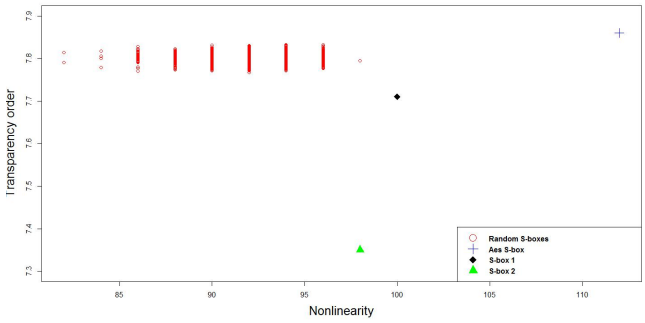

Figure 1: Nonlinearity versus transparency order for S-boxes

In Fig. 1 we displayed comparison between random search results and GA results. Circles represent one million random S-boxes results, the plus symbol represents AES S-box, the diamond symbol represents evolved S-box 1, and finally, the triangle symbol represents evolved S-box 2. Here we note that random search found S-box with nonlinearity of 98 and transparency order of 7.78 which is far worse than the GA results.

As evident from Table 1 and Fig. 1, finding S-boxes with low transparency order and high nonlinearity is hard. Low nonlinearity value does not ensures low transparency order. In fact, it is easy to find S-boxes with nonlinearity below 90 and with transparency order comparable to that of AES S-box. Since we could not find any S-box with nonlinearity level the same as in AES case and with significantly better transparency order, we opted to find S-boxes with nonlinearity lower than in AES, but also with transparency order significantly lower than in AES case. Here, by significantly better transparency order values we mean those S-boxes where we require more traces to perform a DPA attack. Since the evolved S-boxes must be implemented through lookup tables while they have lower transparency value and higher GAC, transparency order must be low enough to justify it.

Here we can also make distinction between two different hard problems, one is finding as low as possible transparency order value while maintaining adequate nonlinearity level, and second problem is to find S-boxes with nonlinearity value between 100 and 112 while having low transparency order values. In the case of evolved S-boxes 1 and 2 from Table 1, we consider S-box 2 to be much better since its nonlinearity is only slightly lower while its transparency order value is significantly lower than in the case of S-box 1.

S-box 2 was evolved in 2325th generation of genetic algorithm which took 96000 seconds and S-box 4 was evolved in 124th generation and that took 4600 seconds. In the evolution process we used a cluster of computers where the average machine is Pentium with 2.6 GHz and 2 GB RAM. For both S-boxes, no further improvement was achieved after reaching the values as specified in Table 1.

4. SIDE-CHANNEL RESISTANCE OF EVOLVED S-BOXES

Using an S-box which has evolved in a way that is ex-

plained in previous sections of the work can be a quite challenging task when area constrained devices are concerned. Since the S-box is not generated through algebraic methods but evolutionary methods, the only way to implement these S-boxes is by using lookup tables (LUTs). When smart cards are considered, a software implementation of AES would make use of lookup tables. Therefore one of the most important platforms where side-channel analysis is considered as a major threat, can be strengthened by using one of the proposed S-boxes at virtually no additional cost. Here, one can argue that it is not entirely area friendly to implement a look-up table for an 8×8 S-box but it should also be considered that side-channel resistance always comes at a cost. A shortcoming of the S-boxes proposed in this work is that they can only be implemented as LUTs in a design. When hardware designs are concerned, this approach can lead to excessive resource usage, and therefore the suitability of using such S-boxes in hardware implementations remains to be considered.

Aside from evaluating the cryptographic properties of the proposed S-boxes, we also evaluated side-channel resistance of software implementation of the new S-boxes. We implemented the new S-boxes on a smart card with an ATMega163 microcontroller. The measurements were collected with a PC oscilloscope at 250 million samples per second sampling rate. A straightforward software implementation of AES is modified to use the proposed S-boxes in side-channel experiments. For running the attacks, the output of the `SubBytes` operation is targeted and Hamming weight model is used to estimate the power consumption. The power estimation for each key candidate is checked for fitness with the actual power measurements through Pearson correlation. The experiment is repeated for 10 different keys selected at random, and the success rate is computed following the methodology proposed by Standaert et al. [28]. The results of our analysis are presented in Figure 2. The analysis is done on an AES implementation which processes the 16 S-box lookups in a random order for each execution of the code. This way, the noise level is increased and therefore the effect of the transparency order is more visible. Practical experiments are done for two new S-boxes: one with the lowest transparency order values we have managed to obtain, and another with the highest nonlinearity and the lowest transparency order for that nonlinearity. It is evident from the figure that using S-boxes with lower transparency order values results in an immediate improvement over the AES S-box in terms of side-channel resistance. Looking at Figure 2, 70% success rate for side-channel analysis on AES S-box requires 12000 traces. However when S-box 2 in Table 1 is considered, one requires at least 14500 traces to achieve the same success rate. Therefore a decrease of 0.5 in transparency order leads to an increase of at least 20.84% in the number of traces required to achieve the same success rate when running side-channel analysis with the noise level present in our experiments. Note that we expect this number to increase with increased noise in measurements.

We further observe that the effect of transparency order is less visible when the level of noise is low. More precisely, the correlation values obtained for the incorrect key guesses increase when S-boxes with lower transparency order values are used. This suggests that more experiments with high level of noise and other countermeasures are interesting for future studies. We can compare the results obtained in this

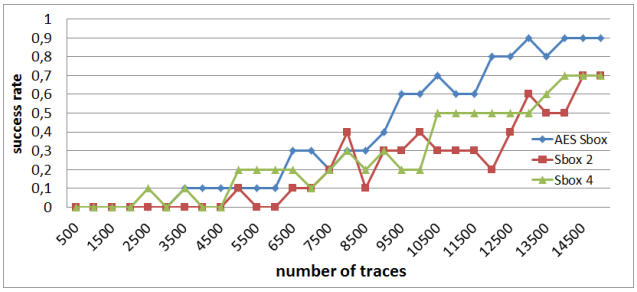

Figure 2: Success rate [28] of the analysis vs the number of traces required.

research with those of Mazumdar et al. [14,15]. When using constrained random search the best results were nonlinearity of 98 and transparency order of 7.782 [15]. The difference of 0.4 for transparency order, from our to their solution, would require substantially more traces for key recovery according to the experiments in [15]. When designing rotation symmetric S-boxes the best nonlinearity was 102 and transparency order 7.76 [14]. Genetic algorithm approach produced S-boxes with comparable nonlinearity (98 to 100), but with far better transparency order of down to 7.35.

5. CONCLUSION

In this work we promote the use of GAs for evolving S-boxes with improved side-channel resistance. Our approach shows potentials in both creation of S-boxes as well as in the evaluation. However, we are aware of the difficulties that lookup table approach could pose. Nevertheless, we do believe that our results have practical values. In general, one can consider the results as a proof of existence of S-boxes with desired properties where we expect that the results can be optimized even further. In this research we used generic GA but the results can be improved by employing custom made GAs or some other evolutionary algorithm. As we have already stated in the previous section, an increase in the level of noise seems to amplify the effect of using S-boxes with lower transparency order values. Therefore, in the same experimental setup, as used in previously mentioned papers, we expect the required number of traces to be significantly higher when our S-boxes are considered. Alas, the fact that no S-boxes were included in those papers restricted us from running experiments with the same setup.

Acknowledgments

This work was supported in part by the Technology Foundation STW (project 12624 - SIDES), The Netherlands Organization for Scientific Research NWO (project ProFIL 628.001.007) and the ICT COST action IC1204 TRUDEVICE.

6. REFERENCES

[1] Matsui, M., Yamagishi, A.: A new method for known plaintext attack of FEAL cipher. In: Proceedings of the 11th annual international conference on Theory and application of cryptographic techniques. EUROCRYPT'92, Berlin, Heidelberg, Springer-Verlag (1993) 81–91

[2] Biham, E., Shamir, A.: Differential Cryptanalysis of DES-like Cryptosystems. In: Proceedings of the 10th Annual International Cryptology Conference on Advances in Cryptology. CRYPTO '90, London, UK, UK, Springer-Verlag (1991) 2–21

[3] Carlet, C.: On highly nonlinear S-boxes and their inability to thwart DPA attacks. In: Proceedings of the 6th international conference on Cryptology in India. INDOCRYPT'05, Berlin, Heidelberg, Springer-Verlag (2005) 49–62

[4] Mangard, S., Oswald, E., Popp, T.: Power Analysis Attacks: Revealing the Secrets of Smart Cards (Advances in Information Security). Springer-Verlag New York, Inc., Secaucus, NJ, US1A (2007)

[5] Shannon, C.: Communication theory of secrecy systems. Bell System Technical Journal **28**(4) (1949) 656–715

[6] Prouff, E.: DPA Attacks and S-Boxes. In: Fast Software Encryption: 12th International Workshop, FSE 2005, Paris, France, February 21-23, 2005, Revised Selected Papers. Volume 3557 of Lecture Notes in Computer Science., Springer (2005) 424–441

[7] Kocher, P., Jaffe, J., Jun, B.: Differential Power Analysis. In Wiener, M., ed.: Advances in Cryptology: Proceedings of CRYPTO'99. Number 1666 in Lecture Notes in Computer Science, Springer-Verlag (1999) 388–397

[8] Braeken, A.: Cryptographic Properties of Boolean Functions and S-Boxes. PhD thesis, Katholieke Universiteit Leuven (2006)

[9] Bernstein, D.J.: Cache-timing attacks on AES (2004)

[10] Millan, W., Clark, A., Dawson, E.: Heuristic Design of Cryptographically Strong Balanced Boolean Functions. In: Advances in Cryptology - EUROCRYPT '98. (1998) 489–499

[11] Clark, J.A., Jacob, J.L., Stepney, S., Maitra, S., Millan, W.: Evolving Boolean Functions Satisfying Multiple Criteria. In: Progress in Cryptology - INDOCRYPT 2002. (2002) 246–259

[12] Clark, J.A., Jacob, J.L., Stepney, S.: The design of S-boxes by simulated annealing. New Generation Computing **23**(3) (September 2005) 219–231

[13] Burnett, L., Carter, G., Dawson, E., Millan, W.: Efficient Methods for Generating MARS-Like S-Boxes. In: Proceedings of the 7th International Workshop on Fast Software Encryption. FSE '00, London, UK, UK, Springer-Verlag (2001) 300–314

[14] Mazumdar, B., Mukhopadhyay, D., Sengupta, I.: 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) 87–92

[15] Mazumdar, B., Mukhopadhay, D., Sengupta, I.: Constrained Search for a Class of Good Bijective S-Boxes with Improved DPA Resistivity. Information Forensics and Security, IEEE Transactions on **PP**(99) (2013) 1–1

[16] Wolpert, D.H., Macready, W.G.: No Free Lunch Theorems for Optimization. IEEE Transactions on Evolutionary Computation **1**(1) (April 1997) 67–82

[17] Kocher, P.: Timing attacks on implementations of Diffie-Hellman, RSA, DSS and other systems. In Koblitz, N., ed.: Advances in Cryptology: Proceedings of CRYPTO'96. Number 1109 in Lecture Notes in Computer Science, Springer-Verlag (1996) 104–113

[18] Quisquater, J.J., Samyde, D.: ElectroMagnetic Analysis (EMA): Measures and Counter-Measures for Smard Cards. In Attali, I., Jensen, T.P., eds.: Smart Card Programming and Security (E-smart 2001). Volume 2140 of Lecture Notes in Computer Science., Springer-Verlag (2001) 200–210

[19] Crama, Y., Hammer, P.L.: Boolean Models and Methods in Mathematics, Computer Science, and Engineering. 1st edn. Cambridge University Press, New York, NY, USA (2010)

[20] Guilley, S., Pacalet, R.: Differential Power Analysis Model and Some Results. In: In proceedings of CARDIS 2004, Kluwer Academic Publishers (2004) 127–142

[21] Weise, T. In: Global Optimization Algorithms Theory and Application. (2009)

[22] Eiben, A.E., Smith, J.E. In: Introduction to Evolutionary Computing. Springer-Verlag, Berlin Heidelberg New York, USA (2003)

[23] Boese, K.D.: Cost Versus Distance In the Traveling Salesman Problem. Technical report (1995)

[24] Whitley, D., Hains, D., Howe, A.: Tunneling between optima: partition crossover for the traveling salesman problem. In: Proceedings of the 11th Annual conference on Genetic and evolutionary computation. GECCO '09, New York, NY, USA, ACM (2009) 915–922

[25] Zhang, X., Zheng, Y.: GAC-the criterion of global avalanche characteristics of cryptographic functions. Journal of Universal Computer Science **1**(5) (1995) 316–333

[26] Burnett, L.D.: Heuristic Optimization of Boolean Functions and Substitution Boxes for Cryptography. PhD thesis, Queensland University of Technology (2005)

[27] Nyberg, K.: 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. Volume 547 of Lecture Notes in Computer Science., Springer (1991) 378–386

[28] Standaert, F.X., Malkin, T., Yung, M.: A unified framework for the analysis of side-channel key recovery attacks. In Joux, A., ed.: Advances in Cryptology - EUROCRYPT 2009. Volume 5479 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2009) 443–461