Block Ciphers that are Easier to Mask: How Far Can we Go?

B. Gérard[1,2], V. Grosso[1], M. Naya-Plasencia[3], F.-X. Standaert[1]

[1] ICTEAM/ELEN/Crypto Group, Université catholique de Louvain, Belgium.
[2] Direction Générale de l'Armement, France. [3] INRIA Paris-Rocquencourt, France.

Abstract. The design and analysis of lightweight block ciphers has been a very active research area over the last couple of years, with many innovative proposals trying to optimize different performance figures. However, since these block ciphers are dedicated to low-cost embedded devices, their implementation is also a typical target for side-channel adversaries. As preventing such attacks with countermeasures usually implies significant performance overheads, a natural open problem is to propose new algorithms for which physical security is considered as an optimization criteria, hence allowing better performances again. We tackle this problem by studying how much we can tweak standard block ciphers such as the AES Rijndael in order to allow efficient masking (that is one of the most frequently considered solutions to improve security against side-channel attacks). For this purpose, we first investigate alternative S-boxes and round structures. We show that both approaches can be used separately in order to limit the total number of non-linear operations in the block cipher, hence allowing more efficient masking. We then combine these ideas into a concrete instance of block cipher called `Zorro`. We further provide a detailed security analysis of this new cipher taking its design specificities into account, leading us to exploit innovative techniques borrowed from hash function cryptanalysis (that are sometimes of independent interest). Eventually, we conclude the paper by evaluating the efficiency of masked `Zorro` implementations in an 8-bit microcontroller, and exhibit their interesting performance figures.

1 Introduction

Masking (aka secret sharing) is a widespread countermeasure against side-channel attacks (SCA) [14]. It essentially consists in randomizing the internal state of a device in such a way that the observation of few (say d) intermediate values during a cryptographic computation will not provide any information about any of the secret (aka sensitive) variables. This property is known as the "d-th order SCA security" and was formalized by Coron et al. as follows [10]: *A masked implementation is d-th order secure if every d-tuple of the intermediate values it computes is independent of any sensitive variable.* Reaching higher-order security is a theoretically sound approach for preventing SCAs, as it ensures that any adversary targeting the masked implementation will have to "combine" the information from at least $d+1$ intermediate computations. More precisely, if one can guarantee that the leakage samples corresponding to the manipulation of the different shares of a masking scheme are independent, then a higher-order security implies that an adversary will have to estimate the $d+1$-th moment of the leakage distribution (conditioned on a sensitive variable), leading to an exponential increase of the SCA data complexity [9][1]. In practice though, this exponential security increase only becomes meaningful if combined with a sufficient amount of noise in the side-channel leakage samples [34]. Also, the condition of independent leakage for the shares may turn out to be difficult to fulfill because of physical artifacts, e.g. glitches occurring in integrated circuits [21]. Yet, and despite these constraints, masking has proven to be one of the most satisfying solutions to improve security against SCAs, especially in the context of protected software implementations in smart cards [24, 30–32].

In general, the most difficult computations to mask are the ones that are non-linear over the group operation used to share the sensitive variables (e.g. the S-boxes in a block cipher). Asymptotically, the time complexity of masking such non-linear operations grows at least quadratically with the order d. As a result, a variety of research works have focused on specializing masking to certain algorithms (most frequently the AES Rijndael, see e.g. [8, 23]), in order to reduce its implementation overheads. More recently, the opposite approach has been undertaken by Piret et al. [26]. In a paper presented at ACNS 2012, the authors suggested that improved SCA security could be achieved at a lower implementation cost by specializing a block cipher for efficient masking. For this purpose, they started from the provably secure scheme proposed by Rivain and Prouff at CHES 2010, and specified a design allowing better performances than the AES Rijndael as the order of the masking increases. More precisely, the authors first observed that bijective S-boxes that are at the same time easy to mask and have good properties for resisting standard cryptanalysis are remarkably close to the AES S-box. As a result, they investigated the gains obtained with non-bijective S-boxes and described a Feistel network with a Substitution-Permutation Network (SPN) based round function taking advantage of this S-box. One interesting feature of this approach is that its impact on the performances of block cipher implementations will grow with the the physical security level (informally measured with the order d). That is, it enables performance gains that become more significant as we move towards physically secure implementations.

In this paper, we complement this first piece of work and further investigate design principles that could be exploited to improve the security of block ciphers implementations against SCAs thanks to the masking countermeasure. In particular, we investigate two important directions left open by Piret et al. First, we observe that non-bijective S-boxes usually lead to simple non-profiled attacks (as their output directly gives rise to "meaningful leakage models" [35]). As recently shown by Whitnall et al., we even have a proof that generic (non-profiled) SCAs against bijective S-boxes cannot exist [36]. This naturally gives a strong incentive to consider bijective S-boxes in block ciphers that are purposed for masked

[1] In certain scenarios, e.g. in a software implementation where all the shares are manipulated at different time instants, masking may also increase the time complexity of the attacks, as an adversary will have to test all the pairs, triples, ... of samples to extract information from a 2nd, 3rd, ... secure implementation.

implementations. Hence, we analyze the possibility to trade a bit of the classical S-box properties (linearity, differential profile, algebraic degree) for bijectivity and more efficient masking. Second, we observe that the previous work from ACNS 2012 focused on the S-box design in order to allow efficient masking. This is a natural first step as it constitutes the only non-linear element of most block ciphers. Yet, it is also appealing to investigate whether the algorithm structure could not be modified in order to limit the total number of S-boxes executed during an encryption. We investigate this possibility and suggest that irregular designs in which only a part of the state goes through an S-box in each round can be used for this purpose, if the diffusion layer is adapted to this setting.

Roughly speaking, our results show that each of the principles that we propose (i.e. the modified S-box and structure) allows dividing the total number of non-linear operations in an AES-like block cipher execution by two (compared to the original AES Rijndael). We then describe a new block cipher for efficient masking, that combines these two ideas in order to reduce this total number of non-linear operations by a factor four. We call this cipher `Zorro` in reference to the masked fictional character. We further provide a detailed security evaluation of our proposal, considering state-of-the-art and dedicated cryptanalysis, in order to determine the number of rounds needed to obtain a secure cipher. Because of the irregular structure of `Zorro`, this analysis borrows recent tools from hash function cryptanalysis and describes new techniques for providing security bounds (e.g. against linear and differential cryptanalysis). We conclude with performance evaluations exhibiting that `Zorro` already leads to interesting performance gains for small security orders $d = 1, 2, 3$. Note that because of place constraints, a part of the security analysis and several background appendices have been deferred the long version of the paper, available from the IACR ePrint.

2 Bijective S-boxes that are easier to mask

In this section we aim at finding an 8-bit S-box having both a small masking cost and good cryptographic properties regarding standard cryptanalysis criteria (i.e. non-linearity, differential profile, algebraic degree. For this purpose, we will use the number of field multiplications and amount of randomness needed to execute a shared S-box as performance metrics. As discussed in [31], reducing this number directly leads to more efficient Boolean masking. Interestingly, it is also beneficial for more advanced (polynomial) masking schemes inspired from the multiparty computation literature, such as proposed by Prouff and Roche [28]. So our proposal is generally suitable for two important categories of masking schemes that (provably) generalize to high security orders. For reference, we first recall that the AES S-box consists in the composition of an inversion of the element in the field $GF(2^8)$ and an affine transformation A: $S_{AES} : x \mapsto A(x^{-1})$. Starting from this standard example, a natural objective would be to find an S-box that can be masked with a lower cost than the AES one (i.e. an S-box that can be computed using less than 4 multiplications [31]), and with similar security properties (i.e. a maximum of the differential spectrum close to 4, a maximum of the Walsh spectrum close to 32, and a high algebraic degree). Since there are $2^8!$ permutations over $GF(2^8)$, an exhaustive analysis of all these S-boxes is computationally unfeasible. Hence, we propose two different approaches to cover various S-boxes in our analysis. First, we exhaustively consider the S-boxes having a sparse polynomial representation (essentially one or two non-zero coefficients). Next, we investigate some proposals for constructing 8-bit S-boxes from a combination of smaller ones. In particular, we consider a number of solutions of low-cost S-boxes that have been previously proposed in the literature.

2.1 Exhaustive search among sparse polynomials

Monomials in $\boldsymbol{GF(2^8)}$. First notice that in $GF(2^8)$ the square function is linear. Hence, we can define an equivalence relation between exponents: $e_1 \sim e_2 \Leftrightarrow \exists\, k \in \mathbb{N}$ st. $e_1 = e_2\, 2^k \mod 255$. This relation groups exponents in 34 different equivalence classes. Only 16 classes out of the 34 lead to bijective functions. The AES exponent has the best security parameters and requires four multiplications. Our goal is to find an S-box with a lower number of multiplications, maintaining good (although not optimal) security features. As detailed in the long version of the paper, exponents $7, 29$ and 37 are interesting candidates.

Binomials in $\boldsymbol{GF(2^8)}$. We also performed an exhaustive search over all the S-boxes defined by a binomial. Note that in this case, an additional (refreshing) mask is required for the additions performed on pairs of dependent variables (in order to maintain the d-th order security). Again, we were only interested in S-boxes that can be computed in less than 4 multiplications. A few examples of the best improvements found are given next:

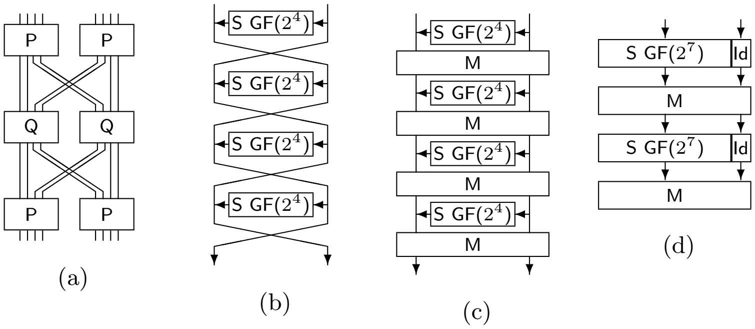
- **2 multiplications.** We found binomials having properties similar to monomials X^7 and X^{37}, with better non-linearity (a maximum of the Walsh spectrum between 64 and 48). Binomial $8X^{97} + X^{12}$ is an example.
- **3 multiplications.** In this case, we additionally found several binomials reducing both the maximum value of the Walsh spectrum (from 64 to 48) and the maximum value of the differential spectrum (from 10 to 6) compared to the monomial X^{29}. Binomial $155X^7 + X^{92}$ is an example.

2.2 Constructing 8-bit S-boxes from smaller ones

As the exhaustive analysis of more complex polynomial representations becomes computationally intractable, we now focus on a number of alternatives based on the combination of smaller S-boxes. In particular, we focus on constructions based on 4-bit S-boxes that were previously proposed, and on 7-bit S-boxes (in order to benefit from the properties of S-boxes with an odd number of bits).

Building on $\boldsymbol{GF(2^4)}$ S-boxes. This is the approach chosen by the designers of PICARO. Namely, they selected an S-box that can be computed using only 4 secure multiplications over $GF(2^4)$. This S-box has good security properties, excepted that its algebraic degree is 4 and that it is non-bijective.

In general, constructing 8-bit S-boxes from the combination of 4-bit S-boxes allows decreasing the memory requirements, possibly at the cost of an increased execution time (as we generally need to iterate these smaller S-boxes). That is, just putting two 4-bit S-boxes side-by-side allows no interaction between the two nibbles of the byte. Hence the maximum of the Walsh spectrum and the maximum of the differential spectrum of the resulting 8-bit S-box are 2^4 times larger than the one of its 4-bit building block. This weakness can be mitigated by using at least two layers of 4-bit S-boxes interleaved with nibble-mixing linear operations. For instance, the KHAZAD [1] and ICEBERG [33] ciphers are using 8-bit S-boxes obtained from three applications of 4-bit S-box layers, interleaved with a bit permutation mixing two bits of each nibble (as illustrated in Figure 4(a)). The resulting S-boxes show relatively good security properties and have maximal algebraic degree. Unfortunately, these proposals are not good candidates to improve the performances of a masked implementations, since six 4-bit S-boxes have to be computed to obtain one 8-bit S-box. As any non-linear permutation in $GF(2^4)$ requires at least 2 multiplications, even using only two layers would cost more secure multiplications than the AES S-box.



(a)

(b)

(c)

(d)

Fig. 1: (a): ICEBERG S-box. (b) 4-round Feistel network w/o linear layer. (c) 4-round Feistel network with linear layer. (d) Combination of 7-bit S-boxes with linear layer.

Another natural alternative to double the size of an S-box is to build on a small Feistel network, as illustrated in Figure 4(b). Note that in this case, we need to perform at least 3 rounds to ensure that security properties against statistical cryptanalyses will be improved compared to the ones of the underlying 4-bit S-box. Indeed, let us choose a differential (or linear) mask with all active bits in the left part of the input; then after 1 round we obtain the same difference in the right part; hence the differential (or linear) approximation probability after two rounds will be the one of the small S-box again. In fact, an exhaustive analysis revealed that 4-round networks are generally required to obtain good cryptanalytic properties. However, it also turned out that adding a linear layer could lead to improved results for S-boxes that are efficiently masked. That is, as illustrated in Figure 4(c), we can add an invertible 8×8 binary matrix to mix the bits of the two Feistel branches between each round. Such a layer allows improving the differential and linear properties of the S-box, with limited impact on the cost of its masked implementations (since the transform is linear).

Example 1. We instantiate the 4-round Feistel network of Figure 4(c) with a 4-bit S-box corresponding to the monomial X^3, and add an 8-bit linear transformation M_1 (given in long version of the paper) at the end of each round. The corresponding 8-bit S-box has a maximum differential spectrum of 10, a maximum of the Walsh spectrum equal to 64 and an algebraic degree of 7. It can be computed using 4 secure multiplications in $GF(2^4)$.

Example 2. We instantiate the 4-round Feistel network of Figure 4(c) with a 4-bit S-box using the polynomial $8X + 7X^2 + 7X^3 + 14X^4 + 3X^6 + 6X^8 + 9X^9 + 5X^{12}$ (which can be computed with 1 multiplication), and add an 8-bit linear transformation M_2 (given in long version of the paper) at the end of each round. The corresponding 8-bit S-box has a maximum differential spectrum of 8, a maximum of the Walsh spectrum equal to 64 and an algebraic degree of 6. It can also be computed using 4 secure multiplications in $GF(2^4)$.

Exploiting $GF(2^7)$ and linear layers. We finally investigated the use of a smaller S-box in $GF(2^7)$. This choice was motivated by the fact that S-boxes in $GF(2^n)$ with n odd provide better security properties against differential cryptanalysis than S-boxes acting on an even number of bits. For instance, the existence of Almost Perfect Non-linear permutations (aka APN permutations) is still an open problem for even values of n while many have been constructed for odd values of n. Hence, we expect that low-cost S-boxes acting on 7 bits will exhibit relatively good security properties. As in the previous paragraph, moving from a 7-bit to an 8-bit S-box can be done by combining the 7-bit S-box with an 8-bit linear transform. That is, we used the S-box in Figure 4(d), where the 7-bit S-box is applied twice, separated by a linear transformation to mix bits inbetween. This implies that good masking properties could only be obtained if the 7-bit S-box uses only a single multiplication. We found several 8-bit S-boxes using 2-multiplications based on this design, having 64 as maximum of the Walsh spectrum, 10 as maximum of the differential spectrum and 4 as algebraic degree.

2.3 Comparing proposed S-boxes to AES one

We compiled the results we obtained in Table 1, in which our performance and security metrics are reported. As explicit with the column “additional operations”, such a table is limited in providing precise estimates of the exact implementation costs, as these costs are always technology-dependent. Yet, it provides general indications about S-box candidates for efficient masking, and also complements the work of Piret et al. in providing some interesting bijective proposals.

3 Reducing the number of S-box executions

The previous section discussed how to reduce the number of multiplications per S-box execution in a block cipher, by trading cryptanalytic properties for more

Table 1: Comparison of the proposals.

	required randomness (bit)			# sec. mult.	additional operations	security properties		
	$d=1$	$d=2$	d			$deg(S)$	max Δ_S	max Ω_S
AES	48	128	$16d^2+32d$	4 (GF(2^8))	7 squ. + 1 Diff. matrix	7	4	32
PICARO	16	48	$8d^2+8d$	4 (GF(2^4))	2 squ.	4	4	68
X^7	24	64	$8d^2+16d$	2 (GF(2^8))	2 squ. + 1 Diff. matrix	3	6	64
X^{29}	32	88	$12d^2+20d$	3 (GF(2^8))	4 squ. + 1 Diff. matrix	4	10	64
X^{37}	24	64	$8d^2+16d$	2 (GF(2^8))	5 squ. + 1 Diff. matrix	3	6	64
$8X^{97}+X^{12}$	32	80	$8d^2+24d$	2 (GF(2^8))	6 squ. + 1 Diff. matrix	3	6	48
$155X^7+X^{92}$	40	104	$12d^2+28d$	3 (GF(2^8))	8 squ. + 1 Diff. matrix	4	6	48
Ex. 1	32	80	$8d^2+24d$	4 (GF(2^4))	4 squ. + 4 Diff. matrix	7	10	64
Ex. 2	48	112	$8d^2+40d$	4 (GF(2^4))	28 squ. + 4 Diff. matrix	6	8	64

efficient masking. A complementary approach in order to design a block cipher that is easy to mask is to additionally reduce the total number of S-box executions. For this purpose, a natural solution is to consider rounds where not all the state goes through the S-boxes. To some extent, this proposal can be viewed as similar to an NLFSR-based cipher (e.g. Grain [16], Katan [6], Trivium [7]), where the application of a non-linear component to the state is not homogeneous. For example, say we consider two n-bit block ciphers with s-bit S-boxes: the first (parallel) one applies n/s S-boxes in parallel in each of its R rounds, while the second (serial) one applies only a single S-box per round, at the cost of a larger number of rounds R'. If we can reach a situation such that $R' < R \cdot \frac{n}{s}$, then the second cipher will indeed require less S-boxes in total, hence being easier to protect against side-channel attacks. Of course, the number of S-box executions in the serial version does not have to be stuck at one, and different trade-offs are possible. In general, the relevance of such a proposal highly depends on the diffusion layer. For example, an AES-like structure is nicely suited to this goal. The rationale behind this intuition essentially relates to the fact that the AES Rijndael has strong security margins against statistical attacks, and the most serious concerns motivating its number of rounds are structural (e.g. [20]). Hence, iterating simplified rounds seems a natural way to prevent such structural attacks while maintaining security against linear/differential cryptanalysis. Furthermore, the impact of linear hulls and differentials in ciphers with strong diffusion could ideally lead to reductions in the total number of S-box executions required to reach a cipher that is secure against statistical attacks. In the following, we show that a modified AES cipher with 4 S-boxes per round (rather than 16 in the standard version) is indeed a good candidate for this purpose.

3.1 The AES Rijndael

The AES Rijndael was designed by Daemen and Rijmen [12]. It operates on message blocks of 128 bits, that can be seen as a matrix of 4 × 4 bytes. One round is composed of four transformations. In SubBytes (SB), a single 8-bit S-box is applied 16 times in parallel to each byte of the state matrix. In ShiftRows (SR), the the 4 bytes in the ith row of the state matrix are rotated by i positions to the left. In MixColumns (MC), a linear transformation defined by an MDS matrix is applied independently to each column of the state matrix. Finally, in AddKey (AK), a 128-bit subkey provided by the key scheduling is added to the internal state by an exclusive or. Depending on the size of the key, the number of rounds varies from 10 to 14. We will compare our design with the 128-bit version, which iterates 10 rounds, with a key whitening in the first one, and no MC in the last one. We do not describe the key scheduling as we will not reuse it.

3.2 Preliminary investigations: how many S-boxes per round?

As in the previous section (about S-boxes that are easier to mask), an exhaustive analysis of all the round structures that could give rise to less S-box executions in total is out of reach. Yet, and as this number of S-box executions mainly depends on the SB operations, we considered several variants of it, while keeping SR, MC and AK unchanged. For this purpose, we have first analyzed how some elementary diffusion properties depend on the number and positions of the S-boxes within the state. Namely, we considered (1) the number of rounds so that all the input bytes have passed at least once through an S-box (NrSbox); (2) the number of rounds so that all the output bytes have at least one non-linear term (NrNlin); and (3) the maximal number of rounds so that an input difference has a non-linear effect in all the output bytes (NrDiff). In all three cases, these number of rounds should ideally be low. They are given in Table 2 for different S-box configurations. While such an analysis is of course heuristic, it indicates that considering four S-boxes per round, located in a single row of the state matrix seems an appealing solution. In the following, our goal will be to show that an AES-like block cipher where each round only applies four "easy-to-mask" S-boxes as found in the previous section can be secure. In particular, we will select the number of rounds as $R' = 24$, so that we have (roughly) twice less S-boxes executed than the original AES Rijndael (i.e. 24×4 vs. 10×16).

	NrSbox	NrNlin	NrDiff
1 S-box	3	2	4
4 S-boxes, 1 line	2	1	3
8 S-boxes, 2 lines	2	1	3
4 S-boxes, 1 column	3	1	3
4 S-boxes, 1 diagonal	2	2	3
4 S-boxes, 1 per column	2	2	3
4 S-boxes, Square	3	2	4

Table 2: Diffusion properties for different SB* configurations.

3.3 The block cipher Zorro: specifications

We will use a block size and key size of $n = 128$ bits, iterate 24 rounds and call the combination of 4 rounds a step. Each round is a composition of four transforms: SB*, AC, SR, and MC, where the two last ones are exactly the same operations as in the AES Rijndael, SB* is a variant of SB where only 4 S-boxes are applied to the 4 bytes of the first row in the state matrix, and AC is a round-constant addition described in Appendix A. We additionally perform a key addition AK before the first and after each step. As for the selection of the S-box, we will use Example 1 from the previous section, and just add the constant 0xB2 to remove a fixed point (a table representation of this S-box is given in Appendix B). Eventually, and order to maintain high implementation efficiency, we did not design any complex key scheduling and simply add the master key each time AK is called - as in the block cipher LED [15]. Using less key additions than in LED is justified by the exclusion of related-key attacks from our security claims. As for other lightweight block ciphers such as NOEKEON [11] or PRINCE [5], we believe that related-key attacks are not relevant for the intended use case (e.g. challenge-response authentication in smart cards), and mainly focused on the generation of a good permutation in the single key setting. A schematic view of the full cipher is given in Figure 2. Reduced-round versions (used in the following) maintain at least three steps, with number of rounds following the pattern: 4-4-4-4-4-4, 4-4-4-4-4-3,4-4-4-4-4-2, 4-4-4-4-4-1, 4-4-4-4-4, . . .

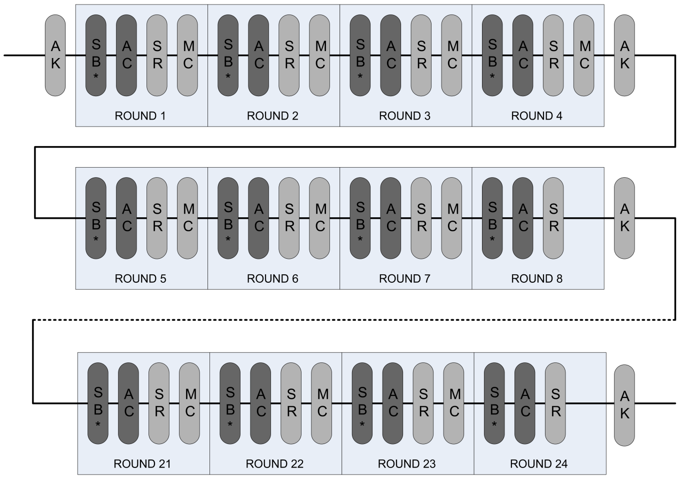


Fig. 2: Block cipher `Zorro`: light gray operations are AES-like, dark gray ones are new.

4 Security analysis

Despite its AES-like flavor, the irregular structure of the block cipher `Zorro` makes it quite different than most recently proposed SPNs. As a result, its security evaluation also requires more dedicated cryptanalysis than usually considered when designing such regular ciphers. In this section, we provide a preliminary investigation of a number of standard and less standard attacks against `Zorro`, paying a particular attention to different solutions to exploit the modified non-linear layer $\texttt{SB}^*$. While further studies by external cryptanalysts would certainly be welcome, we hope that the following analysis provides reasonable confidence that the proposed structure can lead to a secure block cipher.

4.1 Linear/differential cryptanalysis.

In general, security against linear [22] and differential [2] cryptanalysis can be estimated by counting the number of active S-boxes [13]. Based on the specifications in the previous section, we would need to pass through 28 (resp. 32)

S-boxes in order to reach a security level of 2^{128} against differential (resp. linear) cryptanalysis. Nevertheless, since less than 16 S-boxes are applied per round, simple bounds based on the MDS property of the diffusion layer cannot be obtained such as for the AES. An easy shortcoming is that trails that do not start in the first state row will be propagated through the second round with probability one. Besides, since the S-boxes only apply to one out of the 4 input bytes of `MC` in each round, the number of active S-boxes also progresses slower. As a result, the main question for bounding security against these statistical attacks is to determine the extent to which actual characteristics can take advantage of this feature, by keeping a maximum number of inactive S-boxes.

For this purpose, we propose a technique inspired by hash functions cryptanalysis, that finds the best balance between this number of inactive S-boxes and the number of freedom degrees for the differential (or linear) paths. Taking the example of differential cryptanalysis, we first consider a fully active input state (we discuss next how to adapt our reasoning to other input differences) and a fixed (unknown) key. In this case, we have $16 + 16$ degrees of freedom at the beginning of the differential path (in bytes, i.e. we have 2^{32*8} possible trials to test if the differential path is verified). A first observation is that, in order to have x inactive S-boxes in the next round, we need to verify at least x byte conditions through the `MC` operation, which will spend x bytes of the freedom degrees available. Conversely, we have that verifying x byte conditions through `MC` can desactivate at most x S-boxes in the following rounds[2]. Our bounds then follow from the fact that desactivating an S-box is only possible as long as degrees of freedom are available (otherwise there will be no solutions for the differential path). That is, we can consider that for each round i we can ask x_i conditions to be verified through the `MC` transform, and that at most x_i S-boxes will not be activated in the following rounds because of these conditions. Hence, the following inequalities have to be verified for finding a valid path. They represent the degrees of freedom still available after r rounds, and the cumulated number of active S-boxes (that must be smaller than 28 as previously pointed out):

$$\sum_{i=1}^{r} x_i < 32, \qquad \text{and} \qquad 4 \times r - \sum_{i=1}^{r} x_i < 28.$$

For simplicity, we can consider the average number of conditions $\bar{x}$ that we can impose at each round. We observe that the highest number of rounds is achieved for $r = 14$ and $\bar{x} = 32/14 = 2.285$, where we have 24 active S-boxes and no more freedom degrees available (for 15 rounds, the number of active S-boxes exceeds 28). Eventually, we note that when the initial state is not completely active, e.g. taking only Y possible differences, we have that with $c_{in} = \log_2(2^{16*8}/Y)/8$

[2] Consider the case where the 1st output byte of `MC` is inactive, i.e. we have one less active S-box in the next round. For more S-boxes to be inactive, we would have to pay more conditions on `MC`. Alternatively, say `MC` has one active output difference per column (implying $x = 12$ byte conditions). Then, we have at most 6 inactive S-boxes in the two next rounds, before coming back to the whole active state with $6 < x$.

byte conditions we will be able to desactivate at most c_{in} S-boxes. Hence, the inequalities taking all possible input differences into account become:

$$\sum_{i=1}^{r} x_i < 32 - c_{in}, \qquad \text{and} \qquad 4 \times r - \sum_{i=1}^{r} x_i - c_{in} < 28.$$

They provide the same result as before: 14 rounds is the upper bound for building a classical differential path[3]. A similar reasoning for linear cryptanalysis leads to an upper bound of 16 rounds (out of 24, leaving us good security margins).

4.2 Truncated differential attacks

In view of the non-linear transformation in Zorro, a natural extension of differential cryptanalysis to investigate is the use of dedicated truncated differentials [18]. In particular, the most damaging truncated differential patterns are those that would exclude active bytes affected by non-linear operations. For this reason, we analyzed the possible existence of cycles of differences that verify transitions from three active rows of the state to another three active rows with probability one for any number of rounds (i.e. excluding non-linear operations). Such patterns are represented in Figure 3, where big squares represent states, small squares represent bytes, highlighted ones are affected by non-linear transformations and gray bytes are the ones with a non-zero difference. Truncated differentials only following the pattern of the figure would never go through the S-boxes. Quite naturally, staying in this pattern for several rounds implies more conditions, but if an input difference exists so that it follows the pattern for some rounds before regenerating this first input difference again, this would imply that the pattern can be followed for an infinite number of rounds as a cycle would have been created. If no cycle exists, we have essentially 4 byte constraints per round for 12 unknowns, and we run out of degrees of freedom for verifying the pattern after 3 rounds. As a result, we essentially have to ensure that no cycle has been created, that would prevent differences to affect the first state row for an infinite number of rounds. The probability that such a cycle exists is small (about $2^{64-96} + 2^{32-96} + 2^{-96} \approx 2^{-32}$). Yet, in order to be sure they do not exist, we performed an exhaustive search over all the 3-row input differences, and checked whether they generate a cycle or end by spreading the difference. The naive cost of such a search is $2^{12*8} = 2^{96}$. We describe a time and memory efficient alternative in the long version of the paper. It allowed us to verify that the pattern of Figure 3 can be verified for at most two rounds.

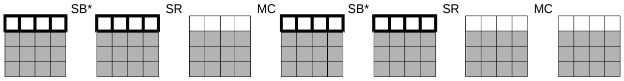


Fig. 3: Two rounds of truncated differential pattern.

[3] Note that despite these bounds to being possibly loose for small number of rounds, they also guarantee security against boomerang attacks. Namely, we have at least 9 active S-boxes after 10 rounds, which would correspond to best differentials with probabilities $p, q \approx 2^{42}$ in a boomerang attack (leading to $p^2q^2 \approx 2^{-168}$).

5 Other cryptanalysis attempts

Because of place constraints, the rest of our security evaluations have only been included in the long version of the paper, in which we additionally evaluated meet-in-the-middle and biclique attacks, impossible differential attacks, derivative and algebraic analyses and rebound attacks. The best cryptanalysis attempt we found is a meet-in-the-middle one, targeting 12 rounds of Zorro. These investigations are admittedly far from exhaustive. Yet, we believe that the attacks evaluated are among the most relevant regarding the structure and components of Zorro. A number of other standard cryptanalysis techniques would naturally apply just like for any other cipher. One can mention the slide attacks introduced in [3] and exploiting the similarity of the round functions (that are prevented by the use of round constants). Another example are integral attacks exploiting properties of the MC transform [20]. Since our modified $\mathtt{SB}^*$ does not affect these diffusion properties, they would target 7 rounds, just as for the AES [19]. We leave the investigation of these other attack paths as a scope for further research.

6 Concluding remarks

To conclude this work, we firstreport on masked implementations of Zorro in an Atmel AtMega644p 8-bit microcontroller. In order to justify the interest of this new cipher, we compared its performance figures with two natural competitors, namely the AES and PICARO. We considered the schemes of Rivain and Prouff [31] for this purpose. In the AES case, we also considered the optimization from Kim et al. [17]. The results of Figure 4 suggest that the AES remains most efficient cipher in the unprotected case, while PICARO and Zorro gradually lead to improved cycle counts with larger masking orders. The fact that Zorro exploits both an improved S-box and a modified structure explains its asymptotic gain over PICARO. Besides, we recall that using bijective S-boxes is important in order to avoid easy attack paths for non-profiled side-channel analysis. Note that considering the polynomial masking scheme of Prouff and Roche in [28] could only lead to more significant gains (the cost of masking is cubic in the security order in this case, compared to quadratic for Boolean masking).

Finally, we stress that the design of Zorro leads to interesting open problems regarding further optimizations for algorithms that are “easy to mask”. Keeping the (generic) criteria of minimizing the number of field multiplications in the algorithm, a natural direction would be to consider cipher designs with stronger diffusion layers such as Khazad [29]. Alternatively, one could also give up a bit of our generality and focus exclusively on Boolean masking (e.g. the Rivain and Prouff 2010 scheme) while giving up polynomial types of masking schemes (e.g. the Prouff and Roche 2011 one). For example, the S-boxes of block ciphers such as PRESENT [4] or NOEKEON [11] require three multiplications in $GF(2^{16})$, which makes them less suitable than Zorro regarding our current optimization criteria (as these ciphers require 16×32 and 31×16 of these S-boxes, respectively). But they have efficient bitslice representations minimizing the number of AND gates, which could lead to further improvements of Boolean masked implementations. In general, taking advantage of bitslicing in this specialized context, while maintaining a "regular" design (e.g. excluding bit manipulations that would leak more on certain bits than others) is an interesting open problem.

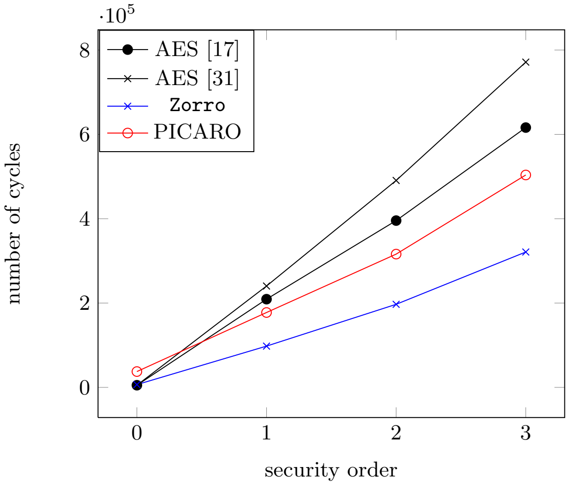


Fig. 4: Performance evaluation.

Acknowledgements. Work funded in parts by the European Commission through the ERC project 280141 (acronym CRASH) and the European ISEC action grant HOME/2010/ISEC/AG/INT-011 B-CCENTRE project. F.-X. Standaert is an associate researcher of the Belgian Fund for Scientific Research (FNRS-F.R.S.).

References

1. P. Barreto and V. Rijmen. The KHAZAD legacy-level block cipher. *Primitive submitted to NESSIE*, page 4, 2000.
2. Eli Biham and Adi Shamir. Differential cryptanalysis of DES-like cryptosystems. In Alfred Menezes and Scott A. Vanstone, editors, *CRYPTO*, volume 537 of *Lecture Notes in Computer Science*, pages 2–21. Springer, 1990.
3. Alex Biryukov and David Wagner. Slide attacks. In Lars R. Knudsen, editor, *FSE*, volume 1636 of *Lecture Notes in Computer Science*, pages 245–259. Springer, 1999.
4. Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: An ultra-lightweight block cipher. In Paillier and Verbauwhede [25], pages 450–466.
5. Julia Borghoff, Anne Canteaut, Tim Gneysu, Elif Bilge Kavun, Miroslav Kneevic, Lars R. Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, Peter Rombouts, Sren S. Thomsen, and Tolga Yalin. PRINCE - a

low-latency block cipher for pervasive computing applications. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT*, volume 7658 of *Lecture Notes in Computer Science*, pages 208–225. Springer, 2012.

6. Christophe De Cannière, Orr Dunkelman, and Miroslav Knezevic. KATAN and KTANTAN - a family of small and efficient hardware-oriented block ciphers. In Christophe Clavier and Kris Gaj, editors, *CHES*, volume 5747 of *Lecture Notes in Computer Science*, pages 272–288. Springer, 2009.
7. Christophe De Cannière and Bart Preneel. Trivium. In Matthew J. B. Robshaw and Olivier Billet, editors, *The eSTREAM Finalists*, volume 4986 of *Lecture Notes in Computer Science*, pages 244–266. Springer, 2008.
8. D. Canright and Lejla Batina. A very compact "perfectly masked" S-Box for AES. In Steven M. Bellovin, Rosario Gennaro, Angelos D. Keromytis, and Moti Yung, editors, *ACNS*, volume 5037 of *Lecture Notes in Computer Science*, pages 446–459, 2008.
9. Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards sound approaches to counteract power-analysis attacks. In Michael J. Wiener, editor, *CRYPTO*, volume 1666 of *Lecture Notes in Computer Science*, pages 398–412. Springer, 1999.
10. Jean-Sébastien Coron, Emmanuel Prouff, and Matthieu Rivain. Side channel cryptanalysis of a higher order masking scheme. In Paillier and Verbauwhede [25], pages 28–44.
11. Joan Daemen, Michaël Peeters, Gilles Van Assche, and Vincent Rijmen. Nessie proposal: NOEKEON, 2000. Available online at `http://gro.noekeon.org/Noekeon-spec.pdf`.
12. Joan Daemen and Vincent Rijmen. Rijndael candidate for AES. In AES *Candidate Conference*, pages 343–348, 2000.
13. Joan Daemen and Vincent Rijmen. The wide trail design strategy. In Bahram Honary, editor, *IMA Int. Conf.*, volume 2260 of *Lecture Notes in Computer Science*, pages 222–238. Springer, 2001.
14. Louis Goubin and Jacques Patarin. DES and differential power analysis (the "duplication" method). In Çetin Kaya Koç and Christof Paar, editors, *CHES*, volume 1717 of *Lecture Notes in Computer Science*, pages 158–172. Springer, 1999.
15. Jian Guo, Thomas Peyrin, Axel Poschmann, and Matthew J. B. Robshaw. The LED block cipher. In Preneel and Takagi [27], pages 326–341.
16. Martin Hell, Thomas Johansson, and Willi Meier. Grain: a stream cipher for constrained environments. *IJWMC*, 2(1):86–93, 2007.
17. HeeSeok Kim, Seokhie Hong, and Jongin Lim. A fast and provably secure higher-order masking of AES s-box. In Preneel and Takagi [27], pages 95–107.
18. Lars R. Knudsen. Truncated and higher order differentials. In Bart Preneel, editor, *FSE*, volume 1008 of *Lecture Notes in Computer Science*, pages 196–211. Springer, 1994.
19. Lars R. Knudsen and Vincent Rijmen. Known-key distinguishers for some block ciphers. In *ASIACRYPT*, volume 4833 of *Lecture Notes in Computer Science*, pages 315–324. Springer, 2007.
20. Lars R. Knudsen and David Wagner. Integral cryptanalysis. In Joan Daemen and Vincent Rijmen, editors, *FSE*, volume 2365 of *Lecture Notes in Computer Science*, pages 112–127. Springer, 2002.
21. Stefan Mangard, Thomas Popp, and Berndt M. Gammel. Side-channel leakage of masked CMOS gates. In Alfred Menezes, editor, *CT-RSA*, volume 3376 of *Lecture Notes in Computer Science*, pages 351–365. Springer, 2005.

22. Mitsuru Matsui. Linear cryptoanalysis method for DES cipher. In Tor Helleseth, editor, *EUROCRYPT*, volume 765 of *Lecture Notes in Computer Science*, pages 386–397. Springer, 1993.
23. Elisabeth Oswald, Stefan Mangard, Norbert Pramstaller, and Vincent Rijmen. A side-channel analysis resistant description of the AES S-Box. In Henri Gilbert and Helena Handschuh, editors, *FSE*, volume 3557 of *Lecture Notes in Computer Science*, pages 413–423. Springer, 2005.
24. Elisabeth Oswald and Kai Schramm. An efficient masking scheme for AES software implementations. In JooSeok Song, Taekyoung Kwon, and Moti Yung, editors, *WISA*, volume 3786 of *Lecture Notes in Computer Science*, pages 292–305. Springer, 2005.
25. Pascal Paillier and Ingrid Verbauwhede, editors. *Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings*, volume 4727 of *Lecture Notes in Computer Science*. Springer, 2007.
26. Gilles Piret, Thomas Roche, and Claude Carlet. PICARO - a block cipher allowing efficient higher-order side-channel resistance. In Feng Bao, Pierangela Samarati, and Jianying Zhou, editors, *ACNS*, volume 7341 of *Lecture Notes in Computer Science*, pages 311–328. Springer, 2012.
27. Bart Preneel and Tsuyoshi Takagi, editors. *Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011*, volume 6917 of *Lecture Notes in Computer Science*. Springer, 2011.
28. Emmanuel Prouff and Thomas Roche. Higher-order glitches free implementation of the AES using secure multi-party computation protocols. In Preneel and Takagi [27], pages 63–78.
29. Vincent Rijmen and Paulo Barreto. Nessie proposal: KHAZAD, 2000. Available online at `http://www.larc.usp.br/~pbarreto/KhazadPage.html`.
30. Matthieu Rivain, Emmanuelle Dottax, and Emmanuel Prouff. Block ciphers implementations provably secure against second order side channel analysis. In Kaisa Nyberg, editor, *FSE*, volume 5086 of *Lecture Notes in Computer Science*, pages 127–143. Springer, 2008.
31. Matthieu Rivain and Emmanuel Prouff. Provably secure higher-order masking of AES. In Stefan Mangard and François-Xavier Standaert, editors, *CHES*, volume 6225 of *Lecture Notes in Computer Science*, pages 413–427. Springer, 2010.
32. Kai Schramm and Christof Paar. Higher order masking of the AES. In David Pointcheval, editor, *CT-RSA*, volume 3860 of *Lecture Notes in Computer Science*, pages 208–225. Springer, 2006.
33. François-Xavier Standaert, Gilles Piret, Gaël Rouvroy, Jean-Jacques Quisquater, and Jean-Didier Legat. ICEBERG : An involutional cipher efficient for block encryption in reconfigurable hardware. In Bimal K. Roy and Willi Meier, editors, *FSE*, volume 3017 of *Lecture Notes in Computer Science*, pages 279–299. Springer, 2004.
34. François-Xavier Standaert, Nicolas Veyrat-Charvillon, Elisabeth Oswald, Benedikt Gierlichs, Marcel Medwed, Markus Kasper, and Stefan Mangard. The world is not enough: Another look on second-order DPA. In Masayuki Abe, editor, *ASIACRYPT*, volume 6477 of *Lecture Notes in Computer Science*, pages 112–129. Springer, 2010.
35. Nicolas Veyrat-Charvillon and François-Xavier Standaert. Generic side-channel distinguishers: Improvements and limitations. In Phillip Rogaway, editor,

CRYPTO, volume 6841 of *Lecture Notes in Computer Science*, pages 354–372. Springer, 2011.

36. Carolyn Whitnall, Elisabeth Oswald, and François-Xavier Standaert. The myth of generic DPA...and the magic of learning. Cryptology ePrint Archive, Report 2012/256, 2012. `http://eprint.iacr.org/`.

A Round constants

The round constants addition is limited to the first state row. Constants are generated as $\{i, i, i, i << 3\}$, with i the round index and $<<$ the left shift operator.

B S-box

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	B2	E5	5E	FD	5F	C5	50	BC	DC	4A	FA	88	28	D8	E0	D1
10	B5	D0	3C	B0	99	C1	E8	E2	13	59	A7	FB	71	34	31	F1
20	9F	3A	CE	6E	A8	A4	B4	7E	1F	B7	51	1D	38	9D	46	69
30	53	E	42	1B	F	11	68	CA	AA	6	F0	BD	26	6F	0	D9
40	62	F3	15	60	F2	3D	7F	35	63	2D	67	93	1C	91	F9	9C
50	66	2A	81	20	95	F8	E3	4D	5A	6D	24	7B	B9	EF	DF	DA
60	58	A9	92	76	2E	B3	39	C	29	CD	43	FE	AB	F5	94	23
70	16	80	C0	12	4C	E9	48	19	8	AE	41	70	84	14	A2	D5
80	B8	33	65	BA	ED	17	CF	96	1E	3B	B	C2	C8	B6	BB	8B
90	A1	54	75	C4	10	5D	D6	25	97	E6	FC	49	F7	52	18	86
A0	8D	CB	E1	BF	D7	8E	37	BE	82	CC	64	90	7C	32	8F	4B
B0	AC	1A	EA	D3	F4	6B	2C	FF	55	A	45	9	89	1	30	2B
C0	D2	77	87	72	EB	36	DE	9E	8C	DB	6C	9B	5	2	4E	AF
D0	4	AD	74	C3	EE	A6	F6	C7	7D	40	D4	D	3E	5B	EC	78
E0	A0	B1	44	73	47	5C	98	21	22	61	3F	C6	7A	56	DD	E7
F0	85	C9	8A	57	27	7	9A	3	A3	83	E4	6A	A5	2F	79	4F