On the security of symmetric primitives and the properties of their inner components

Christina Boura

université PARIS-SACLAY

Habilitation à diriger des recherches

On the security of symmetric primitives and the properties of their inner components

Habilitation à diriger des recherches de l'Université Paris-Saclay

présentée et soutenue à Paris, le 17 février 2023, par

Christina BOURA

Composition du jury

Pierre-Alain FOUQUE Professeur, Université Rennes 1	Rapporteur
Gregor LEANDER Professeur, Ruhr University Bochum	Rapporteur
Bart PRENEEL Professeur, KU Leuven	Rapporteur
Orr DUNKELMAN Professeur, University of Haifa	Examinateur
Caroline FONTAINE Directrice de Recherche, CNRS	Examinatrice
Henri GILBERT Responsable du laboratoire de cryptographie de l'ANSSI	Examinateur
Louis GOUBIN Professeur, UVSQ	Examinateur
María NAYA-PLASENCIA Directrice de Recherche, Inria	Examinatrice

Acknowledgements

I would like to sincerely thank Pierre-Alain Fouque, Gregor Leander and Bart Preneel for accepting to review this manuscript, despite their extremely full time schedules. I'm extremely grateful that this thesis has been reviewed by three researchers that have brought so much to the symmetric cryptography community. Thank you Pierre-Alain for all the discussions we had since last year, your guidance and advices. Thank you Bart and Gregor for contributing to make ToSC and FSE such a success.

I'm also truly thankful to Orr Dunkelman, Caroline Fontaine, Louis Goubin, Henri Gilbert and María Naya-Plasencia for accepting to be part of the jury. Thank you Orr for the long trip to attend the defense and I hope that the French administration didn't discourage you to visit Paris more often. Caroline, thank you for the discussions, your kindness and your support. Thank you Louis for being the best possible leader for our CRYPTO team at Versailles and for contributing to make this a so great place to work. Thank you for all the discussions and for always having a solution to any possible administrative problem. Henri, I was always amazed by your enthusiasm for the research, your great ideas and would like to thank you for your exceptional guidance of the PhD students of Versailles. Finally, María, thank you for everything. Your enthusiasm, your great research projects, your support, the encyclopedia, but also the Christmas lunches, crypto diners and all the great time spent together at the different conferences.

After the jury the next words of acknowledgement go naturally to Anne Canteaut. It is entirely thanks to Anne that I'm working today in symmetric cryptography as she proposed me 13 years ago an exciting PhD project and accompanied me in this, and well beyond this, in the most remarkable way. I would like to thank her for all the support, guidance, caring during all these years but also for being the best literature guide ever!

I'm also very grateful to all present and past members of the CRYPTO team at the University of Versailles (Louis, Yann, Michaël, Jacques, Luca, Nicolas, Margot, Rachelle, Pierre, Axel, Édouard, Alex, Cyril, ...) for making this place a wonderful place to work. Luca, thank you for being the nicest colleague at my arrival, for sharing many advises, for having always been available and for the teaching experience. I learned so many things at your side. Another big thanks goes to Yann for all the research and teaching projects together, his funny stories and his very communicative enthusiasm. It is so nice having you as part of the team!

Apart Versailles, the Inria Cosmiq team has always been for me as a second home. I

would like to thank all its members: André, Anne, Anthony, Jean-Pierre, Gaëtan, Léo, María, Nicolas, Pascale and all the students, for the always warm reception and in particular the symmetric part of the team for all the time spent together, all the serious and funny discussions. A special thanks to Léo for his great help whenever this is needed and for having taught my course when I was on a leave.

I feel particularly grateful to be part of this huge and very strong French cryptographic community. I had the chance to meet wonderful colleagues (André, Damien, Emmanuel, Jérémy, Magali, Marine, Thomas F., Thomas I., Thomas P., Matthieu, Pascal, Patrick, Pierre, Sonia, Valentin, Virginie, Xavier, . . .) and to have a lot of interesting discussions, or just spend nice moments during conferences, seminars, research retreats, summer schools, selection committees, ANR meetings, etc. Valentin, thanks for your precious friendship. Patrick thanks for all the nice projects started this year, your friendship and having accompanied my HDR project since its beginning.

A big thanks to Bart Mennink for having been the best possible co-editor for ToSC, for his incredible efficiency, for making this experience so pleasant and for all the funny socks and gossip meetings.

I was extremely lucky to meet Daniel, Margot and Rachelle, the 3 wonderful PhD students I had or have still to co-advise. Thank you for all what I learned at your side.

Thanks to all the members of the ANR project SWAP, Ali, Anne, Fabien, Gaëtan, Henri, Jérémy, Jules, Léo, Louis, Margot, Matthieu, Nicolas, Pascal, Philippe, Rachelle, Sonia, Valentin, Yann, and Yves for all the productive meetings so far.

I am also thankful to my non-cryptographer colleagues at the departement of computer science at Versailles, especially Béatrice, Coline, Franck, Jean-Michel, Sandrine, Pierre, Thierry and Yann S. for all the discussions, advises and their availability. It is a big pleasure to be part of such a great teaching team.

I would also like to thank my homomorphic friends, Ilaria, Mariya and Nicolas for all the passionate discussions on torus and cats, always around a sushi plate. Thanks also to Ilaria, Mariya and Maria C. for all the girls-out evenings at the classical Opera café. Ilaria, thanks for having been the best office-mate ever! Mariya and Maria thank you for your long lasting friendship! Nicolas, thank you for the New York experience and for all the rest.

Finally, these last words go to my family, my parents, my sister, Alexandre and our two wonderful boys Simon and Martin, for all their love and support. Thank you for making everyday life so beautiful!

Table of Contents

Chapter 1

General Introduction

Symmetric cryptography is a central discipline for ensuring information security. Indeed, the use of cryptographic algorithms allows to protect sensitive information when stored on some physical device and permits two parties to communicate safely via an unreliable communication channel. Among these algorithms, those known as symmetric are the only ones that can guarantee a good performance in terms of speed or circuit size for most applications. In public key algorithms, the security evaluation of the primitives is usually done by reducing their security to the difficulty of a more general mathematical problem, and then using the existing literature to argue that its resolution is hard. For symmetric algorithms, there isn't usually such a proxy: insights about the security of an algorithm are obtained through a dedicated cryptanalysis effort. Cryptanalysis can be seen as an essential process for establishing trust towards the symmetric ciphers to be used and deployed and is one of the central themes of this thesis.

The results of this manuscript are centered around the understanding of the security offered by symmetric primitives. This understanding can be achieved via different processes. First, it can be enhanced by a careful study and analysis of the existing cryptanalysis techniques. Since my PhD, I was particularly interested in the generalization and improvement of well-known families of attacks. Some of my works permitted notably to improve the understanding of impossible differential cryptanalysis by providing generic complexity formulas as well as new techniques to improve the complexity of these attacks [BNS14; Bou+18]. Recently, I studied differential attacks, another classical and extremely powerful family of cryptanalysis, and our work permitted to provide improvements to the key-recovery phase of these attacks and to break the full version of `SPEEDY-7-192`, a newly proposed lightweight cipher [Bou+23]. Finally, while studying a new approach to extend the length of meet-in-the-middle (MITM) attacks, we came up with a new cryptanalysis technique, that we called the differential MITM attack. The idea of this new technique is to combine differential and MITM attacks and this combination allowed us to provide the best results against a variant of the `SKINNY` family of block

ciphers [Bou+22].

In parallel with the study of well established attacks against symmetric primitives and the proposal of new variants, I was also interested in the analysis of newly proposed cryptanalysis techniques. When a new attack gets introduced it is often not clear what is the exact influence of the inner components of the cipher on the success of the attack or if a different mathematical modelization of the method could allow to exploit additional properties of the non-linear and linear parts. In this direction, we proposed in [BC16] an alternative perception of the division property, a powerful algorithmic approach to detect integral distinguishers. We proposed to modelize this technique by means of the so-called parity sets and this modelization enabled us on the one hand to provide a simpler formulation and interpretation of the division property and on the other hand to improve the strength of the distinguishers of this type. In a different work, we proposed an alternative formulation of the property multiple-of-n, a structural distinguisher that was initially proposed for the `AES`. Our work allowed us for the first time to understand the exact influence of the internal components of the `AES` on the observed property, to propose a new compact proof of it and to generalize the property to other constructions [BCC19].

Understanding the security of symmetric primitives cannot be fully achieved without an understanding of the properties of their inner components, in particular of the non-linear ones. All components of a symmetric primitive can be seen as vectorial Boolean functions. Studying the properties of these functions, being able to classify them and to propose robust candidates, especially for the non-linear layer, is a work at the border between cryptography and more fundamental mathematics. In my research I worked in particular on Differential Distribution Tables (DDT) [Bou+19], whose properties determine the resistance of a primitive against differential attacks and also on the Boomerang Connectivity Tables (BCT) [BC18; TBP20], which enable a more accurate estimate of the security of a cryptographic function against boomerang attacks. For example, our work permitted both to advance research on optimal functions against differential cryptography, known as APN functions, and to discover families of optimal functions against boomerang attacks by introducing a new notion called boomerang uniformity.

Finally, to provide arguments on the resistance of symmetric primitives against classical attacks designers use very often automatic tools (typically SAT, MILP or CP). This approach, which is gaining more and more popularity, requires, among others, a good understanding of the modeling techniques. I worked on these topics with Daniel Coggia,

my PhD student at the time and we managed to propose efficient modeling techniques for MILP solvers [BC20]. However, it is not always possible to use automated tools to prove the resistance of symmetric cryptosystems against some attacks. Indeed, for specific algorithms working on very large internal states and whose linear layer is defined using operations at the bit level, the best existing MILP models do not manage to reach more than 2, 3 or even 4 rounds. In a recent work with Yann Rotella and our PhD student Margot Funk, we analyzed the hash function `Troika`, which works on $\mathbb{F}_3$, and follows a design very close to that of `Keccak`. To propose bounds on the probability of the differential trails of this function, since using automated tools was not possible, we needed to develop specific highly non-trivial algorithms [BFR22].

Chapter 2

Differential Cryptanalysis and its Generalizations

Contents

This chapter is dedicated to the study of three different cryptanalysis techniques, all of them exploiting the differential properties of the underlying primitives. Section 2.1.1 describes a differential attack against the low-latency cipher SPEEDY and provides, through

this application, some improvement techniques to the key-recovery part of these attacks. Section 2.2 presents a general framework as well as new techniques for the key recovery part of impossible differential attacks, while Section 2.3 introduces a new cryptanalysis method that we called differential meet-in-the-middle attack. Finally, Section 2.4 presents new modelizations by means of linear constraints of the propagation of differential and other properties through a symmetric primitive that permit, among others, to efficiently measure the resistance of SPN ciphers to differential attacks by the use of a MILP solver.

2.1 Differential Cryptanalysis of SPEEDY

The SPEEDY family of ciphers is a family of lightweight block ciphers introduced by Leander, Moos, Moradi and Rasoolzadeh at CHES 2021 [Lea+21]. Together with Nicolas David, Rachelle Heim Boissier and María Naya-Plasencia we analyzed the resistance of this cipher against differential cryptanalysis and managed to break the full-round version of SPEEDY-7-192, the variant offering 192-bit security. We achieved this result by developing an efficient method that permitted us to find high-probability trails and by proposing several improvement techniques for the key-recovery part. This work [Bou+23] was accepted at EUROCRYPT 2023.

2.1.1 Differential Cryptanalysis

Differential attacks are a very popular chosen-plaintext cryptanalysis technique against symmetric primitives [BS90]. Similarly to the majority of attacks against block ciphers, differential attacks are built around a distinguisher. A differential distinguisher exploits as a distinguishing property the existence of a pair of differences $(\delta_{in}, \delta_{out}) \in \mathbb{F}_2^n$, where n is the block size, such that the input difference δ_{in} propagates through some rounds of the cipher to the output difference δ_{out} with a probability significantly higher than 2^{-n}. This distinguisher can then be extended in both directions by adding some rounds that will serve as the key recovery part. We start by presenting the general framework of a classical differential attack (see Figure 2.1).

Let $\Delta = (\delta_{in}, \delta_{out})$ be a differential of probability $P = 2^{-p}$ covering r_Δ rounds. The difference δ_{in} (resp. δ_{out}) then maps to a truncated difference in D_{in}, r_{in} rounds before (resp. D_{out} and r_{out}) with probability 1. We denote by d_{in} (resp. d_{out}) the log_2 of the size of the input (resp. output) difference such that $|D_{in}| = 2^{d_{in}}$ (resp. $|D_{out}| = 2^{d_{out}}$).

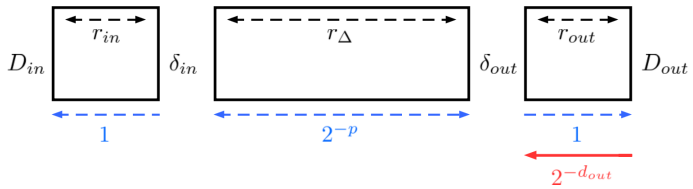


Figure 2.1 – Differential cryptanalysis framework.

Note that the attack can be done in both directions (encryption or decryption) and the most interesting direction is determined by the concrete parameters. However, the general description of the attack remains unchanged no matter the direction. For this it suffices to replace in what follows "ciphertexts", D_{out} and d_{out} by "plaintexts", D_{in} and d_{in}.

Data complexity In order to have enough data to expect that one pair satisfies the differential, one typically uses *structures*. A structure is a set of ciphertexts that have a fixed value in the non-active bits, and that take all possible values in the remaining d_{out} bits. This approach permits to build $(2^{2d_{out}-1})$ pairs inside a structure. The probability to start from a difference in D_{out} and to fall back to a difference δ_{out} is usually $2^{-d_{out}}$. This means that to have one pair that satisfies the differential trail, we need a total of $2^{p+d_{out}}$ pairs that we will obtain by using 2^s structures where s is such that $2^{s+2d_{out}-1} = 2^{p+d_{out}}$, that is $s = p - d_{out} + 1$. Therefore, we need to generate $2^{d_{out}+s} = 2^{p+1}$ ciphertexts and thus the data complexity is $\mathcal{D} = 2^{p+1}$.

Pair sieving The next step for a cryptanalyst is to perform a sieving of the pairs step that will permit her to discard pairs that cannot follow the differential trail. We denote by C_S the average cost of sieving a pair. This cost is in general quite small as it might simply correspond to a table lookup.

Key recovery This step permits to generate a final number of triplets formed by plaintext (or ciphertext) pairs and candidate associated keys that satisfy the differential. This can be done by performing partial key guesses that can be merged thanks to efficient list merging algorithms like the ones of [Nay11], with a quite low additional factor. We denote by C_{KR} the average cost to perform the key recovery step per pair.

Total time and memory complexity We denote by C_E the cost of one encryption. Taking into account the data generation, the data sieve and the key recovery step described

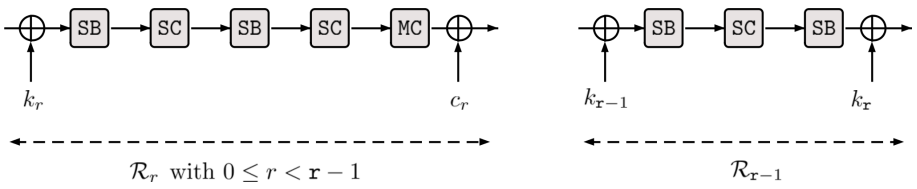


Figure 2.2 – The round function of `SPEEDY-r-192` for the first $\mathtt{r} - 1$ rounds (left) and the last round (right).

above, the *time complexity* $\mathcal{T}$ [1] is given by

$$\mathcal{T} = \left(2^{p+1} + 2^{p+1}\frac{C_S}{C_E} + 2^{s+2d_{in}-1-n+d_{out}}\frac{C_{KR}}{C_E}\right)C_E.$$

2.1.2 Specification of the SPEEDY Family of Block Ciphers

The main design goal of `SPEEDY` was to be fast in CMOS hardware by achieving extremely low latency. This goal was notably reached thanks to the design of a dedicated 6-bit bijective S-box. The three main variants of `SPEEDY` have a block size and a key size of 192 bits and iterate a round function for `r` rounds, where `r` is 5, 6 or 7. They are denoted by `SPEEDY-r-192`. The internal state is viewed as a 32×6 rectangle-array of bits.

Round function of `SPEEDY-r-192` The internal state is first initialized with the 192-bit plaintext. Then, a round function $\mathcal{R}_r$ is applied to the state `r` times, where `r` is typically 5, 6 or 7. The round function is composed of four operations: First, `AddRoundKey` ($\mathtt{A}_{k_r}$) XORs the round subkey k_r to the state. Then, the `SubBox (SB)` operation applies a 6-bit S-box to each row of the state. Follows the `ShiftColumns (SC)` operation that rotates each column of the state by a different offset. These two operations (`SB` and `SC`) are repeated twice in an alternating manner. After this, the `MixColumns (MC)` operation multiplies each column of the state by a binary matrix. Finally, a constant c_r is XORed to the state by the `AddRoundConstant (AC)` operation. Note that, for the last round, the last `ShiftRows` as well as the `MixColumns` and the `AddRoundConstant` operations are omitted, while a post-whitening key is XORed to the state. The round function $\mathcal{R}_r$ for the rounds $0 \leq r < \mathtt{r} - 1$ while also for the round $\mathtt{r} - 1$ are depicted in Figure 2.2.

1. If k is the size of the secret key, for the attack to be valid, the time complexity $\mathcal{T}$ should be smaller than $2^k C_E$.

Key Schedule The 192-bit master key of SPEEDY-r-192 is loaded to the state of the first round key k_0. To obtain the next round key, the key schedule consists in simply applying a bit-permutation PB, where the exact specification of the permutation PB can be found in [Lea+21].

Security Claims The authors made security claims for the three main versions of SPEEDY-r-192. For the 5-round version the authors expect no attack with complexity better than 2^{128} in time when data complexity is limited to 2^{64}. On the other hand, SPEEDY-6-192 should achieve 128-bit security, while SPEEDY-6-192 is expected to provide full 192-bit security.

2.1.3 Searching for Good Differential Trails

To find high-probability trails for our attack our idea was to precompute all good one-round trails and then chain them to create longer trails. A particular fact that we exploited is that for the 6-bit S-box of SPEEDY almost all 1 to 1-bit differences are possible and many of them happen with the highest possible probability for this S-box, that is 2^{-3}.

Let M be the matrix used in the MixColumns operation. In order to find good one-round trails, we first computed and stored all ordered pairs of columns $(x, M(x)) \in \mathbb{F}_2^{32} \times \mathbb{F}_2^{32}$ such that both columns x and $M(x)$ have at most 7 active bits each. This led to a total of 5248 pairs $(x, M(x)) \in \mathbb{F}_2^{32} \times \mathbb{F}_2^{32}$, that can be divided into 164 equivalence classes, each equivalence class corresponding to the 32 rotations of a different activity pattern inside a column. We then stored in a table T one representative per equivalence class and used these pairs to precompute and store all 1-round trails satisfying some particular criteria. To describe this phase we need to introduce the following notation. Let st[0] be the initial state for our computation. We denote by st[1] the resulting state after applying MC to st[0], st[2] the state after applying SB to st[1] and so on:

$$\texttt{st[0]} \xrightarrow{\texttt{MC}} \texttt{st[1]} \xrightarrow{\texttt{SB}} \texttt{st[2]} \xrightarrow{\texttt{SC}} \texttt{st[3]} \xrightarrow{\texttt{SB}} \texttt{st[4]} \xrightarrow{\texttt{SC}} \texttt{st[5]} \xrightarrow{\texttt{MC}} \texttt{st[6]}.$$

We computed all such propagations (st[0], st[6]) satisfying the following conditions:

- st[0] has a single active column $\texttt{c}_0$ such that $(\texttt{c}_0, M(\texttt{c}_0)) \in \texttt{T}$,
- st[5] has a single active column $\texttt{c}_5$ such that $(\texttt{c}_5, M(\texttt{c}_5)) \in \texttt{T}$,
- st[2] has at most two active bits per row,
- the probability of the trail (st[0], st[6]) is strictly higher than 2^{-49}.

We obtained a total number of 48923 one-round trails, which we stored. Note that each trail can be be shifted column-wise to form 32 other valid one-round trails. Our strategy does not of course guarantee to find the best trails but consists in a reasonable trade-off between optimality and efficiency.

The next step was to find a high-probability r-round trail that would at the same time lead to an optimal sieving and key-recovery. We did this by searching for a sequence of 1-round trails that could be successfully chained. This research permitted us to find the 4-round core trail depicted in red in Figure 2.3. We then extended this core trail one-round backwards (blue part of the trail) to obtain a 5-round trail of probability $2^{-170.56}$. By searching for the existence of a cluster of trails that would have the same starting and final state as the 5-round trail of Figure 2.3 (states surrounded in red) we managed to find 409 interesting trails that permitted us to slightly improve the probability of the identified 5-round trail to $2^{-169.95}$

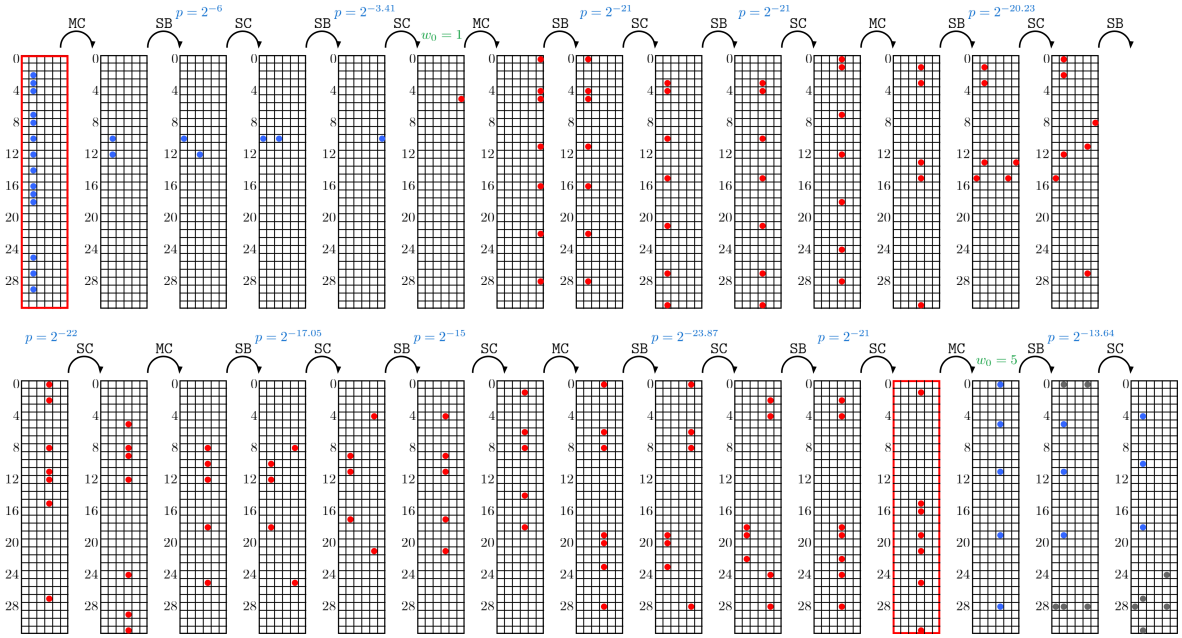


Figure 2.3 – 5.5-round differential trail used to attack `SPEEDY-7-192`. The red part corresponds to the 4-round core trail, while the blue part corresponds to the 1.5-round extension. Grey bits are bits with unknown difference. The two states surrounded in red are the starting and final states of the multiple differentials considered.

We briefly explain now how we extended this differential 0.5 round forwards (see Figure 2.3) as we followed a particular approach which impacted both the complexity of the distinguisher and the key-recovery part. More precisely, we decided to fix the transition

0x4 → 0x10 for the actives S-boxes of rows 5, 11 and 19 and to allow more transitions for the S-boxes of rows 0 and 28 and for this we computed the highest probability to have at most 4 rows active between rows between rows 23 and 31 and also row 0 after SC. We exhausted all possible configurations and we found the best one to be the one having the rows 24, 27, 28 and 31 active after SC. One possibility for this was to force the output difference of the S-box of row 0 to be of the form (0,*,0,0,*,0) and the output difference of the S-box of row 28 to be of the form (*,*,0,0,*,0), where * means that the corresponding bit is potentially active. The probability then to start from any difference of the above form in rows 0 and 28 and to activate at most the rows 24, 27, 28 and 31 after the SC is $2^{-3.41}$. This fact, together with the probability of $2^{-3.41}$ for the transition 0x4 → 0x10 for the other three active rows, gives a total probability of $2^{-13.64}$.

To summarize, as can be seen from Figure 2.3, our 5.5-round trail has then a total probability of $2^{-169.95} \times 2^{-13.64} = 2^{-183.59}$.

2.1.4 Attack on SPEEDY-7-192

Our attack on SPEEDY-7-192, the 7-round variant of the SPEEDY family of ciphers is depicted in Figure 2.4.

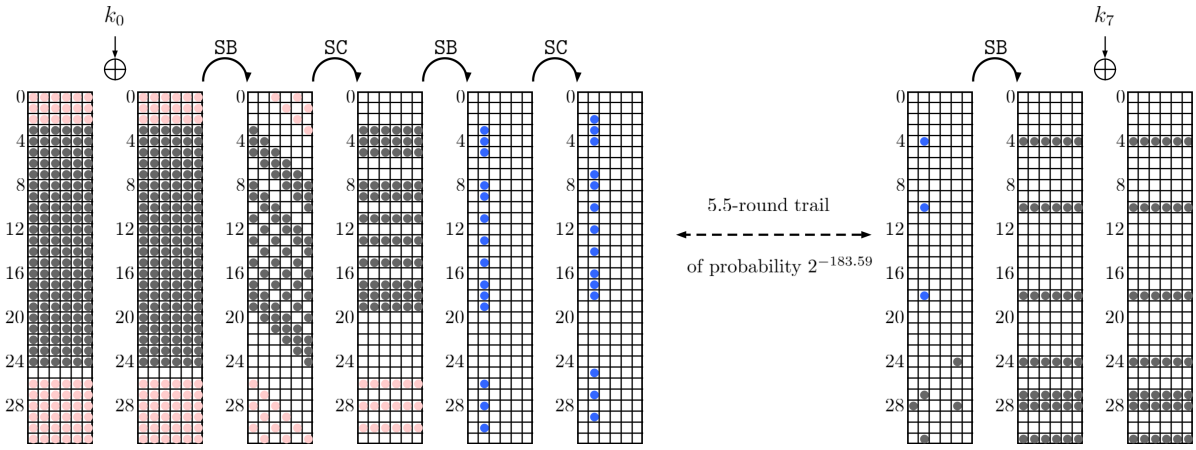


Figure 2.4 – Key recovery part of the 7-round attack against SPEEDY-7-192

A non-trivial approach we applied to permit for efficient plaintext sieving is to restrict the permitted transitions through the second S-box layer of Round 0. More precisely, the condition is that the three active bits after the second S-box in rows 26, 28 and 30 only generate a maximum of three active rows (among rows 26 to 31 and among rows 0 to 2) in the plaintext state. We computed the probability of this event to be $2^{-2.69}$. This

permitted us to have 7 non-active rows on the plaintext instead of only one if a truncated differential was used, leading to a better sieving.

Data generation and sieving We built the data required for our attack in the decryption direction. Since there are 7 active rows on the ciphertexts, the size of each structure is $2^{7\times 6} = 2^{42}$. This implies $2^s = 2^{145.28}$ structures and $2^{145.28+2\cdot 42-1} = 2^{228.28}$ potential pairs. The cost of this part is $2^{187.28}C_E$.

The sieving of the pairs is applied with regards to the differences in the plaintext and is done both on inactive rows (with a 6-bit sieving per such row) but also on the active rows by taking into account the activity pattern after each S-box application of the first `SB` step with a particular treatment of the rows `[26-31]` and `[0-2]`. The total number of pairs left after the sieving step is $2^{186.42}$.

Key recovery The key recovery algorithm is performed for each pair of data on the fly. For each pair, we check whether there exists a secret key that allows the pair to follow the differential. If not, the pair is discarded. Otherwise, we obtain a partial key on which all bits are determined but a small number n_l. For each of the remaining pairs and associated partial key, we then try exhaustively all possible 2^{n_l} keys. For each pair, the key recovery is divided into three steps which can be summarized as follows. First, we determine bits of the last subkey k_7 using the fact that if the pair follows the trail, then it must belong to δ_{out} before the last `SB` application. Since the key schedule of `SPEEDY` consists simply in a permutation of the key bits, this in turn constrains the bits of k_0. Second, we determine more bits of k_0 using the fact that the pair must belong to δ_{in}. Lastly, we determine a few extra key bits using the penultimate S-box. The details of the procedure are omitted here and can be found in [Bou+23].

The final time complexity of our attack is $2^{187.61}C_E$. The data complexity is $2^{187.28}$ and the memory complexity 2^{42}.

2.1.5 Conclusion

The key-recovery phase of differential attacks remains a very technical and tedious procedure. An interesting direction to explore would be to propose a non-trivial algorithm, and after this a tool, to optimize this step for a given cipher.

2.2 Impossible Differential Cryptanalysis

Impossible differential cryptanalysis is a powerful family of attacks against block ciphers that was independently introduced by Knudsen [Knu98] and Biham et al. [BBS99] in the late 90s. The idea of these attacks is to exploit impossible differentials, that is differentials occurring with probability zero. The general approach is then to extend the impossible differential by some rounds, possibly in both directions, guess the key bits that intervene in these rounds and check whether a trial pair is partially encrypted (or decrypted) to the impossible differential. In this case, we know that the guessed key bits are certainly wrong and we can remove the subsequent key from the space of candidate keys, i.e. the set that contains all keys that could pretend to be the right encryption key.

These attacks have successfully been applied to a high number of block ciphers following both the Feistel and the SPN construction. In some cases these attacks lead to the best cryptanalysis results for the target cipher, as this is for example the case for the multiple standardized Feistel cipher `Camellia` [LCJ11; BNS14]. Furthermore, impossible differential attacks were for a long time the most successful attacks against `AES-128` [ZWF07; Lu+08a; Mal+10], before the publication of the meet-in-the-middle attacks against this variant [DFJ13].

The key recovery step of impossible differential cryptanalysis can however be a highly technical procedure and many of the published attacks of this type were discovered to present flaws in the computation of the complexities or in some internal steps of the attack [WZF07a; ZH08; WZZ08; Lu+08b; AL13; MN12]. To simplify and help the construction and verification of these attacks, we proposed in 2014, with María Naya-Plasencia and Valentin Suder [BNS14] a complete and general complexity analysis of these attacks against Feistel ciphers. This analysis led to the detection of more flaws in previously published cryptanalyses and permitted us, by introducing or improving some key recovery techniques, to mount powerful attacks against numerous lightweight Feistel constructions. Later, together with Virginie Lallemand, María Naya-Plasencia and Valentin Suder [Bou+18] we extended this analysis to SPN ciphers, introduced more techniques and improved the time complexity formula so that it takes into account the key schedule.

2.2.1 The Proposed Framework

As most cryptanalysis techniques, an impossible differential attack is divided into two parts: a distinguisher part followed by a key-recovery step. The first one deals with the

discovery of a maximum-length impossible differential, that is an input difference $\mathcal{D}_X$ and an output difference $\mathcal{D}_Y$ such that the probability that $\mathcal{D}_X$ propagates after a certain number of rounds, r_Δ, to $\mathcal{D}_Y$ is zero. The key-recovery step, whose main part is the *key sieving*, consists in the addition of some external rounds that serve to remove the keys that would partially encrypt (resp. decrypt) data to the impossible differential. More precisely, we suppose that the impossible differential is extended r_{in} rounds backwards to obtain a difference that we denote by $\mathcal{D}_{in}$ and r_{out} rounds forwards to obtain a difference called $\mathcal{D}_{out}$. The $\log_2$ of the size of a set $\mathcal{D}$ will be denoted by Δ. This scenario is depicted in Figure 2.5.

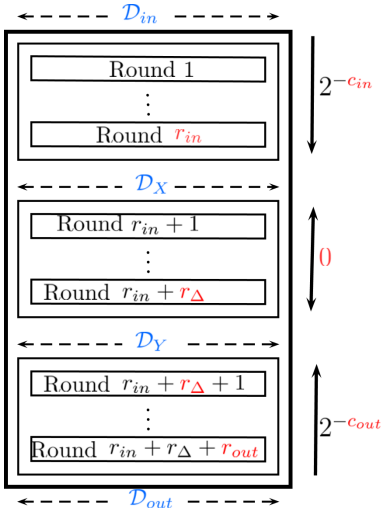


Figure 2.5 – Basic impossible differential attack

A first important quantity for the success of an impossible differential attack is the number of bit-conditions that must be satisfied during the first and the last appended rounds. For this, we denote by c_{in} (resp. c_{out}) the number of bit-conditions to be satisfied in order to get $\mathcal{D}_X$ from $\mathcal{D}_{in}$ (resp. $\mathcal{D}_Y$ from $\mathcal{D}_{out}$). The corresponding probabilities to these events are $2^{-c_{in}}$ and $2^{-c_{out}}$. Another important quantity is the total number of key bits that intervene in the $r_{in} + r_{out}$ appended rounds, called the *information key bits* of the attack. Let k_{in} the subset of subkey bits involved in the attack during the first r_{in} rounds, and k_{out} during the last r_{out} ones. We denote by $|k_{in} \cup k_{out}|$ the entropy of the all the involved key bits when taking the key schedule into account.

Data, time and memory complexities

From now on, we suppose that we are attacking a block cipher of block size n parametrized by a key K of size $|K|$. The attack starts by collecting enough pairs to discard keys that generate a difference $\mathcal{D}_X$ at the beginning of round $(r_{in}+1)$ and at the same time, a difference $\mathcal{D}_Y$ at the output of round $(r_{in}+r_\Delta)$. We need in particular enough pairs so that the number of non-discarded keys is significantly lower than the a priori total number of key candidates. To decide on the efficiency of an attack, different parameters have to be accurately estimated. We start by discussing the number of needed plaintext (or ciphertext) pairs, corresponding to the *memory complexity* of the attack.

Number of pairs required for an attack The probability that for a given key, a pair of inputs already satisfying the differences $\mathcal{D}_{in}$ and $\mathcal{D}_{out}$ satisfies all the $(c_{in}+c_{out})$ bit-conditions is $2^{-(c_{in}+c_{out})}$. In other words, this is the probability that for a correct pair of inputs a key is discarded from the set of candidate keys. Therefore, by repeating the procedure with N different input (or output) pairs, the probability that a trial key is not discarded is

$$P = (1-2^{-(c_{in}+c_{out})})^N.$$

A popular strategy for the first impossible differential attacks was to choose N such that only the right key is left after the sieving procedure. This amounts to choose P as

$$P = (1-2^{-(c_{in}+c_{out})})^N < \frac{1}{2^{|k_{in}\cup k_{out}|}},$$

leading to a potentially high value of N. However, we showed in [BNS14] that an alternative strategy for choosing N can help to reduce the number of pairs needed for the attack and to offer better trade-offs between the data and time complexity. Indeed, we suggested to equally consider smaller values of N. In this way, one will be probably left with more than one key in the set of candidate keys and will need to proceed to an exhaustive search among the remaining candidates, but the total time complexity of the attack might be much lower in the end. To find the most suitable value of N, one can start by examining values such that P is slightly smaller than $\frac{1}{2}$, reducing thus the exhaustive search by at least one bit. In this way, the starting value for N, denoted by N_{min} should be chosen such as

$$P = (1-2^{-(c_{in}+c_{out})})^{N_{min}} \approx e^{-N_{min}\times 2^{-(c_{in}+c_{out})}} < \frac{1}{2},$$

leading to $N_{min} = 2^{c_{in}+c_{out}}$. One then will have to choose some $N \geq N_{min}$. This quantity corresponds to the memory complexity of the attack.

Data complexity We also provided in [BNS14] a detailed analysis of the cost needed to construct N pairs satisfying the differential $(\mathcal{D}_{in}, \mathcal{D}_{out})$. This cost, C_N, that corresponds to the *data complexity* of the attack, and should therefore be such that $C_N \leq 2^n$ is given by

$$C_N = \max\left\{\min_{\Delta\in\{\Delta_{in},\Delta_{out}\}}\left\{\sqrt{N2^{n+1-\Delta}}\right\}, N2^{n+1-\Delta_{in}-\Delta_{out}}\right\}, \tag{2.1}$$

where Δ_{in} is the number of active bits in $\mathcal{D}_{in}$ ($\log_2$ of the dimension of the input space) and Δ_{out} is the number of active bits in $\mathcal{D}_{out}$.

Time complexity The naive approach of this attack is to test each one of the candidate keys, and for each, to check if any of the N pairs leads to the impossible differential. In this case, the tested key is discarded. This means however that this step has a potential cost of $(N2^{|k_{in}\cup k_{out}|})C'_E$, where C'_E is the ratio of the cost of partial encryption to the full encryption. In practice, to reduce the term $N2^{|k_{in}\cup k_{out}|}$, this step is always done by applying the so-called *early-abort technique* (used for example in [Lu+08b]). This technique consists in storing the needed values for the good differential transitions of each active S-box of the partial extended differential trail. As we do not test the incompatible combinations, and the proportion of tested pairs is therefore given by $\frac{1}{2^{c_{in}+c_{out}}}$, the number of partially encrypted pairs will be $2^{|k_{in}\cup k_{out}|}\frac{N}{2^{c_{in}+c_{out}}}$. The attack scenario consists now in storing the N pairs and considering, step by step, the partial possible key candidates to each of the remaining combination of the N pairs with the already guessed key bits.

The approximation of the time complexity is given by

$$C_T = \left(C_N + \left(N + 2^{|k_{in}\cup k_{out}|}\frac{N}{2^{c_{in}+c_{out}}}\right)C'_E + 2^K P\right)C_E. \tag{2.2}$$

The first term is the cost C_N, that is the amount of needed data (Formula (2.3)) for obtaining the N pairs, where N is such that $P < 1/2$. The second term corresponds to the number of candidate keys $2^{|k_{in}\cup k_{out}|}$, multiplied by the average cost of testing the pairs. Finally, the third term is the cost of the exhaustive search for the key candidates still in the set of the candidate keys after the sieving, where C_E is the cost of one encryption.

This formula is a lower-bound approximation of the time complexity as each of the terms corresponds to the minimum amount of computations needed for each step.

2.2.2 Some Improvements

Multiple impossible differentials or multiple extension paths

In [Tsu+08] and later in our work [BNS14], the idea of multiple impossible differentials was introduced and developed, showing how this technique could generically allow to reduce the data complexity. We denote by n_{in} the number of considered input differences $\mathcal{D}_X$ and by n_{out} the number of considered output differences $\mathcal{D}_Y$. These numbers correspond typically in practice to the number of rotated versions of a concrete input and a concrete output to an impossible differential. In [Bou+18] we proposed also to use, when applicable, multiple differential transitions in the appended r_{in} and r_{out} key-recovery rounds. We denote by m_{in} (resp. m_{out}) the number of possible different external input differences $\mathcal{D}_{in}$ (resp. output differences $\mathcal{D}_{out}$) for one impossible differential. Considering these new quantities, the new data complexity can be approximated by:

$$C'_N = \frac{C_N}{m_{in}m_{out}n_{in}n_{out}}. \tag{2.3}$$

The above formula is derived from the data complexity formula given in [BNS14].

Reducing the size of the partial keys to guess or state-test

In [BNS14] we proposed a new technique, called *state-test*. Its idea is to make a test for some part of the internal state instead of guessing the necessary key bits for computing this part. This technique allows then in certain cases to reduce the number of key bits to guess by s by fixing s bits of all the plaintexts to a common value. Let's see how this technique works.

Suppose x is an s-bit word of the internal state of the cipher needed to verify if a condition is satisfied in the second round. Suppose further, that we can express x as

$$x = x' \oplus M(S(p \oplus k_i) \oplus k_j, \tag{2.4}$$

where M is part of the linear layer, S is a bijective S-box and x' is a value that we suppose to know for each pair after having guessed a part of the key material. Further, suppose that p is an s-bit part of the plaintexts that is fixed for all considered pairs and let k_i and k_j be two s-bit key words that have not been guessed nor determined yet. The idea of the state-test technique is instead of guessing the $2s$ bits of (k_i, k_j) to guess the s-bit value

$x \oplus x'$ and construct a table will all possible guesses. The idea is to see if all values of $x \oplus x'$ can appear in the table. If this is the case, as p is fixed for all pairs, the only undetermined variables of the left part of Eq. (2.4) are k_i and k_j. Then, since S is a permutation, for any choice of k_i and any choice of k_j, there will always exist (at least) one pair such that $M(S(p \oplus k_i) \oplus k_j$ is in the table, leading thus to the impossible differential. As a conclusion, we know that if $x \oplus x'$ takes all the possible values in the table, we can remove the keys composed by the guessed values on which depend $x \oplus x'$ from the candidate keys set, as for all the values of (k_i, k_j), they would imply the impossible differential. If instead, $x \oplus x'$ does not take all the possible values for a certain guessed value of the partially known key, we can test this partial key combined to all the possibilities of the remaining key bits that verify Eq. (2.4) for the missing $x \oplus x'$, as they belong to the remaining key candidates. The main gain of the state-test technique is that it decreases the number of information key bits and therefore the time complexity. The time complexity in this case becomes:

$$C_T = \left(C_N + \left(N + 2^{|k_{in} \cup k_{out}| - s} \frac{N}{2^{c_{in}+c_{out}}} \right) C'_E + 2^K P \right) C_E. \tag{2.5}$$

On the key-schedule

Until now, we imagined that the key schedule was simple enough so that recovering one bit of information of k_{in} or k_{out} could be directly translated in one bit of the master key. This is not always the case, for example when analyzing the `AES`. This issue is discussed in [Der16] and we also analyzed it in [Bou+18], where we proposed to add a term to the general time complexity, that basically takes into account the cost of, from each of the $P2^{|k_{in} \cup k_{out}|}$ candidate keys for the $2^{|k_{in} \cup k_{out}|}$ external bits, computing the full candidate master keys using the key schedule (see [Bou+18] for all the details).

2.2.3 Applications and Conclusion

Our techniques permitted us to mount impossible differential attacks against both Feistel and SPN ciphers, that showed to be the best impossible differential attacks against all ciphers that we analyzed. In some cases our results improved the best attacks in general. See Table 2.1 for the results on Feistel constructions and Table 2.2 for the results on SPN ciphers.

Algorithm	# Rounds	Data	Time (CP)	Memory (Blocks)	Reference
CLEFIA-128	13	$2^{117.8}$	$2^{121.2}$	$2^{86.8}$	[MDS11]
using multiple impossible differentials	13	$2^{111.02}$	$2^{122.26}$	$2^{82.60}$	◇ our paper [BNS14]*
combining with state-test technique	13	$2^{114.58}$	$2^{116.16}$	$2^{83.16}$	◇ our paper [BNS14]*
	13	$2^{114.4}$	$2^{114.4}$	2^{80}	our paper [Bou+18]
Camellia-128	11	2^{122}	2^{122}	2^{98}	[LCJ11]
	11	$2^{118.4}$	$2^{118.43}$	$2^{92.4}$	our paper [BNS14]*
Camellia-192	12	2^{123}	$2^{187.2}$	$2^{155.41}$	[LCJ11]
	12	$2^{119.7}$	$2^{161.06}$	$2^{150.7}$	our paper [BNS14]*
Camellia-256	13	2^{123}	$2^{251.1}$	2^{203}	[LCJ11]
	13	$2^{225.06}$	$2^{119.71}$	$2^{198.71}$	our paper [BNS14]*
Camellia-256	14	2^{120}	$2^{250.5}$	2^{120}	[LCJ11]
(without whitening keys	14	2^{118}	2^{220}	2^{173}	◇ our paper [BNS14]
and FL layers)	14	$2^{117.7}$	$2^{215.7}$	$2^{166.7}$	our paper [Bou+18]
LBlock	22	2^{58}	$2^{79.28}$	$2^{72.67}$	[KDH12]
	22	2^{60}	$2^{71.53}$	2^{59}	our paper [BNS14]*,[Bou+14]
	23	2^{59}	$2^{75.36}$	2^{74}	our paper [BNS14]*
	23	$2^{55.5}$	2^{72}	2^{65}	our paper [Bou+18]*
Simon32/64	19	2^{32}	$2^{62.56}$	2^{44}	our paper [BNS14]*
Simon48/72	20	2^{48}	$2^{70.69}$	2^{58}	our paper [BNS14]*
Simon48/96	21	2^{48}	$2^{94.73}$	2^{70}	our paper [BNS14]*
Simon64/96	21	2^{64}	$2^{94.56}$	2^{60}	our paper [BNS14]
Simon64/128	22	2^{64}	$2^{126.56}$	2^{75}	our paper [BNS14]
Simon96/96	24	2^{94}	$2^{94.62}$	2^{61}	our paper [BNS14]
Simon96/144	25	2^{128}	$2^{190.56}$	2^{77}	our paper [BNS14]
Simon128/128	27	2^{94}	$2^{126.6}$	2^{61}	our paper [BNS14]
Simon128/192	28	2^{128}	$2^{190.56}$	2^{77}	our paper [BNS14]
Simon128/256	30	2^{128}	$2^{254.68}$	2^{111}	our paper [BNS14]

Table 2.1 – Summary of the best impossible differential attacks on CLEFIA-128, Camellia, LBlock and Simon. The presence of a '*' mentions if the current attack was the best known attack against the target cipher at the moment our papers were published. Note here that we provide only the best of our results with respect to the time complexity. Other trade-offs are possible. ◇ Incorrect result not taking into account the key-schedule.

Algorithm	Rounds	Data (**CP**)	Time	Memory (**Blocks**)	Technique	Ref.
AES-128	7	$2^{106.2}$	$2^{110.2}$	$2^{90.2}$	ID	[Mal+10]
	7	2^{105}	$2^{105} + 2^{99}$	2^{90}	MITM	[DFJ13]
	7	2^{97}	2^{99}	2^{98}	MITM	[DFJ13]
	7	2^{121}	$2^{121} + 2^{83}$	2^{74}	MITM §	[DF13]
	7	2^{113}	$2^{113} + 2^{75}$	2^{82}	MITM §	[DF13]
	7	$2^{113.1}$	$2^{113.1} + 2^{105.1}$	$2^{74.1}$	ID	our paper [Bou+18]
	7	2^{105}	$2^{106.88}$	2^{74}	ID	our paper [Bou+18]
CRYPTON-128	7	2^{97}	$2^{97.2}$	2^{100}	Trunc. Diff.	[Kim+04]
	7	2^{121}	$2^{121} + 2^{116.2}$	2^{119} †	ID	[MSD10]
	7	$2^{114.92}$	$2^{114.92} + 2^{113.7}$	$2^{88.5}$	ID	our paper [Bou+18]
	8	2^{126}	$2^{126.2}$	2^{100}	Trunc. Diff.	[Kim+04]
ARIA-128	6	2^{113}	$2^{121.6}$	2^{113}*	ID	[Li+08]
	6	2^{121}	$2^{121} + 2^{112}$	2^{121}*	ID	[WZF07b]
	6	$2^{120.5}$	$2^{120.5} + 2^{104.5}$	2^{121}*	ID	[Li+08]
	6	2^{120}	$2^{120} + 2^{96}$	2^{120}*	ID	[LS08]
	6	2^{111}	$2^{111} + 2^{82}$	2^{71}	ID	our paper [Bou+18]
	7	$2^{105.8}$	$2^{105.8} + 2^{100.99}$	$2^{79.73}$	LC	[Liu+11]

Table 2.2 – Summary of the best single-key attacks against AES-128, CRYPTON-128 and ARIA-128 at the moment our article [Bou+18] was published. * Estimated memory requirements since not given in the original papers. † Complexity estimated in [Mal14]. § Additional trade-offs of the attacks in [DF13] provided by P. Derbez (private communication).

.

One of the main results of our work on impossible differential cryptanalysis is the proposal of generic complexity formulas and of complexity formulas in case some particular techniques are used. To show that our formulas are indeed good estimates of the real complexities, we implemented in [Bou+18] the state-test technique and the use of multiple (impossible) differentials on toy examples. In general, it has to be noted that our generic time complexity formula is a lower-bound approximation of the time complexity and to determine the exact complexity a detailed analysis of the attack steps is necessary. This approximation is most of the times very close to the real complexity but counter-examples

exist, as shown for example by Derbez in [Der16].

2.3 Differential Meet-In-The-Middle Attacks

Differential cryptanalysis [BS90] is among the most well-studied cryptanalysis methods against symmetric primitives. We demonstrated their power in Section 2.1.1 against the block cipher `SPEEDY`. Another very old and popular technique that has been useful in many cryptanalysis applications, and the subject of a large number of improvements and further studies is meet-in-the-middle (MITM) cryptanalysis [DH77]. The idea of basic MITM attacks is to split the cipher into two parts, where each part can be computed with partial knowledge of the key. An attacker can then validate partial key guesses by checking for a match in the middle. Many extensions and refinements of the basic attack exist today [DSP07; IS12; AS08; AS09; Guo+10; CNV13; BKR11].

In a very recent paper, we introduced together with Nicolas David, Patrick Derbez, Gregor Leander and María Naya-Plasencia a new cryptanalysis technique that we named the *differential meet-in-the-middle attack* [Bou+22]. The idea of this new technique is to use a differential to cover several middle rounds of the cipher while running a meet-in-the-middle attack on its external rounds. This technique permitted us to improve by 2 rounds the previous best attacks against `SKINNY-128-384`, a popular and well-analyzed cipher.

2.3.1 The New Attack Framework

Our differential meet-in-the-middle attack can be seen as a new cryptanalysis technique against symmetric primitives. This new attack aims at combining meet-in-the-middle (MITM) attacks together with differential cryptanalysis. The main motivation of our work was to investigate whether there exists a method for reaching more rounds than the sieve-in-the-middle attack [CNV13], an extension of classical MITM attacks. However, our technique can also be interpreted as a new key-recovery method to apply in differential cryptanalysis. We start by providing a high-level description of the new technique. More precisely, we will provide a general framework that describes how to mount a differential MITM attack in a generic and simple way, and we will show how to combine this generic method with two techniques: the parallel treatment of data partitions in order to add one round mostly for free, as well as a technique to reduce the data complexity.

General framework

Consider an n-bit cipher E parametrized by a secret key k and decomposed into three sub-ciphers: $E_{out} \circ E_m \circ E_{in}$, as depicted in Figure 2.6. Let the number of rounds of E_{in}, E_m and E_{out} be r_{in}, r_m and r_{out} respectively. Finally, let Δ_x be the input difference to the middle part E_m, Δ_y the output difference of E_m and suppose that the differential $\Delta_x \to \Delta_y$, covering the r_m middle rounds, has probability 2^{-p}.

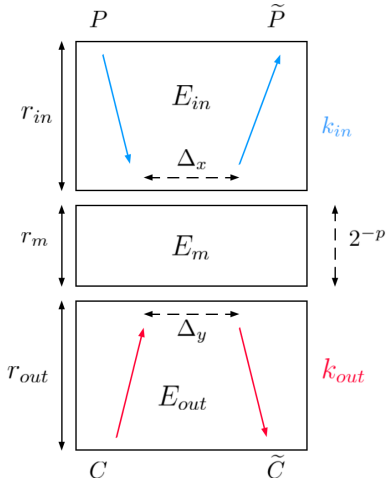


Figure 2.6 – A high-level description of the Differential MITM technique

We start our analysis with a first randomly chosen plaintext P and its associated ciphertext C, and we aim at generating a second plaintext-ciphertext pair $(\widetilde{P}, \widetilde{C})$ such that together they satisfy the differential on the middle rounds. Our new idea is to generate $(\widetilde{P}, \widetilde{C})$ with a meet-in-the-middle approach. For this, candidate plaintexts $\widetilde{P}$ are computed from both the plaintext P and the difference Δ_x while candidate ciphertexts $\widetilde{C}$ are computed from C and Δ_y. The match is then performed on the relation $E(\widetilde{P}) = \widetilde{C}$ (or $\widetilde{P} = E^{-1}(\widetilde{C})$).

Note that the roles of the upper and lower part can be interchanged without loss of generality in order to optimize the data and memory complexity, if we consider that the access to both the encryption and the decryption oracles is granted.

Upper part. Given P, the aim is to guess the minimal amount of key information, that we will denote by k_{in}, such that we can compute the associated $\widetilde{P}$ that ensures $E_{in}(P) \oplus E_{in}(\widetilde{P}) = \Delta_x$ if the guess of k_{in} corresponds to the secret key. For each guess i for k_{in}, we obtain a different candidate for $\widetilde{P}$, that we denote by $\widetilde{P}^i$, leading to a total of

$2^{|k_{in}|}$ such values. From them, we can compute the $2^{|k_{in}|}$ associated ciphertexts $\widehat{C}^i = E(\widetilde{P}^i)$ with calls to the encryption oracle and store them in a hash table H.

Lower part. Similarly, given C, we can guess some key material k_{out}, of length $|k_{out}|$, and compute a new ciphertext $\widetilde{C}$ that satisfies the equation $E_{out}^{-1}(C) \oplus E_{out}^{-1}(\widetilde{C}) = \Delta_y$ if the key guess is correct. We obtain $2^{|k_{out}|}$ values for $\widetilde{C}^j$, each associated to a guess j for k_{out}.

This procedure is summarized in Algorithm 1.

Number of pairs and match. For the correct key guess, the transition $\Delta_x \to \Delta_y$ will happen with a probability 2^{-p}. Therefore, we will repeat the upper and lower procedures 2^p times with 2^p different messages P_ℓ so that we can expect one pair $(P_\ell, \widetilde{P}_\ell^i)$ to satisfy the differential together with the associated pair $(C_\ell, \widetilde{C}_\ell^j)$. When this is the case, we will find a collision for a certain ℓ between a $\widehat{C}_\ell^i$ computed in the upper part and stored in H and a $\widetilde{C}_\ell^j$ computed from the lower part. Each collision (i, j) has an associated key guess $k_{in} = i, k_{out} = j$, that we will consider as a potential candidate. The number of expected collisions for each fixed P_ℓ is $2^{|k_{in}|+|k_{out}|-|k_{in}\cap k_{out}|-n}$.

Algorithm 1 Differential MITM attack

```
while right key not found do                          ▷ 2^p trials expected
    Randomly pick P
    C ← E(P)                                          ▷ Oracle call
    H ← ∅                                             ▷ hash table initialisation
    for each guess i for k_in do                      ▷ Forward computation
        Compute P̃^i from i and P
        Ĉ^i ← E(P̃^i)                                  ▷ Oracle call
        H[Ĉ^i] ← H[Ĉ^i] ∪ {i}
    for each guess j for k_out do                     ▷ Backward computation
        Compute C̃^j from j and C
        for each i ∈ H[C̃^j] do
            Complete (i, j) to retrieve the master key
            Try candidates against extra data
```

Complexity. The first term of our time complexity formula corresponds to the computations done in E_{in} and E_{out} and is thus $2^p \times (2^{|k_{in}|} + 2^{|k_{out}|})$. The second term is related to the number of expected candidates $2^{|k_{in}\cup k_{out}|-n+p}$. Two cases must be considered here.

First, if we expect fewer key candidates than the whole set $\{k_{in} \cup k_{out}\}$, which holds as long as $p < n$, we can guess the remaining bits of the master key and test the guess with additional pairs. In this case the term $2^{k-(|k_{in} \cup k_{out}|)}$ should be added. Otherwise, if the number of expected candidates is higher or equal to $|k_{in} \cup k_{out}|$ then the term 2^{k-n+p} should be added to retrieve the master key. To summarize, the time complexity formula is given by

$$\mathcal{T} = 2^p \times (2^{|k_{in}|} + 2^{|k_{out}|}) + 2^{|k_{in} \cup k_{out}|-n+p} + 2^{k-|k_{in} \cup k_{out}|} \times \max\{1, 2^{|k_{in} \cup k_{out}|-n+p}\}.$$

The (naive) data complexity of this first version of the attack can be estimated as

$$\mathcal{D} = \min(2^n, 2^{p+\min(|k_{in}|,|k_{out}|)}).$$

Finally, the naive memory complexity is given by $\mathcal{M} = 2^{\min(|k_{in}|,|k_{out}|)}$, though it can be improved to $2^{\min(|k_{in}|-|k_{in} \cap k_{out}|,k_{out}-|k_{in} \cap k_{out}|)}$ by first guessing the common key material before running the attack.

2.3.2 Improvement Techniques

We propose now several improvements to the above the basic attack.

Parallel Partitions for Layers with Partial Subkeys

We showed that in the case where the round key addition does not affect the whole n-bit state but only $m < n$ bits of it (as is for instance the case in Feistel constructions [Fei73], or in the `SKINNY` [Bei+16] and `GIFT` [Ban+17] ciphers), one additional round can be added to the attack. Moreover, if $p > m$, the addition of this round doesn't add anything to the time complexity of the attack. On the other hand, the memory complexity will be a priori increased, while the data complexity should be checked case-by-case, as it might depend on the configuration of the differences in the external states.

To explain this technique, suppose that S_{r-1} is the final state of the differential MITM attack. Denote by X the state after appending 1 round to S_{r-1}, except for the final application of the last subkey K_r, and let $Y = X \oplus K_r$ (see Figure 2.7). Suppose here that the subkey K_r only applies to m bits of the state. The part of X affected by the key addition, will take 2^m possible values and for each of them we can compute the state S_{r-1}. From this state, we will guess the k_{out} bits, in order to compute a state $\widetilde{S_{r-1}}$ such that

$E_{out}(S_{r-1}) \oplus E_{out}(\widetilde{S_{r-1}}) = \Delta_y$. This procedure will yield $2^{|k_{out}|+m}$ candidates to match. In parallel, the state Y after the key addition will also take all the 2^m possible values, and for each of them we decrypt in order to obtain the plaintext, and do the upper key guessing procedure to deduce the good pairs, obtaining $2^{|k_{in}|+m}$ candidates to match.

The number of candidates to match might seem higher now by a factor of 2^{2m}, however the filter is much more important also and permits in general to entirely amortize the cost. Indeed, we have to match X and Y, as well as their associated states X' (obtained from $\widetilde{S_{r-1}}$) and $Y' = X' \oplus K_r$, and these associated states must satisfy $X \oplus X' = Y \oplus Y'$. This adds m bit-conditions. However, extra m bit-conditions can be added if K_r depends linearly on k_{in} and k_{out} or more generally if it can be written as $K_r = f(k_{in}) \oplus g(k_{out})$, which is often the case. As $2^{2m}2^{-m}2^{-m} = 1$, the cost, given by the number of solutions, stays exactly the same as the attack with one round less.

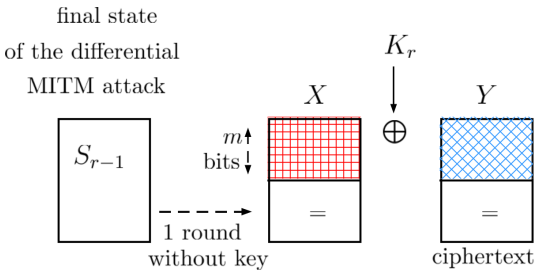


Figure 2.7 – Partial guess of the final state to add one round for free. S_{r-1} is the final state of the simple differential MITM attack.

Reducing data with imposed conditions

We explain here a way to obtain time-data-memory trade-offs for the original attack. If when choosing the plaintext P, we force x of its bits, that might have been active otherwise, to a certain value, and if we expect the same from the associated plaintext $\widetilde{P}$, the overall probability of the attack will decrease to 2^{-p-x}, as we will have to repeat the procedure until a $\widetilde{P}$ that satisfies this constraint is found. More precisely, if $\widetilde{P}$ does not fit this condition, the corresponding tuple will not be stored in the hash table since we do not have access to its ciphertext. However by doing so, the data as well as the memory complexity will be reduced by a factor of 2^x. When combining this technique with the previous one, we can derive the following two inequalities for x:

$$p + x \leq n - x \quad \text{and} \quad 2^{p+x}(2^{|k_{in}|} + 2^{|k_{out}|}) < 2^k.$$

Data reduction without time increase As the total number of candidates for the key of the input part (respectively output) will be $2^{|k_{in}|-x}$ (respectively $2^{|k_{out}|-x}$), if we are able to find these candidates with their associated $\widetilde{P}$ (respectively $\widetilde{C}$) in a complexity given by the number of solutions, the time complexity would become:

$$2^{p+x}(2^{|k_{in}|-x} + 2^{|k_{out}|-x}) = 2^p(2^{|k_{in}|} + 2^{|k_{out}|}),$$

which allows us to reduce the data complexity to $\max(2^{p+x}, 2^{n-x})$ while not increasing the time complexity. The optimal data complexity in this case will be given when $x \approx \frac{n-p}{2}$, and be around $2^{\frac{n+p}{2}}$.

This can actually be done in many cases using rebound-like techniques [Men+09]. This is the case of all of our attacks summarized in Table 2.3, for which we considered the highest possible x to compute the given complexities.

As argued before, our new cryptanalysis technique is closely related to two families of cryptanalysis: MITM attacks and differential attacks. In this section we will discuss similarities and differences between these families and will try to identify cases where our new technique might be efficient or cases where it permits to reach better results compared to the best known attacks.

2.3.3 Relation to MITM and Differential Attacks

Our new cryptanalysis technique is closely related to two families of attacks: MITM and differential attacks. We briefly describe here similarities and differences between these attacks and our newly introduced technique.

Our new technique with the addition of the middle differential has the potential of reaching more rounds than a MITM or a sieve-in-the-middle attack [CNV13]. The data complexity could however be higher than in a classical MITM attack as now we compute a new $\widetilde{P_\ell^i}$ for each guess i of the key and for each one of the 2^p different plaintexts P_ℓ (needed in order to hope for one that will satisfy the middle differential). On the other hand, despite the fact that the sets of bits k_{in} or k_{out} involved in the parallel computations of the differential MITM attack are not determined in the same way as the key bits involved in MITM attacks, we expect those quantities to be relatively close under similar settings, as this principally depends on the propagation properties of the round function. More precisely, it seems that more aligned [Bor+21] the round function is, closer the sets will be. Therefore, we expect that ciphers where classical MITM attacks work well, can also

be interesting targets for differential MITM attacks.

Curiously enough, our new attack can also be seen as a new way of performing the key-recovery part associated to a differential distinguisher. The time complexity of a typical differential attack can be estimated as (see also Section 2.1.1)

$$2^{p+1} + 2^{p-s+1}2^{2s-1}2^{-c}C_k \approx 2^{p+s-n+a},$$

where the first term corresponds to the number of plaintexts to generate, 2^{p-s+1} is the number of needed structures of size 2^s, $2^{-c} = 2^{-n-a}$ corresponds to the filter, with a being the bit-size of the active part in the ciphertexts and C_k the average cost of determining the key bits for each candidate pair.

We can see that if $a + s >> n$, which can happen when several rounds are appended, the complexity of a differential MITM attack for an equivalent number of rounds might be more interesting. Indeed, the influence of the input and output extensions to the complexity are added and not multiplied. In particular, our attack can become much more efficient when the key size of the cipher is bigger than the state size, otherwise 2^p might already be close to the limit.

2.3.4 Application to SKINNY-128-384

We applied our new technique to SKINNY-128-384 and were able to break 23, 24 and 25 rounds of this cipher. The best previous attack against this variant was a MITM attack [Don+21] that reached 23 rounds. The best single-key attacks against SKINNY-128-384 are summarized in Table 2.3.

Our 24-round attack is based on a 15-round truncated differential. By using the constrained programming Choco-solver [PFL16], and the method developed in [Del+21] we checked that this differential could be successfully instantiated and by taking clustering into account, we could check that its probability is at least $2^{-116.5}$. We appended 4 rounds on the top and 5 rounds on the bottom to obtain a 24-round differential MITM attack by following the description of the attack given in Algorithm 1 and the improvements explained in Section 2.3.2. This attack could then be extended by one more round for free by using the technique explained in Section 2.3.2.

Table 2.3 – Best attacks against `SKINNY-128-384` in the single key (SK) model together with the results presented in this paper. ID stands for impossible differentials, MITM for meet-in-the-middle attacks and DS-MITM for Demirci-Selçuk-type MITM.

# Rounds	Data	Time	Memory	Type	Ref.
21	2^{123}	$2^{353.6}$	2^{341}	ID	[YQC17]
21	$2^{122.89}$	$2^{347.35}$	2^{336}	ID	[HSE22]
22	2^{96}	$2^{382.46}$	$2^{330.99}$	DS-MITM	[Shi+18]
22	$2^{92.22}$	$2^{373.48}$	$2^{147.22}$	ID	[TAY17]
23	2^{104}	2^{376}	2^{8}	MITM	[Don+21]
23	$2^{116.95}$	$2^{361.9}$	$2^{118.45}$	Diff-MITM	our paper [Bou+22]
24	$2^{116.95}$	$2^{361.9}$	$2^{182.95}$	Diff-MITM	our paper [Bou+22]
24	$2^{122.25}$	$2^{372.5}$	$2^{123.75}$	Diff-MITM	our paper [Bou+22]
25	$2^{122.25}$	$2^{372.5}$	$2^{188.25}$	Diff-MITM	our paper [Bou+22]

2.3.5 Conclusion

The introduction of our new technique releases naturally numerous questions and opens many new research directions. First, we would like to further understand the link between the new attack and classical MITM attacks and how these two attacks can be compared. For example, we would like to identify for what kind of primitives the quantity of the involved key material in the differential MITM attack would be typically smaller compared to a classical MITM attack applied to the same cipher in a similar setting.

As MITM attacks combine particularly well with the technique of bicliques, another natural question is whether differential MITM attacks combine well with bicliques as well. Furthermore, is it possible to find any concrete application where the combination of a differential MITM with bicliques could improve previous MITM or other attacks? Finally, can the technique of bicliques be combined with the method of partitions we proposed in the case of partial subkey additions, and how do they compare?

A last open question is whether instead of combining MITM techniques with differential attacks, one could successfully combine MITM with some other well-known family of cryptanalysis, such as for example linear or differential-linear attacks.

2.4 New MILP Modelings for SPN Ciphers

In symmetric-key cryptography, a popular technique introduced by Mouha et al. [Mou+11] and by Wu and Wang [WW11] for proving resistance against classical attacks is to model the behaviour of the cipher as a Mixed Integer Linear Programming (MILP) problem and solve it by some MILP solver. The use of MILP by designers and cryptanalysts saw a spectacular development over the last years.

For Substitution Permutation Networks (SPN), it is possible to get easy and relatively small models when searching for properties at the word level, however one needs much more constraints to obtain accurate models at the bit level. Sun et al. [Sun+14a; Sun+14b] were the first to propose bit-oriented modelings for SPN ciphers. A non-trivial problem in doing so is to find an efficient representation of the valid differential propagations through an S-box. Indeed, S-boxes are non-linear Boolean vectorial functions, therefore modeling their differential properties with $\mathbb{R}$-linear inequalities is not natural. Several approaches have been suggested to solve this problem, as the convex hull and the logical condition modelling proposed in [Sun+14b; Sun+14a]. The main issue with the above approaches is that they do not adapt well to large (e.g. 8-bit) S-boxes. To tackle this problem, Abdelkhalek et al. [Abd+17] observed that generating a minimal number of constraints in logical condition modeling can be converted into the problem of minimizing the product-of-sum representation of Boolean functions. This last problem is well-studied and algorithms for solving it exist, for example the *Quine-McCluskey* (QM) [Qui52; Qui55; McC56] or the *Espresso* [Bra+84] algorithms. In this way, Abdelkhalek et al. managed for the first time to generate linear constraints for 8-bit S-boxes, notably for the S-boxes of `AES` [NIS01] and `SKINNY`-128 [Bei+16]. While the number of linear constraints for the S-box of `SKINNY` provided in [Abd+17] is as low as 372, the same method yields 8302 linear inequalities for the S-box of `AES`, a modeling that is often too heavy to be used in practice.

Efficiently representing the S-boxes is a crucial part of the modeling process. But, a bad modeling of the diffusion layer can render the optimization process very slow or even impractical. Indeed, with the exception of some ciphers, e.g. `Present` [Bog+07], where the linear layer is just a bit-permutation, the diffusion is usually ensured by `XOR` gates. Yet, the `XOR` operation, while linear in $\mathbb{F}_2$, models very badly in $\mathbb{R}$. So, bitwise modelings of heavy linear layers, that need many `XOR`s to be represented, can lead to impractical systems with many linear inequalities or with many dummy variables.

In a joint work with Daniel Coggia, published at ToSC [BC20], we proposed several new bitwise MILP modelings for the propagation of differential properties through both S-boxes and linear layers. Our methods permitted for the first time to model efficiently 8-bit S-boxes with dense DDTs, as the one of the `AES`. However, it has to be noted, that our new techniques for modeling the DDT of an S-box are general enough for modeling an LAT or any Boolean function in general.

2.4.1 MILP Modeling for Boolean Functions and S-boxes

The general problem we addressed in the first part of our work was the following: Given the truth table of a Boolean function f of m variables, how can one efficiently model the constraint $f(x) = 1$ by a system of $\mathbb{R}$-linear inequalities. This general problem can then be divided into two sub-problems:

Problem 1 How to generate a (possibly large) set of inequalities on variables $x_0, \ldots, x_{m-1}$ that correctly models f?

Problem 2 How to choose a (typically much smaller) subset of this set of inequalities that still correctly represents f but leads to more efficient MILP models?

For differential cryptanalysis, the above general problem corresponds to modeling the fact that $(x_0, \ldots, x_{n-1}) \to (x_n, \ldots, x_{2n-1})$ is a possible transition in a DDT.

In our work we mainly tackled Problem 1. We proposed for this problem three different heuristic methods described in the remaining of this section.

Convex hull techniques for up to 6-bit S-boxes

Sun et al. proposed in [Sun+14a; Sun+14b] the convex hull method to model the possible differential propagations through an S-box. This method consists in computing the H-representation of the convex hull of all possible points $a \in \mathbb{F}_2^m$ such that $f(a) = 1$ seen as vectors of $\mathbb{R}^m$. Taking then the $(m-1)$-dimensional faces of the convex hull yields a correct set of inequalities. The H-representation can be for example computed through an algebra computer system such as `Sage` [The20] and gives a system of linear inequalities excluding all impossible points (e.g. all points a such that $f(a) = 0$).

What we showed is that it is possible to compute many other, potentially better linear inequalities from this initial set by simply adding up some of them. Indeed, if a possible differential transition $z = (x, y) \in \{0, 1\}^m$, satisfies the k inequalities $C_1, \ldots, C_k$:

$c_0^\ell z_0 + \cdots + c_{m-1}^\ell z_{m-1} + b_\ell \geq 0$, with $\ell \in [1, k]$ then it obviously also satisfies the inequality

$$\Big(\sum_{i=1}^{k} c_0^i\Big) z_0 + \cdots + \Big(\sum_{i=1}^{k} c_{m-1}^i\Big) z_{m-1} + \sum_{i=1}^{k} b_i \geq 0$$

produced by simply summing up the k inequalities and denoted as $C_{new} = C_1 + \ldots + C_k$.

Of course, most of the inequalities produced by randomly summing k inequalities from the H-representation of the convex hull, do not present any interest, as they will very probably be satisfied by the whole space $\{0,1\}^m$. In order to produce meaningful new linear inequalities from the H-representation of the convex hull, we noticed that if k hyperplanes of the H-representation share a vertex on the cube $\{0,1\}^m$, (i.e. a possible transition), then the addition of the k corresponding inequalities will probably yield an interesting new constraint, given that its hyperplane intersects with the cube at least on this particular vertex. By "interesting" we mean here that the new inequality C_{new} will remove a different (potentially larger) set of impossible transitions than the original inequalities.

The algorithm we proposed is rather fast for 4-bit S-boxes — a few minutes for $k = 2$ and a few hours at most for $k = 3$ — and the set of constraints obtained that way does not have too many elements, which allows fast minimization in solving *Problem 2*. We applied our algorithm to solving *Problem 1* for different S-boxes and decreased by several inequalities all previous modelizations.

Unfortunately, for larger S-boxes, and notably for $n = 8$ computing the convex hull is computationally hard. For this reason, we proposed in parallel alternative methods that can be used for modeling 8-bit S-boxes.

Logical condition techniques for 8-bit S-boxes

We showed next that one can easily derive simple inequalities to remove spaces of a particular form. Let $f : \mathbb{F}_2^m \to \mathbb{F}_2$. For some $u = (u_0, u_1, \ldots, u_{m-1}) \in \mathbb{F}_2^m$ we denote by $\text{supp}(u) = \Big\{ i \Big| u_i = 1 \Big\} \subseteq [0, m-1]$. Furthermore $\text{Prec}(u)$ denotes the space $\Big\{ x \in \mathbb{F}_2^m \Big| x \preceq u \Big\}$, where $x \preceq u$ means that $x_i \leq u_i$ for all $i \in [0, m-1]$. The second heuristic method that we provided is based on the above proposition that states that spaces $a \oplus \text{Prec}(u)$ can be removed by a single inequality.

Proposition 1. *Let $a \in \mathbb{F}_2^m$ and $u \in \mathbb{F}_2^m$ such that $\text{supp}(a) \bigcap \text{supp}(u) = \emptyset$ and let $I =$*

$[0, m-1] \setminus \Big(\text{supp}(a) \bigcup \text{supp}(u)\Big)$. *Then, for all* $x \in \mathbb{F}_2^m$,

$$-\sum_{i \in \text{supp}(a)} x_i + \sum_{i \in I} x_i \geq 1 - \text{wt}(a) \Leftrightarrow x \notin a \oplus \text{Prec}(u).$$

The interest of this second method is that if there exist big spaces $a \oplus \text{Prec}(u) \subseteq \mathbb{F}_2^m$ (i.e. implying that the Hamming weight of u is high) where all points x inside are such that $f(x) = 0$, then it is possible to remove all these points at once with a single inequality.

We showed next that our proposed method is strongly related to the Quine-McCluskey (QM) algorithm [Qui52; Qui55; McC56]. Indeed, we demonstrated that the approach of this section is identical to the first step of the QM algorithm despite the fact that the QM algorithm had never been presented before from such an algebraic point of view. The important thing for us is that we principally need this first step to find good modelings. This corresponds to solving *Problem 1*. The second step of the QM algorithm, corresponds to providing a solution for *Problem 2*, by minimizing the number of terms with the objective of finding a good circuit for a Boolean function, but not necessarily a good MILP modeling. Moreover, this second step is computationally harder than the first one and acts as a bottleneck when using QM as a black box inequality generator for MILP modelings. Indeed, it is much faster to use our method alone for solving *Problem 1* together with a greedy or a MILP-based algorithm for solving *Problem 2*. This is not only faster but can also provide a significantly lower number of inequalities.

We applied our method to the 8-bit S-boxes of `SKINNY`-128 and the `AES`. While for `SKINNY`-128 this method did not improve upon the number of inequalities already given in [Abd+17], for the `AES` our method gave 7461 inequalities, contrary to the 8302 inequalities found [Abd+17] with the Espresso algorithm. Still this number is quite high for practical applications, thus we searched for alternative methods for modeling large S-boxes that could outperform in most of the cases the methods provided up to now.

Modeling an S-box with inequalities issued from balls $\mathcal{B}(d, c)$

With the third heuristic method we proposed, we showed that points lying in a ball of radius d centred at a point $c \in \mathbb{F}_2^m : \mathcal{B}(d, c) = \Big\{ x \in \mathbb{F}_2^m \Big| \text{wt}(x \oplus c) \leq d \Big\}$, can be removed together by a simple inequality.

This is illustrated in the following proposition:

Proposition 2. *Let $c \in \mathbb{F}_2^m$. Then, the inequality*

$$\sum_{i=0}^{m-1} (1 - c_i)x_i + c_i(1 - x_i) \geq d + 1, \tag{2.6}$$

holds if and only if $x \notin \mathcal{B}(d, c)$.

When searching for inequalities removing impossible transitions for a DDT, we have to be sure that the corresponding ball does not contain any possible transitions that we would mistakenly remove. In S-boxes used in practice, removing entire balls does not usually work. We then showed that it is still possible to extract an inequality from a ball from which we have removed from its edge all points $x \in \mathbb{F}_2^m$ such that $f(x) = 1$. We called such balls *distorted*. Finally, we provided an even more powerful method that permits us to remove with a single linear constraint points belonging to the union of three different (distorted) balls of radius 1. This method consisted in combining inequalities of distorted balls of radius 1.

We applied our algorithms to the S-boxes of `SKINNY`-128 and the `AES`. The results can be visualized in Table 2.4.

	# Inequalities	
S-box	[Abd+17]	Our techniques
`SKINNY`-128	372	**302**
`AES`	8302	**2882**

Table 2.4 – Number of inequalities to model the corresponding S-boxes, where the set of initial inequalities was generated by three or our different methods.

2.4.2 Linear Layer Modeling

In our work, we also tackled the problem of how to efficiently model a linear layer for MILP. Modeling a linear layer is often related to how the `XOR` operation is modeled. We first showed that modeling this operation is extremely difficult. Indeed, we demonstrated that the equation $x_0 \oplus x_1 \oplus \ldots \oplus x_{n-1} = 0$ needs at least 2^{n-1} $\mathbb{R}$-linear inequalities to characterise the set of its solutions in $\mathbb{F}_2^n$.

The idea of the two algorithms that we proposed was to minimize the number of `XOR`s needed to represent the linear layer. First, note that modeling a matrix M means modeling

the kernel of $A = (M|I)$, where I is the identity matrix. Indeed, for any matrix M with entries in $\mathbb{F}_2$,

$$Mx = y \Leftrightarrow Mx \oplus y = 0 \Leftrightarrow A \begin{pmatrix} x \\ y \end{pmatrix} = 0.$$

Since it is known that for any invertible matrix $P \in \mathrm{GL}_n(\mathbb{F}_2), \mathrm{Ker}(P \cdot A) = \mathrm{Ker}\, A$, the idea is to find a matrix $P \in \mathrm{GL}_n(\mathbb{F}_2)$ such that the rows of $P \cdot A$ have minimum Hamming weight and induce thus a minimal number of `XOR` operations. The idea of our first algorithm was to initialize P to the identity matrix and to proceed in a row-wise manner by searching at each step to replace the current row with a better one. To start with, our algorithm searches to replace the first row of A with a codeword of the form

$$m \cdot A,\ m \in \left\{ x \in \mathbb{F}_2^n \,\middle|\, x_1 = 1 \right\} \text{ and } \mathrm{wt}(m \cdot A) < \mathrm{wt}(A_{1,\star}).$$

After this first step, the first row of the matrix P is updated with the vector $(1, m_2, \cdots, m_n)$. The algorithm then searches for a replacement for the other rows of the matrix $P \cdot A$ in the same way and updates the matrix P if a lower weight codeword for some row has been found.

The second algorithm that we developed consisted in incorporating a part of the linear layer into the S-boxes and then repeat the same procedure as above. This modifies both the modeling of the linear and the S-box layer but led in many cases to much better results. As an example, this new algorithm permitted to drop down the number of initial inequalities for representing the `AES MixColumns` operation from 2176 drops to 1088.

2.4.3 Conclusion

The problem of modeling in MILP the support of a given Boolean function was recently analyzed further by Li and Sun in [LS22], where they presented a more general method for deriving linear constraints. Independently, Udovenko studied in [Udo21] how to provide the minimum possible number of inequalities for some small functions and provided lower bounds on the number of needed inequalities in some general contexts. He notably proved that the minimal number of inequalities for modeling the DDT of the `AES` lies between 2008 and 2699.

Chapter 3

New Insights into Recent Cryptanalysis Techniques

Contents

The first attacks against symmetric primitives appeared mainly in the 80s and 90s. Since then, these attacks were constantly improved and new variants were developed. From time to time new cryptanalysis techniques got introduced however this happened only scarcely. When a new technique gets introduced, it is not always clear from the seminal description what is the influence of the inner components to the success of the attack or what exact properties are exploited. It is also not always evident if the original formulation of the property or of the algorithm permitting to reveal it is the one that permits to optimally exploit the potential of the attack.

One of the research directions I explored during the last years was the understanding, reinterpretation and generalization of some newly proposed attacks or distinguishers. My work focused notably on the division property, a new powerful algorithmic method proposed by Yosuke Todo to construct integral distinguishers [Tod15b] and the multiple-of-8 property, a distinguishing property against the AES proposed by Lorenzo Grassi, Christian Rechberger and Sondre Rønjom [GRR17].

3.1 Another View of the Division Property

In 2015, Yosuke Todo introduced the *division property*, a new tool to search for integral distinguishers and attacks against symmetric primitives [Tod15b]. This method showed to be extremely powerful, as demonstrates notably the full break of the block cipher `MISTY-1` soon after the technique was introduced [Tod15a].

The division property as described by Todo [Tod15b] can be defined as follows.

Definition 1. *A multiset $X \subset \mathbb{F}_2^n$ is said to have the division property $\mathcal{D}_k^n$ for some $1 \leq k \leq n$, if the sum of all monomials over X of degree strictly less than k is equal to 0, i.e.*

$$s_u = \bigoplus_{x \in X} x^u = 0 \text{ for all } u \in \mathbb{F}_2^n \text{ such that } wt(u) < k,$$

where $wt()$ denotes the Hamming weight.

In this original formulation, known today as *conventional division property*, the division property implies that the space $\mathbb{F}_2^n$ is divided into two sets: A set $S = \{u \in \mathbb{F}_2^n$, such that $wt(u) < k\}$ for which we know that s_u vanishes and the set $\mathbb{F}_2^n \setminus S$ for which we do not know anything about s_u.

It can be easily seen that the notion of division property generalizes the notion of integrals. Indeed, $\mathcal{D}_2^n$ means that the integral of X vanishes, implying that the set X is balanced [KW02], while $\mathcal{D}_n^n$ means that X is saturated. But the novelty is that it introduces intermediate properties, $\mathcal{D}_k^n$, for $3 \leq k \leq n-1$, which do not appear in classical integral attacks. Even if these intermediate properties do not have a simple interpretation like $\mathcal{D}_2^n$ and $\mathcal{D}_n^n$, they allow to easily propagate the property through the successive rounds of a cipher by capturing some information resulting from the algebraic degree of the round function.

In 2016, in a joint article with Anne Canteaut [BC16] we introduced a novel vision of division property, by means of the so-called *parity sets*, permitting a simpler formulation and interpretation of this technique. A particular advantage of the parity sets, as notably mentioned in [Heb+20] is that it is often easier to trace the impact of a function on its parity set than on the set itself while it permits a more natural link with the algebraic normal form of the functions involved. Parity sets are considered today as a *perfect* variant of the division property as the propagation of parity sets permits to compute exactly the coefficient of some monomials in the algebraic number form of an output coordinate of

the cipher. Doing so, is however computationally hard.

3.1.1 Parity Sets

Definition 2. *Let X be a set of elements in $\mathbb{F}_2^n$. The* parity set *of X, denoted by $\mathcal{U}(X)$, is the subset of $\mathbb{F}_2^n$ defined by*

$$\mathcal{U}(X) = \{u \in \mathbb{F}_2^n : \bigoplus_{x \in X} x^u = 1\}.$$

It can be shown that there is a one-to-one correspondence between a set and its parity set. Indeed, the mapping $\mathcal{U} : X \mapsto \mathcal{U}(X)$ is involutive, i.e. $\mathcal{U}(\mathcal{U}(X)) = X$.

In the following, we write $u \preceq x$, if $u_i \leq x_i$ for all $1 \leq i \leq n$. Some useful examples are described in the next corollary.

Corollary 1. *Let X be a subset of $\mathbb{F}_2^n$. Then,*
- *$\mathcal{U}(X) = \{u \in \mathbb{F}_2^n : u \preceq x\}$ if and only if $X = \{x\}$.*
- *$\mathcal{U}(X) = \{u\}$ if and only if $X = \{x \in \mathbb{F}_2^n : x \preceq u\}$.*
- *$\mathcal{U}(X) = \{\underline{1}\}$ if and only if $X = \mathbb{F}_2^n$,*

where $\underline{1}$ denotes the all-one vector in $\mathbb{F}_2^n$.

It is now possible to reformulate the original division property of order k, $\mathcal{D}_k^n$ on a set X by a simple property of $\mathcal{U}(X)$. Indeed, $\mathcal{D}_k^n$ corresponds to a lower bound on the weights of all elements in $\mathcal{U}(X)$.

Definition 3. *A set X of elements in $\mathbb{F}_2^n$ is said to fulfill the division property of order k, $\mathcal{D}_k^n$, if all elements in $\mathcal{U}(X)$ have weight at least k, i.e.,*

$$\mathcal{U}(X) \subseteq \{u \in \mathbb{F}_2^n : wt(u) \geq k\} .$$

With the interpretation of the division property with parity sets it is easy to provide a simple characterization of the sets satisfying the division property $\mathcal{D}_k^n$ for small or large values of k.

Proposition 3.

1. *X fulfills $\mathcal{D}_1^n$ if and only if its cardinality is even.*
2. *X fulfills $\mathcal{D}_2^n$ if and only if its cardinality is even and it has the* balance property *[KW02], i.e., $\bigoplus_{x \in X} = 0$.*

3. *X fulfills the division property of order* 3, $\mathcal{D}_3^n$, *if and only if X and all the n subsets*

$$\{x \in X \text{ with } x_i = 0\},\ 1 \leq i \leq n,$$

satisfy the balance property.

4. *X fulfills $\mathcal{D}_{n-1}^n$ and not $\mathcal{D}_n^n$ if and only if X is an (affine) hyperplane of $\mathbb{F}_2^n$.*
5. *X fulfills $\mathcal{D}_n^n$, if X is either empty, or equal to the whole set $\mathbb{F}_2^n$.*

3.1.2 Todo's Distinguishers and How to Improve Them

The strategy proposed by Todo in its seminal work to build a distinguisher was to exhibit an affine subspace $a + V$ such that the corresponding output set $E_K(a + V)$ satisfies the division property of order 2, i.e., such that $E_K(a + V)$ is balanced. This property can be easily interpreted in terms of higher-order derivatives in the sense of the following definition.

Definition 4. *[Lai94] Let F be a function from $\mathbb{F}_2^n$ into $\mathbb{F}_2^m$. Let $a \in \mathbb{F}_2^n$. The derivative of F with respect to a is the function from $\mathbb{F}_2^n$ into $\mathbb{F}_2^m$ defined by*

$$D_a F(x) = F(x \oplus a) \oplus F(x)\ .$$

For any k-dimensional subspace V of $\mathbb{F}_2^n$ and for any basis of V, $\{a_1, \ldots, a_k\}$, the k-th order derivative of F with respect to V is the function defined by

$$D_V F(x) = D_{a_1} D_{a_2} \ldots D_{a_k} F(x) = \bigoplus_{v \in V} F(x + v)$$

We introduce now the following notation.

Notation 1. *Let P be a permutation of $\mathbb{F}_2^n$ and $P_1, P_2, \ldots, P_n$ be the n coordinates of P. If $x = (x_1, \ldots, x_n)$ and $u = (u_1, \ldots, u_n)$ are vectors of $\mathbb{F}_2^n$, we denote by $P^u(x)$ the coordinate product $\prod_{i=1}^n P_i(x)^{u_i}$.*

The following proposition that we demonstrated shows how the division property can be deduced by looking at the algebraic normal form (ANF) of the output function.

Proposition 4. *Let P be a permutation of $\mathbb{F}_2^n$. Let V be a linear subspace of $\mathbb{F}_2^n$ and $a \in \mathbb{F}_2^n$. Then, an element u belongs to $\mathcal{U}(P(a+V))$ if and only if the derivative of P^u with*

respect to V satisfies $D_V P^u(a) = 1$. In the particular case where $V = \{x \in \mathbb{F}_2^n : x \preceq v\}$ for some $v \in \mathbb{F}_2^n$, the following formulations are equivalent:

(i) *For all $a \in \mathbb{F}_2^n$, $u \notin \mathcal{U}(P(a+V))$*

(ii) *The ANF of the Boolean function $x \mapsto P^u(x)$ contains no monomial multiple of x^v.*

The distinguishers presented in [Tod15b] correspond to the existence of a word v such that $E_K(a + \{x \in \mathbb{F}_2^n : x \preceq v\})$ satisfies the division property of order 2 for all a, which equivalently means that the monomial x^v does not appear in any coordinate of E_K. However, it clearly appears that this type of distinguishers can be improved in the following two directions:

— it may happen that a given monomial x^u does not appear in the coordinates of E_K even if $wt(u) \leq \deg P$. This type of property, derived from the sparsity of some coordinates of the cipher, was extensively used in cube attacks, e.g. [Aum+09; DS11; Din+15]

— it may happen that a given monomial x^u appears in one coordinate of E_K but not in all functions $x \mapsto E_K^v(x)$. Then we obtain a weaker distinguisher based on the fact that a given v does not belong to the parity set of $E_K(a + \{x \in \mathbb{F}_2^n : x \preceq u\})$.

3.1.3 Exhibiting Distinguishers on SPNs by Means of Parity Sets

We then showed how to propagate some information on the parity set through the successive rounds of an SPN cipher. For this, we have to choose an affine subspace as the input set and look how the propagation evolves through the most important operations of an SPN construction, mainly key addition and S-boxes.

Propagation Through Key Addition

One of the difficulties for finding a distinguisher for a block cipher is that the distinguishing property must hold for any value of the secret key. For this reason, we need to exploit a property which can be easily propagated through the operation inserting the round key, which is usually an XOR. This is the case of differential properties, or of the algebraic degree. Next proposition shows that the parity set can also be easily propagated.

Proposition 5. *Let X be a subset of $\mathbb{F}_2^n$ with parity set $\mathcal{U}(X)$. Then, for any $k \in \mathbb{F}_2^n$, the parity set of $(k+X)$ satisfies*

$$\mathcal{U}(k+X) \subseteq \bigcup_{u \in \mathcal{U}(X)} \{x \in \mathbb{F}_2^n : x \succeq u\} .$$

It is worth noticing that there is no general improvement of the previous result which holds without any further assumption on k or on X. Thus, $\bigcup_{u \in \mathcal{U}(X)} \{x \in \mathbb{F}_2^n : x \succeq u\}$ is the smallest set which contains the parity sets of all cosets $(k+X)$ in this case.

Propagation Through an S-box

We then showed how a parity set propagates through a permutation, for instance through an S-box or through a linear permutation.

Proposition 6. *Let S be a permutation of $\mathbb{F}_2^n$. For any $v \in \mathbb{F}_2^n$, we define*

$$V_S(u) = \{v \in \mathbb{F}_2^n : S^v(x) \text{ contains } x^u\}.$$

Then, for any set X of elements of $\mathbb{F}_2^n$,

$$\mathcal{U}(S(X)) \subseteq \bigcup_{u \in \mathcal{U}(X)} V_S(u) .$$

Propagation Through One Round

We now consider an SPN where the round key is inserted by addition at the end of the round. This implies that each S-box layer comes after a round-key addition. Thus, if $\mathcal{U}(X)$ denotes the parity set of the input set X before the key addition, then the parity set after the key addition is included in a union of sets of the form $\{x \in \mathbb{F}_2^n : x \succeq u\}$, for some $u \in \mathbb{F}_2^n$. It follows that the parity set after the S-box layer satisfies

$$\mathcal{U}(S(X+k)) \subseteq \bigcup_{u \in \mathcal{U}(X)} \left(\bigcup_{v \in \{x \in \mathbb{F}_2^n : x \succeq u\}} V_S(v) \right) .$$

Therefore, propagating the information from $\mathcal{U}(X)$ to $\mathcal{U}(S(X+k))$ involves the sets

$$\mathcal{V}_S(u) = \bigcup_{v \in \{x \in \mathbb{F}_2^n : x \succeq u\}} V_S(v)$$

which depend on the S-box only. This result can then be generalized to any S-box layer, where an S-box is applied in parallel to the state.

As an application, we exploited the particular form of the sets $\mathcal{V}_S(u)$ of the S-box used inside the block cipher `PRESENT` to mount low-data distinguishers for up to 6 rounds of this cipher. This result was important at that time, as such fine-grained distinguishers became possible with the language of parity sets and could not be exhibited by using the conventional division property of the first papers of Todo [Tod15b; Tod15a].

Finally, an interesting observation we exhibited and proved is that propagations through a permutation S are closely related to propagations through the inverse of S as shows the following result.

Proposition 7. *Let S be a permutation of $\mathbb{F}_2^n$. Then, for any $u, v \in \mathbb{F}_2^n$, the ANF of $x \mapsto S(x)^v$ contains x^u if and only if the ANF of $x \mapsto \overline{S^{-1}(\overline{x})}(x)^{\overline{u}}$ contains $x^{\overline{v}}$, where $\overline{u}$ denotes the vector $u \oplus \underline{1}$.*

This result permitted us then to exhibit several criteria for an S-box in order to ensure a good resistance against the division property.

3.1.4 Conclusion

Focusing on the parity set, and not only on the minimal weight of its elements as done with the conventional division property, permitted us to capture some algebraic properties of the nonlinear functions used in the cipher, besides the algebraic degree. Soon after the publication of our paper, Todo and Morii proposed the *three-subset division property* [TM16] that permitted them in addition to capture cases of monomials for which the coefficients in the ANF of the function are known to be one. Later, Hao et al. [Hao+20; Hao+21] refined the framework further by proposing the *three-subset division property without unknown subset* that removes the set of monomials for which nothing can be said. For this they had to treat the key as a usual variable in order to be able to compute exactly the coefficients of some monomials in the ANF.

An important step for the division property was done when Xiang et al. [Xia+16] proposed the idea of division (or monomial) trails to propagate through the subsequent rounds information on monomials and furthermore suggested the idea to model the propagation with MILP. In 2020, Hu et al. and Hebborn et al. [Hao+20; Heb+20] use the notation $u \xrightarrow{F} v$ to represent the fact that the monomial x^u is present in the ANF of F^v. They also showed that the coefficient of the monomial x^u in the ANF of F^v is equal to

the number of monomial trails starting with u and ending with v modulo 2. Counting the number of trails with MILP-techniques seems to be a very prominent approach that permitted for example to find lower bounds on the algebraic degree [Heb+20] or to determine the exact degree [Hao+20] in some cases.

3.2 A general Framework for Recent AES Distinguishers

For many years, all known distinguishers of the AES in the single-key model could reach at most 4 rounds. However, since 2016, the first 5-round AES distinguishers appeared [Sun+16; GRR17; RBH17; Gra18] and this topic became again a subject of broad and current interest. The importance of these distinguishers is that they exhibit new, unexplored properties of the AES and led to improved attacks on reduced-round versions of the cipher, like the attack on 5 rounds described in [Bar+18].

The main breakthrough in these attacks is the identification by Grassi, Rechberger and Rønjom [GRR17] of the following property of the AES round function $\mathcal{R}$. There exist two well-chosen linear subspaces V and W of $\mathbb{F}_2^{128}$ satisfying the following property: for any coset of V, $(c+V)$, the number of distinct pairs of elements x, x', $x \neq x'$ in $(c+V)$ such that $\mathcal{R}(x)$ and $\mathcal{R}(x')$ belong to the same coset of W is always divisible by 8. This behaviour, known as the *multiple-of-8* property, is then combined with two 2-round deterministic subspace trails to form a 5-round distinguisher. Moreover, Grassi presented in [Gra18] new 4-round distinguishers, that exploit a property appearing in the proof of the multiple-of-8 property but that is expressed in a way that facilitates key-recovery attacks. These new distinguishers were given the name of *mixture-differential distinguishers*.

The proofs of the multiple-of-8 property and its variants given in [GRR17] were divided into many special cases that needed to be proven separately. Their main disadvantage, despite their length, was that it was not clear from them what are the characteristics and the properties of the inner components of the AES that have an influence on the multiple-of-8 behaviour. Furthermore, it was unclear, whether this kind of property was proper to the AES or whether it could be adapted to other ciphers and the original proofs did not provide hints to answer this question. The same questions could be asked for the mixture-differential distinguishers [Gra18].

In a joint work with Anne Canteaut and Daniel Coggia published at ToSC [BCC19], we provided a general formulation of the mixture-differential distinguishers and of the

multiple-of-8 property and showed that those can be applied in a systematic way to a more general class of SPN ciphers than the ones mentioned in [GRR17; Gra18]. Also, our result precisely identified the conditions to be satisfied for the property to hold and permitted to clarify which parts of the cipher have an influence on the property.

3.2.1 Distinguishers Based on Subspace Trails

Mixture-differential distinguishers, and by extension the multiple-of-8 property, are based on the notion of subspace trails [GRR16]. Let $\mathbb{K} = \mathbb{F}_{2^d}$ where d is the S-box size.

Definition 5 (Subspace trail [GRR16]). *Let $\mathcal{F} : \mathbb{K}^N \to \mathbb{K}^N$ be any map. Two linear subspaces $U, V \subseteq \mathbb{K}^N$ form a (one-round) $\mathcal{F}$-subspace trail if*

$$\forall a \in \mathbb{K}^N, \exists b \in \mathbb{K}^N : \mathcal{F}(U + a) \subseteq V + b, \tag{3.1}$$

which is denoted by $U \overset{\mathcal{F}}{\rightrightarrows} V$.

For subspace cryptanalysis, mainly exact subspace trails are of interest, i.e., trails for which equality holds in (3.1). Grassi et al. defined a number of subspaces playing a crucial role in the subspace cryptanalysis of the AES [GRR16; GRR17]. To describe these spaces, let $\mathcal{M}_4(\mathbb{K})$ denote the set of all 4×4-matrices over $\mathbb{K}$ and let $(e_{i,j})_{i,j \in \{0,\dots,3\}}$ be the canonical basis of $\mathcal{M}_4(\mathbb{K})$. Let $\mathrm{vect}_{\mathbb{K}}(v_0, \dots, v_{k-1})$ denote the linear space formed by all linear combinations with coefficients in $\mathbb{K}$ of the vectors $v_0, \dots, v_{k-1} \in \mathcal{M}_4(\mathbb{K})$. The following subspaces of $\mathcal{M}_4(\mathbb{K})$ for $i \in \{0, \dots, 3\}$ have indices computed modulo 4:

$$\begin{array}{lcrcl}
\text{The column spaces} & : & \mathcal{C}_i & = & \mathrm{vect}_{\mathbb{K}}(e_{0,i}, e_{1,i}, e_{2,i}, e_{3,i}), \\
\text{The diagonal spaces} & : & \mathcal{D}_i & = & \mathrm{vect}_{\mathbb{K}}(e_{0,i}, e_{1,i+1}, e_{2,i+2}, e_{3,i+3}), \\
\text{The anti-diagonal spaces} & : & \mathcal{ID}_i & = & \mathrm{vect}_{\mathbb{K}}(e_{0,i}, e_{1,i-1}, e_{2,i-2}, e_{3,i-3}), \\
\text{The mixed spaces} & : & \mathcal{M}_i & = & \mathsf{MC}(\mathcal{ID}_i)
\end{array}$$

where MC is the MixColumns operation of the AES.

If $I \subseteq \{0, 1, 2, 3\}$, ne can define :

$$\mathcal{C}_I = \bigoplus_{i \in I} \mathcal{C}_i, \quad \mathcal{D}_I = \bigoplus_{i \in I} \mathcal{D}_i, \quad \mathcal{ID}_I = \bigoplus_{i \in I} \mathcal{ID}_i, \quad \mathcal{M}_I = \bigoplus_{i \in I} \mathcal{M}_i.$$

The 5-round distinguisher on the AES presented in [GRR17] is based on the following keystone lemma. Note that we always consider *unordered pairs* of elements and denote

them as pair sets, i.e. $\{a, b\}$.

Lemma 1 ([GRR17]). *Let* $a \in \mathcal{M}_4$, $i \in \{0,1,2,3\}$, $J \subseteq \{0,1,2,3\}$. *We define*

$$n = \#\{\ \{p^0, p^1\}\ \textit{with}\ p^0, p^1 \in \mathcal{M}_i + a \mid \mathcal{R}(p^0) + \mathcal{R}(p^1) \in \mathcal{D}_J\}.$$

Then $n \equiv 0 \mod 8$.

This lemma can then be composed before and after with 2-round exact subspace trails :

$$\mathcal{D}_i \overset{\mathcal{R}}{\rightrightarrows} \mathcal{C}_i \overset{\mathcal{R}}{\rightrightarrows} \mathcal{M}_i \quad \xrightarrow{\mathcal{R}} \quad \mathcal{D}_J \overset{\mathcal{R}}{\rightrightarrows} \mathcal{C}_J \overset{\mathcal{R}}{\rightrightarrows} \mathcal{M}_J,$$

to form a 5-round distinguisher. Lemma 1 has a five-page-long proof in [GRR17] and a central result of our work was that we managed to provide a much more compact and informative proof that we could then easily generalize to other SPN designs.

3.2.2 Our Proof of Lemma 1

Instead of proving Lemma 1, we proved directly a more general variant, which considers $\mathcal{M}_I$ with any subspace $I \subseteq \{0,1,2,3\}$, instead of $|I| = 1$. This generalization was present in the original paper but its proof was only sketched as the original proof framework of [GRR17] did not allow a compact proof of this generalization.

A central notion for our proof is the notion of *information set*, defined for a pair of elements in a coset of $\mathcal{M}_I$ as follows.

Definition 6 (Information set). *Let* $\{p^0, p^1\}$ *be a set of elements in* $\mathcal{M}_I + a$, *written as*

$$p^0 = \sum_{k=0}^{3} \sum_{i \in I} p^0_{i,k} \mathsf{MC}(e_{i-k,k}) + a \quad \textit{and} \quad p^1 = \sum_{k=0}^{3} \sum_{i \in I} p^1_{i,k} \mathsf{MC}(e_{i-k,k}) + a$$

for some (uniquely defined) $p^0_{i,k}, p^1_{i,k} \in \mathbb{K}$, $i \in I, 0 \leq k \leq 3$. *The* information set K *of* $\{p^0, p^1\}$ *is defined as*

$$K = \{k \in \{0,1,2,3\} \mid \exists i \in I : p^0_{i,k} \neq p^1_{i,k}\}\ .$$

Our approach for proving Lemma 1 can be divided into three steps. The first step is to define the following equivalence relation between pairs of elements in $\mathcal{M}_I + a$:

Definition 7 (Equivalence relation). *Let* $P = \{p^0, p^1\}$ *and* $Q = \{q^0, q^1\}$ *with* $p^0, p^1, q^0, q^1 \in \mathcal{M}_I + a$. *We say that* $P \sim Q$ *if:*

— $\{p^0, p^1\}$ *and* $\{q^0, q^1\}$ *have the same information set* K.
— $\forall k \in K, \exists b \in \{0,1\} : \forall i \in I, q^0_{i,k} = p^b_{i,k}$ *and* $q^1_{i,k} = p^{1-b}_{i,k}$.

Clearly, $\sim$ *is an equivalence relation on unordered pairs of* $\mathcal{M}_I + a$.

Example 1. *The following two sets* $\{p_0, p_1\}$ *and* $\{q_0, q_1\}$, *with* $p_0, p_1, q_0, q_1 \in \mathcal{M}_0$ *are equivalent.*

$$\{p_0, p_1\} = \left\{ \begin{pmatrix} 2\cdot x_0 & x_1 & z_2 & 3\cdot z_3 \\ x_0 & x_1 & 3\cdot z_2 & 2\cdot z_3 \\ x_0 & 3\cdot x_1 & 2\cdot z_2 & z_3 \\ 3\cdot x_0 & 2\cdot x_1 & z_2 & z_3 \end{pmatrix}, \begin{pmatrix} 2\cdot y_0 & y_1 & z_2 & 3\cdot z_3 \\ y_0 & y_1 & 3\cdot z_2 & 2\cdot z_3 \\ y_0 & 3\cdot y_1 & 2\cdot z_2 & z_3 \\ 3\cdot y_0 & 2\cdot y_1 & z_2 & z_3 \end{pmatrix} \right\}$$

$$\sim$$

$$\{q_0, q_1\} = \left\{ \begin{pmatrix} 2\cdot x_0 & y_1 & w_2 & 3\cdot w_3 \\ x_0 & y_1 & 3\cdot w_2 & 2\cdot w_3 \\ x_0 & 3\cdot y_1 & 2\cdot w_2 & w_3 \\ 3\cdot x_0 & 2\cdot y_1 & w_2 & w_3 \end{pmatrix}, \begin{pmatrix} 2\cdot y_0 & x_1 & w_2 & 3\cdot w_3 \\ y_0 & x_1 & 3\cdot w_2 & 2\cdot w_3 \\ y_0 & 3\cdot x_1 & 2\cdot w_2 & w_3 \\ 3\cdot y_0 & 2\cdot x_1 & w_2 & w_3 \end{pmatrix} \right\}$$

The information set of these pairs has cardinality $|K| = 2$.

Then, we proved that for any $a \in \mathcal{M}_4(\mathbb{K})$, the function Δ operating on unordered pairs of elements in $\mathcal{M}_I + a$ and defined by

$$\Delta : \quad \{p^0, p^1\} \quad \longmapsto \quad \mathcal{R}(p^0) + \mathcal{R}(p^1)$$

is constant over the equivalence classes for $\sim$. Finally, we showed that the cardinality of any equivalence class $\mathfrak{C}$ with information set K is

$$|\mathfrak{C}| = 2^{|K|-1+d|I|(4-|K|)},$$

This number is then always a multiple of 8.

The generalization of Lemma 1 where $\mathcal{M}_i$ is replaced with $\mathcal{M}_I$ can then be easily proved. Indeed, if $\mathcal{P}^2(\mathcal{M}_I + a)$ denotes the set of all unordered pairs of elements in $\mathcal{M}_I + a$, and $\mathcal{P}^2(\mathcal{M}_I + a)\big/\sim$ denotes the set of all equivalence classes for $\sim$, since the equivalence

classes form a partition of $\mathcal{P}^2(\mathcal{M}_I + a)$, we have that

$$n = \left|\Delta^{-1}(\mathcal{D}_J)\right| = \sum_{\mathfrak{C} \in \mathcal{P}^2(\mathcal{M}_I+a)\Big/\sim} \left|\Delta^{-1}(\mathcal{D}_J) \cap \mathfrak{C}\right| .$$

We know now that Δ is constant on the equivalence classes, implying that $(\Delta^{-1}(\mathcal{D}_J) \cap \mathfrak{C})$ equals either 0 or $|\mathfrak{C}|$. In other words, there exists a function δ from $\mathcal{P}^2(\mathcal{M}_I + a)\Big/\sim$ into $\{0, 1\}$ such that

$$\left|\Delta^{-1}(\mathcal{D}_J) \cap \mathfrak{C}\right| = \delta(\mathfrak{C}) \times |\mathfrak{C}| .$$

It follows that

$$n = \sum_{\mathfrak{C} \in \mathcal{P}^2(\mathcal{M}_I+a)\Big/\sim} \delta(\mathfrak{C}) \times |\mathfrak{C}| \equiv 0 \mod 8,$$

since all equivalence classes have a cardinality divisible by 8.

Influence of the Branch Number The original proof of the multiple-of-8 property given in [GRR17], used the fact that the differential branch number b of MC is maximal. Our proof shows that the branch number b does have an influence on the exact number of n but not the fact that it is a multiple of 8.

Indeed, the influence of the branch number on n can be expressed with the formula

$$n = \sum_{h=b-|J|}^{4} \sum_{\mathfrak{C}:|K(\mathfrak{C})|=h} \Delta(\mathfrak{C}) \times |\mathfrak{C}| ,$$

which does not affect the multiple-of-8 property.

Grassi's distinguishers Our methodology permitted us then to prove very easily all of the mixture-differential distinguishers presented in [Gra18]. Indeed, the multiple-of-8 property is a consequence of the fact that the function Δ defined above is constant over each equivalence class. However, this invariance can be directly used as a distinguishing property. This is actually what is done in [Gra18].

3.2.3 Generalization and Applications

An objective of our work was to find out what is the particular property of the subspaces $\mathcal{M}_I$ for the generalization of Lemma 1 to work and whether these spaces could be

replaced by some other subspace V of $\mathbb{K}^N$ without altering the result. With this goal in mind, we analyzed the form of such subspaces with respect to the non-linear layer, for a general SPN cipher and provided necessary conditions for their successful combination.

The proof of the fact that the function Δ is constant on the equivalence classes relied on the fact that the coordinates of the elements in the input subspace V can be decomposed into several subsets in such a way that each S-box involves only one coordinate subset. This property is captured by the following definition: it requires the existence of a basis of V which can be decomposed over the original basis $\{f_0, \ldots, f_{N-1}\}$ by a block-diagonal matrix.

For our generalization we suppose that the substitution layer $\mathcal{S}$ applies an invertible S-box : $\mathbb{K} \to \mathbb{K}$ to each word of the internal state in a certain basis $\{f_0, \ldots, f_{N-1}\}$ of $\mathbb{K}^N$.

Definition 8. *Let V be a subspace of $\mathbb{F}^N$. We say that V is* compatible with $\mathcal{S}$ *if there exists a basis of V whose elements written in the basis $\{f_0, \ldots, f_{N-1}\}$ form a block-diagonal matrix (with blocks having potentially different dimensions) for a certain order of the elements in both bases. We call such a basis of V a* compatibility basis.

Given an arbitrary basis of a subspace V, it is quite easy to check whether V is compatible with $\mathcal{S}$ by computing the unique reduced echelon form of the corresponding matrix. If, for a given ordering of the rows, this matrix has a reduced echelon form which is block-diagonal, then V is compatible with $\mathcal{S}$ and the reduced echelon form provides a compatibility basis. Otherwise, V is not compatible with $\mathcal{S}$. The number of blocks h is an essential parameter as it occurs in the following generalization of of the multiple-of-8 property.

Corollary 2. *Let $\mathcal{E}$ be any subset of $\mathbb{F}^N$ and*

$$n = \#\{\ \{p^0, p^1\} \text{ with } p^0, p^1 \in (V + a) \mid \mathcal{R}(p^0) + \mathcal{R}(p^1) \in \mathcal{E}\}.$$

Then $n \equiv 0 \mod 2^{h-1}$.

Applications After having identified the conditions for the multiple-of property and the mixture-differential distinguishers to hold for an SPN cipher, we provided applications of our generalization to ciphers other than the AES, demonstrating that these properties are not proper to the AES but that they hold for many other SPN constructions. With the generalization of the previous section at hand, the application to other ciphers was almost

straightforward. We showed indeed that the ciphers `Midori` [Ban+15a], `LED` [Guo+11], `KLEIN` [GNL12] and `SKINNY` [Bei+16] had the multiple-of-h property for some integer value of h and that for the ciphers `CRYPTON` [Lim98] and `PRINCE` [Bor+12] this property does not apply as well as expected because of some compatibility issues due to the form of the trails.

Chapter 4

Differential properties of cryptographic S-boxes

Contents

Non-linearity is essential for achieving security and the design of a non-linear function having good cryptographic properties is a crucial problem. However, applying a single large non-linear function to the whole state is very expensive both in terms of time and memory. For this reason, it is common to divide the internal state into small words (typically 3-8 bits) and apply a small non-linear function, called Substitution box, or simply *S-box* to each word independently.

As a cipher should be designed to resist basic state of the art attacks (e.g. differential, linear, algebraic, boomerang, ...) and as the S-box is typically the only non-linear component of the cipher, the resistance of the entire construction heavily depends on the properties of the underlying S-box.

This chapter is dedicated to the analysis of the S-box properties related to the resistance of the cipher against differential and boomerang attacks. It presents the research results I obtained together with my co-authors in [BC18], [Bou+19] and [TBP20].

4.1 Two Notions of Differential Equivalence on S-boxes

Differential cryptanalysis [BS90] exploits differentials, i.e. pairs of input-output differences (a, b) that appear with high probability. For evaluating the resistance of a cipher against this type of attacks, it is important to evaluate the maximum probability that a given input difference yields a given output difference through the S-box. This property can be summarized in the so-called *Difference Distribution Table (DDT)*.

Definition 9 (DDT and differential uniformity)**.** *Let F be an S-box from $\mathbb{F}_2^n$ into $\mathbb{F}_2^n$. The difference distribution table (DDT) of F is the two-dimensional table defined by*

$$\delta_F(a, b) = \#\{x \in \mathbb{F}_2^n : F(x) \oplus F(x \oplus a) = b\},$$

for all $a, b \in \mathbb{F}_2^n$. An important characteristic of the DDT introduced in [NK92; Nyb93] is the differential uniformity of F denoted by δ_F and defined as the highest non-trivial value in the DDT, i.e.

$$\delta_F = \max_{a,b \in \mathbb{F}_2^n, a \neq 0} \delta_F(a, b).$$

The differential uniformity should be as small as possible in order to maximize the complexity of differential attacks and has become an important design criterion for S-boxed-based constructions. It it easy to see that the lowest possible value for the differential uniformity of a function from $\mathbb{F}_2^n$ into itself is 2 and the functions with differential uniformity 2 are called *almost perfect nonlinear (APN)*. However, besides the differential uniformity of the S-box, the whole differential spectrum and even the form of the difference distribution table (DDT) are important when the resistance against several variants of differential cryptanalysis is quantified [Knu94; Par+03; BG11; CR15].

When designing a block cipher, it would then be of major interest to be able to start from a desired DDT which guarantees a high resistance against all variants of differential cryptanalysis, and to construct S-boxes having this specific DDT. Instead, the main technique currently available to the designers consists in randomly generating S-boxes until one with a suitable DDT is found. However, constructing S-boxes from a prescribed DDT is a difficult problem, related to many open issues in the area. In a joint work with Anne Canteaut, Jérémy Jean and Valentin Suder that was published in 2019 at DCC [Bou+19], we aimed to advance general knowledge about DDTs and tried to provide an answer to the following two questions:

1. What can we say about S-boxes that share the same DDT?
2. How can we compute all S-boxes that share the same DDT?

To study the first problem we introduced two different equivalence notions for vectorial Boolean functions, that we called *DDT and γ-equivalence*.

4.1.1 DDT-Equivalence and γ-Equivalence

Besides differential uniformity, another important characteristic of the DDT of an S-box $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$, introduced in [CCZ98a] is the so-called *indicator of the DDT*. This corresponds to the Boolean function γ_F of $2n$ variables defined by

$$\gamma_F(a, b) = 0 \text{ if and only if } F(x \oplus a) \oplus F(x) = 0 \text{ or } a = 0.$$

Definition 10. *We say that F and G are* DDT-equivalent *if they have the same DDT, and that they are* γ-equivalent *if their DDTs have the same support, or equivalently if $\gamma_F = \gamma_G$.*

Note that the notion of γ-equivalence has been investigated under the name *differential equivalence* by Gorodilova [Gor16; Gor19].

Obviously, DDT-equivalence implies γ-equivalence. However, the converse also holds in some particular cases.

Proposition 8. *Let F and G be two functions from $\mathbb{F}_2^n$ into itself which are γ-equivalent. Assume that, for each derivative of F and G, there exists some integer λ such that the derivative is a λ-to-1 function. Then, F and G are DDT-equivalent. Most notably, this situation holds when both F and G are quadratic functions. It also implies that any function which is γ-equivalent to an APN function F is also DDT-equivalent to F.*

It is worth noticing that the previous proposition does not mean that the γ-equivalence class $\mathcal{C}_\gamma(F)$ of a quadratic function F equals its DDT-equivalence class $\mathcal{C}_{\mathsf{DDT}}(F)$. A quadratic function F such that $\mathcal{C}_\gamma(F) \neq \mathcal{C}_{\mathsf{DDT}}(F)$ is given in the following example.

Example 2. *Let* $F : \mathbb{F}_2^4 \to \mathbb{F}_2^4$ *with*

$$F = [1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1].$$

F is quadratic and has differential uniformity 8*. Its DDT-equivalence class* $\mathcal{C}_{\mathsf{DDT}}(F)$ *contains* 14336 *functions, while its γ-equivalence class* $\mathcal{C}_\gamma(F)$ *has* 484352 *functions. The γ-equivalence class of F contains notably functions of different algebraic degrees.*

However, the two notions coincide for APN functions, implying that the γ-equivalent APN functions exhibited in [Gor16] are also DDT-equivalent. In general however, the two notions of differential equivalence do not coincide and it is not difficult to exhibit two γ-equivalent functions with different DDTs.

Properties of Differential-Equivalence Classes

When trying to tackle our first general problem *"What can we say about S-boxes that share the same DDT?"*, a natural question is: *"How many functions share the same DDT?"* which corresponds to the problem of determining the size of a given DDT-equivalence class.

A lower bound on these sizes is given in the following proposition.

Proposition 9. *Let F be a function from $\mathbb{F}_2^n$ into itself and let ℓ denote the dimension of its linear space, i.e., of the space formed by all linear structures of F. Then, the DDT-equivalence class of F contains the $2^{2n-\ell}$ distinct functions of the form*

$$x \mapsto F(x \oplus c) \oplus d, \quad c, d \in \mathbb{F}_2^n. \tag{4.1}$$

We will say that two functions are trivially DDT-equivalent if they satisfy Eq. (4.1) from Proposition 9. Moreover, we say that a DDT-equivalent class is *trivial* if its size matches the lower-bound given in Proposition 9.

An interesting question is what cryptographic properties are preserved under DDT-equivalence and γ-equivalence.

Algebraic degree The algebraic degree is not preserved under DDT-equivalence. The following proposition gives a nice illustration of this fact.

Proposition 10. *For any even $n \geq 4$, there exists a quadratic function Q from $\mathbb{F}_2^n$ into $\mathbb{F}_2^n$ whose DDT-equivalence class contains a function of any degree d, $2 \leq d \leq n/2$.*

Differential uniformity Obviously, the DDT-equivalence preserves the differential uniformity. However, this is not the case for γ-equivalence. For instance, the function

$$F' = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1]$$

is γ-equivalent to the function F of Example 2. However, the differential uniformity of F' is 10, while F has differential uniformity 8.

Invariance of the Sizes of Differential-Equivalence Classes Under CCZ-Equivalence

Another important property of the size of these equivalence classes is the following result proved in [Gor16] for γ-equivalence, which can easily be generalized to DDT-equivalence.

Proposition 11 (adapted from [Gor16])**.** *Let F and G be two functions which are EA-equivalent, i.e., there exist three affine functions A_0, A_1, A_2 where A_1 and A_2 are bijective such that $G = A_2 \circ F \circ A_1 \oplus A_0$. Then, the DDT-equivalence classes (resp. γ-equivalence classes) of F and of G have the same size. Moreover, the class of G is composed of all $A_2 \circ F' \circ A_1 \oplus A_0$ where F' varies in the class of F.*

Whether an analogue of the previous result holds for CCZ-equivalence is a natural question raised by Gorodilova [Gor16; Gor19]. An obvious case for which it can be proved that two CCZ-equivalent functions F and G have DDT-classes of the same size is when F is a permutation and $G = F^{-1}$. Indeed, if F is a permutation, any function Φ in $\mathcal{C}_{\mathsf{DDT}}(F)$ is a permutation too and the DDT of Φ^{-1} satisfies

$$\delta_{\Phi^{-1}}(a, b) = \delta_{\Phi}(b, a) = \delta_F(b, a) = \delta_{F^{-1}}(a, b).$$

Thus, $\mathcal{C}_{\mathsf{DDT}}(F^{-1}) = \{\Phi^{-1}, \Phi \in \mathcal{C}_{\mathsf{DDT}}(F)\}$. But the general case of two CCZ-equivalent functions F and G seems more difficult. Indeed, CCZ-equivalence means that $\{(x, G(x)), x \in \mathbb{F}_2^n\}$ is the image of $\{(x, F(x)), x \in \mathbb{F}_2^n\}$ by a linear permutation $\mathcal{L}$ of $\mathbb{F}_2^n \times \mathbb{F}_2^n$. However,

this implies that, if $\mathcal{L}$ is seen as a pair of functions, $\mathcal{L} : (x, y) \mapsto (L_1(x, y), L_2(x, y))$, then $x \mapsto L_1(x, F(x))$ is a permutation. This last condition is then required for transforming a given function F into a CCZ-equivalent function. Thus, if we want to prove, as it holds when F and G are EA-equivalent, that the DDT-equivalence class of F can be transformed into the DDT-equivalence of G by applying the same $\mathcal{L}$, we need to prove that $x \mapsto L_1(x, \Phi(x))$ is a permutation for all $\Phi \in \mathcal{C}_{\mathsf{DDT}}(F)$. This is the keypoint in the following theorem.

Theorem 1. *Let F and G be two CCZ-equivalent functions from $\mathbb{F}_2^n$ into itself and let $\mathcal{L}$ be a linear permutation of $\mathbb{F}_2^n \times \mathbb{F}_2^n$ seen as a pair of two linear functions: $\mathcal{L}(x, y) = (L_1(x, y), L_2(x, y))$ such that $F_1 : x \mapsto L_1(x, F(x))$ is a permutation and*

$$\mathcal{L}(x, F(x)) = (F_1(x), G \circ F_1(x)) \text{ for all } x \in \mathbb{F}_2^n.$$

Then, the DDT-equivalence classes (resp. γ-equivalence classes) of F and of G have the same size. More precisely,

$$\mathcal{C}_{\mathsf{DDT}}(G) = \left\{ L_2(\Phi_1^{-1}(x), \Phi \circ \Phi_1^{-1}(x)),\ \Phi \in \mathcal{C}_{\mathsf{DDT}}(F) \right\},$$
$$\mathcal{C}_{\gamma}(G) = \left\{ L_2(\Phi_1^{-1}(x), \Phi \circ \Phi_1^{-1}(x)),\ \Phi \in \mathcal{C}_{\gamma}(F) \right\},$$

where $\Phi_1 : x \mapsto L_1(x, \Phi(x))$.

It follows that the sizes of these differential-equivalence classes can be computed for one representative in each CCZ-equivalence class only.

It is also possible to provide other characterizations of the DDT-equivalence in terms of the Walsh transform or by the Hamming weights of the derivatives of the components of the S-box (see [Bou+19][Prop. 7, 8]).

4.1.2 An Algorithm to Compute the Differential-Equivalence Classes

To answer our second question *"How can we compute all S-boxes that share the same DDT?"* we provided a guess-and-determine algorithm capable to compute the DDT or γ-equivalence class for a given function.

Our algorithm takes as input a $2^n \times 2^n$ table D filled with non-negative integers and returns all functions F from $\mathbb{F}_2^n$ into itself, if any, whose difference distribution table has the

same indicator as the one of D, which we denote γ_D. In other words, our algorithm retrieves the γ-equivalence class of functions of a given table D. Note that one can also derive the DDT-equivalent functions from this class, by post-filtering the functions returned by the algorithm against a desired DDT.

The algorithm determines all possible values for $F(i)$, $i = 0, \ldots, 2^n - 1$, by taking into account the constraints imposed by the table D and the values $F(j)$, $j < i$, that have already been computed. It essentially implements a tree-traversal algorithm, where each Level i contains the nodes corresponding to the possible values that $F(i)$ can take. The tree therefore has depth 2^n and its details can be found in [Bou+19].

4.1.3 Experimental Results

Using our algorithm, we have been able to compute the DDT and γ-equivalence classes of some cryptographically relevant functions, by running the algorithm for a single representative in each CCZ-equivalence class (see Theorem 1). What we were principally trying to investigate during our experiments was whether there exist two cryptographically interesting DDT-equivalent permutations F and G, which are not related by $G(x) = F(x \oplus c) \oplus d$ for some c, d.

For this, we analyzed all known APN permutations over $\mathbb{F}_2^n$, for $n \leq 9$ and could affirm that all of them had trivial DDT-equivalence classes.

We also examined all permutations of dimension $n = 4$ with optimal differential uniformity (equal to 4) and optimal nonlinearity from the 16 different affine-equivalence classes given in [LP07] and proved that both their DDT and γ-equivalence classes are trivial.

To conclude, none of the permutations with the lowest possible differential uniformity in dimension $n < 6$ has a DDT-equivalence class with size bigger than 2^{2n}. However, it is possible to construct such permutations when we increase the differential uniformity. For example, the two permutations over $\mathbb{F}_2^5$ given in Table 4.5, are differentially 32-uniform and can be shown to be non-trivially DDT-equivalent.

Finally our algorithm permitted us to conduct many experiments with non-bijective APN functions and the results can be found in [Bou+19].

4.1.4 A Conjecture, a Related Work and Open Problems

Our work permitted, among others, to exhibit new examples of functions having a non-trivial DDT-equivalence class. An interesting question is whether it is possible to identify

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$F(x)$	0	1	2	3	4	5	6	7	8	9	10	11	13	12	15	14
$F'(x)$	0	1	2	3	4	5	6	7	8	9	10	11	13	12	15	14
x	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
$F(x)$	16	17	19	18	20	21	23	22	25	24	26	27	28	29	31	30
$F'(x)$	16	17	19	18	21	20	22	23	24	25	27	26	28	29	31	30

Table 4.1 – Two non-trivially DDT-equivalent permutations.

some particular property or characteristic of functions having non-trivial DDT-equivalence classes. By observing the few examples of known functions having this property, e.g. the Gold functions for $n \equiv 0 \bmod 4$ [Gor16], a quadratic APN function in six variables (Number 12 in Table 5 in [BL08]), and the examples exhibited in this paper, we remarked that a common point between these functions is that they have non-distinct rows in their DDTs. Based on the computations we performed and this fact, we stated the following conjecture.

Conjecture 1. *The DDT-equivalence class of a permutation F, such that the rows in its DDT are pairwise distinct, only contains permutations of the form $F(x \oplus c) \oplus d$, with $c, d \in \mathbb{F}_2^n$ (i.e. is trivial).*

Note that this conjecture does not hold for γ-equivalence.

Some other questions about the size of the DDT-equivalence and the γ-equivalence classes naturally arose during this work. For example, we were not able to find an example of an APN function in an odd number of variables having a non-trivial γ-equivalence class. The question is then if such APN functions exist. Another question is whether there exist differentially 4-uniform permutations whose DDT-equivalence class is different from its γ-equivalence class. We have found examples of differentially 4-uniform non-bijective functions for which the two associated classes are different, but in the bijective case, for all the tested functions, the two notions coincide. Finally, an interesting future direction would be to study the differential equivalence classes, and in particular the sizes, of functions either in higher dimensions and/or without any particular structure.

Some time after our paper was published, Dunkelman and Huang provided in [DH19] a different algorithm for computing the DDT-equivalence class of a given function. Their algorithm is different in nature and exploits the link between the DDT and the linear approximation table (LAT) of a given function. It has the particularity to work well in

some particular cases while for many others, notably for APN functions, it is worse than the algorithm we provided in our work.

4.2 Boomerang Uniformity of Cryptographic S-boxes

The boomerang attack introduced by Wagner in 1999 [Wag99], is an important cryptanalysis technique against block ciphers that can be seen as an extension of classical differential attacks [BS90]. The main idea is to join in a clever way two (short) high-probability differentials to obtain a high-probability distinguisher. In the basic version of this attack, a cipher E is considered as the composition of two sub-ciphers E_0 and E_1, i.e. $E = E_1 \circ E_0$. Boomerang attacks work by forming a quartet structure based on a differential $a \to d$ for E_0 of probability p and a differential $c \to b$ for E_1 of probability q. By supposing the two differentials to be independent, it is possible to obtain a boomerang distinguisher with probability

$$\Pr[E^{-1}(E(x) \oplus b) \oplus E^{-1}(E(x \oplus a) \oplus b) = a] = p^2 q^2. \tag{4.2}$$

Since Wagner's seminal paper, many improvements and variants of boomerang attacks have been proposed [KKS00; BDK01; BDK02; BDD03; BK09; DKS10; Kim+12]. However, Murphy [Mur11] pointed out that the independence assumption used for establishing Eq. (4.2) may fail and the examples he provided showed that a rigorous analysis of the dependence between the two involved differentials (especially at their junction) is necessary, both in the classical attack as also in all its generalizations.

An important breakthrough in boomerang cryptanalysis was the introduction in 2018 by Cid et al. of a new method for treating the dependency issues in a more systematic way [Cid+18]. The introduced approach consisted in precomputing all boomerangs for a single S-box by means of a table, similarly with what is done in differential cryptanalysis and the difference distribution table (DDT). The table introduced in [Cid+18] was named the Boomerang Connectivity Table (BCT) and its role was to keep for each a and b the number of solutions of the equation

$$S^{-1}(S(x) \oplus b) \oplus S^{-1}(S(x \oplus a) \oplus b) = a.$$

In this way, the values of the BCT record the probability of generating boomerang quartets at the S-box level (up to a factor 2^{-n}) and permit to visualize known *switches* or to

reveal better ones. Understanding the properties of such tables allows then designers and cryptanalysts to better evaluate the applicability of boomerang attacks and provides new criteria for designing S-boxes.

The introduction of the BCTs in [Cid+18] was accompanied by a preliminary analysis of their properties and especially of their link with the corresponding DDTs. The authors showed notably that the maximum in the BCT, β, is at least equal to the differential uniformity of the S-box. Moreover, for APN permutations, the BCT and DDT tables coincide. While this means that the APN case is entirely settled, this result is mainly of theoretical interest as such permutations are only known to exist for odd dimensions, with the only exception being Dillon et al.'s permutation in dimension 6 [Bro+10]. Therefore, in even dimensions, notably for $n = 4$ or $n = 8$, S-boxes with differential uniformity 4 are usually preferred. Studying therefore the resistance of such S-boxes against boomerang attacks is an important task. Notably, an open question raised in [Cid+18] was whether optimal differentially 4-uniform S-boxes exist against boomerang attacks, where optimal means that the maximal value in the BCT is 4.

In a joint paper with Anne Canteaut published at ToSC [BC18] we studied the properties of the newly introduced BCT, solved some of the problems raised in [Cid+18] and provided results for the BCTs of some important cryptographic families of S-boxes.

4.2.1 Definition and Basic Properties

We provide here the definition of the Boomerang Connectivity Table (BCT) for a permutation of $\mathbb{F}_2^n$, as introduced in [Cid+18] and the notion of *boomerang uniformity* that we introduced in [BC18].

Definition 11. *Let F be a permutation of $\mathbb{F}_2^n$. The* Boomerang Connectivity Table *(BCT) of S is the two-dimensional table defined by*

$$\beta_S(a,b) = \#\{x \in \mathbb{F}_2^n : S^{-1}(S(x) \oplus b) \oplus S^{-1}(S(x \oplus a) \oplus b) = a\}, \text{ with } a, b \in \mathbb{F}_2^n.$$

The boomerang uniformity*, denoted by β_S, is the highest value in the BCT without considering the row and the column of index* 0*:*

$$\beta_S = \max_{a,b \in \mathbb{F}_2^n \setminus \{0\}} \beta_S(a,b).$$

We first showed that the multi-set composed of all values in the BCT is preserved

under affine equivalence and inversion. This very simple result is useful as it restrains the study of BCTs in a given dimension to the study of the properties of a single representative of the affine equivalence class.

Proposition 12. *Let F and G be two permutations of $\mathbb{F}_2^n$ which are affine-equivalent, i.e., there exist two affine permutations A_1 and A_2 such that $G = A_2 \circ F \circ A_1$. Then, the BCT of F and G are related by*

$$\beta_G(a, b) = \beta_F(L_1(a), L_2^{-1}(b)), \quad \text{for all } a, b \in \mathbb{F}_2^n ,$$

where L_1 and L_2 denote the linear parts of A_1 and A_2 respectively.

A similar relation can be exhibited between the BCT of a permutation and the BCT of its inverse.

Proposition 13. *Let S be a permutation of $\mathbb{F}_2^n$. Then, the BCT of S and of S^{-1} are related by*

$$\beta_{S^{-1}}(a, b) = \beta_S(b, a), \quad \text{for all } a, b \in \mathbb{F}_2^n .$$

Note that the behaviour of the BCT with respect to affine equivalence and inversion is exactly the same as the behaviour of the DDT. However, while the differential spectrum of a function is also preserved by the extended-affine (EA) equivalence, this is not the case for the BCT. For instance, the two permutations G_4 and G_6 from [LP07] are EA-equivalent, but their boomerang uniformities differ: $\beta_{G_4} = 10$ and $\beta_{G_6} = 8$. As a consequence, as EA-equivalence is a special case of CCZ-equivalence, we deduced that the boomerang uniformity is also not always preserved under CCZ-equivalence [CCZ98b].

4.2.2 BCT for 4-bit Permutations

We showed that the maximum value in the BCT is preserved under affine equivalence. It is then sufficient to study the BCT for one representative of the affine equivalence class. For $n = 4$ full classifications exist, see for example [De 07] or [LP07]. Following the classification by De Cannière, we show in Table 4.2 the spectrum of the BCT for all classes of 4-bit permutations with $\delta_S = 4$, i.e., the values n_i corresponding to the number of times the value i appears in the BCT. This classification includes all optimal permutations with $\delta_S = 4$ and optimal linearity $\mathcal{L}(S) = 8$ listed in [LP07], and also permutations with $\delta_S = 4$ and a higher linearity.

	Representative	$\mathcal{L}(S)$	[De 07]	[LP07]	n_2	n_4	n_6	n_8	n_{10}	n_{16}	β_S
1	[8, 0, 1, 12, 15, 5, 6, 7, 4, 3, 10, 11, 9, 13, 14, 2]	8	3	G_3	60	15	30	0	0	0	**6**
2	[2, 0, 1, 8, 3, 11, 6, 7, 4, 9, 10, 15, 12, 13, 14, 5]	8	6	G_5	72	27	18	0	0	0	**6**
3	[8, 0, 1, 12, 2, 5, 6, 9, 4, 3, 10, 11, 7, 13, 14, 15]	8	2	G_6	80	27	10	4	0	0	**8**
4	[8, 0, 1, 9, 2, 5, 13, 7, 4, 6, 10, 11, 12, 3, 14, 15]	8	8	G_{11}	85	30	5	5	0	0	**8**
5	[4, 0, 1, 15, 2, 11, 6, 7, 3, 9, 10, 5, 12, 13, 14, 8]	8	1	G_{13}	78	28	11	2	1	0	**10**
6	[2, 0, 1, 8, 3, 13, 6, 7, 4, 9, 10, 5, 12, 11, 14, 15]	8	4	G_4	72	23	14	0	4	0	**10**
7	[2, 0, 1, 8, 3, 15, 6, 7, 4, 9, 5, 11, 12, 13, 14, 10]	8	5	G_7	80	30	5	0	5	0	**10**
8	[4, 8, 1, 2, 3, 11, 6, 7, 0, 9, 10, 14, 12, 13, 5, 15]	8	7	G_{12}	75	25	10	0	5	0	**10**
9	[8, 14, 1, 2, 3, 5, 6, 7, 4, 12, 10, 11, 9, 13, 0, 15]	8	9	G_9	69	28	14	5	1	0	**10**
10	[8, 14, 1, 2, 3, 5, 6, 7, 4, 9, 15, 11, 12, 13, 0, 10]	8	10	G_{14}	70	27	13	6	1	0	**10**
11	[8, 15, 1, 2, 3, 5, 12, 7, 4, 9, 10, 11, 6, 13, 14, 0]	8	11	G_{15}	70	27	13	6	1	0	**10**
12	[8, 15, 1, 2, 3, 5, 6, 13, 4, 9, 10, 11, 12, 7, 14, 0]	8	12	G_{10}	69	30	12	3	3	0	**10**
13	[12, 0, 1, 9, 3, 5, 4, 7, 6, 2, 10, 11, 8, 13, 14, 15]	8	13	G_2	64	32	8	12	0	2	**16**
14	[12, 11, 1, 2, 3, 5, 4, 7, 6, 9, 10, 0, 8, 13, 14, 15]	8	14	G_1	60	36	12	8	0	2	**16**
15	[12, 9, 1, 2, 3, 5, 4, 7, 6, 0, 10, 11, 8, 13, 14, 15]	8	15	G_8	72	32	0	16	0	2	**16**
16	[8, 14, 1, 2, 3, 5, 4, 7, 6, 9, 10, 0, 12, 13, 11, 15]	8	16	G_0	64	32	8	12	0	2	**16**
17	[8, 15, 1, 2, 3, 12, 6, 7, 4, 9, 10, 11, 5, 13, 14, 0]	12	34	–	57	35	14	0	7	0	**10**
18	[8, 0, 1, 12, 2, 5, 11, 7, 4, 9, 10, 6, 3, 13, 14, 15]	12	35	–	60	34	15	4	3	0	**10**
19	[8, 0, 1, 12, 2, 5, 13, 7, 4, 9, 10, 11, 3, 6, 14, 15]	12	36	–	60	34	15	4	3	0	**10**
20	[12, 0, 1, 2, 3, 15, 6, 7, 4, 9, 10, 11, 8, 13, 14, 5]	12	37	–	58	30	14	12	0	1	**16**
21	[12, 0, 1, 2, 3, 5, 6, 13, 4, 9, 10, 11, 8, 7, 14, 15]	12	38	–	62	36	8	10	2	1	**16**

Table 4.2 – Spectrum of the BCT for all 4-bit permutations with differential uniformity 4. Column 4 mentions the link with the functions of Table 5.2 in [De 07]. For the first 16 permutations, we also mention the corresponding equivalence class in the Leander-Poschmann classification [LP07].

A first important observation from Table 4.2 is that all 4-bit permutations with $\delta_S = 4$ have boomerang uniformity at least 6. This then proves that 4-bit S-boxes with boomerang uniformity 4 do not exist, as conjectured in [Cid+18, Section 6.1]. We can also observe from this same table that any two 4-bit permutations with $\delta_S = 4$ that are not related by inversion or affine equivalence have different BCT spectra.

Table 4.2 confirms several observations on 4-bit S-boxes reported in [Cid+18], which have been obtained experimentally by examining all S-boxes having specific properties. Most of these phenomena can actually be deduced from the following lemma which is very specific to the case of mappings over $\mathbb{F}_2^4$.

Lemma 2. *Let S be a permutation on $\mathbb{F}_2^4$ such that there exist $a, b_1, b_2 \in \mathbb{F}_2^4$ satisfying*

$\delta_S(a, b_1) = \delta_S(a, b_2) = 4$. *Then,*

$$\{S(x) : S(x) \oplus S(x \oplus a) = b_1\} \cap \{S(x) : S(x) \oplus S(x \oplus a) = b_2\} \neq \{0\} .$$

The previous lemma implies a strong relationship between the boomerang uniformity of a 4-bit S-box and the number of values 4 in a row of its DDT. This relationship explains for instance the fact observed in [Cid+18, Lemma 3]: if the DDT of a 4-bit S-box S has a row with entries 0 and 4 only, then $\beta_S = 16$. Lemma 2 permitted us to prove the following results.

Proposition 14. *Let S be a permutation of $\mathbb{F}_2^4$ with $\delta_s = 4$. Then,*

- *If its DDT has a row with at least two values 4, then $\beta_S \geq 8$;*
- *If each row in its DDT has at most two values 4, then $\beta_S \leq 10$;*
- *If its DDT has a row with four values 4, then $\beta_S = 16$.*

Note that this result does not hold anymore for higher even dimensions.

4.2.3 BCT of the Inverse Mapping Over $\mathbb{F}_{2^n}$ and Quadratic Differentially 4-Uniform Permutations

In our work we managed to determine the boomerang properties for the inverse mapping. In odd dimension this permutation is known to be APN while in even dimension it is differentially 4-uniform. An important result we demonstrated is that the differential uniformity of the inverse mapping is 6 for $n \equiv 0 \mod 4$ and equals 4 for $n \equiv 2 \mod 4$. This proved notably that the inverse mapping has an optimal BCT in such dimensions for a non-APN S-box and solved for $n \equiv 2 \mod 4$, the open problem raised in [Cid+18].

Proposition 15. *Let S be the inverse mapping over $\mathbb{F}_{2^n}$ for n even. Then:*

$$\beta_S = \begin{cases} 4, & \text{if} \quad n \equiv 2 \mod 4 \\ 6, & \text{if} \quad n \equiv 0 \mod 4 \end{cases}$$

Moreover, for any nonzero a, b, we have

$$\beta_S(a, b) = \begin{cases} \beta_S & \text{if } b \in \{a^{-1}\omega, a^{-1}(\omega \oplus 1)\} \\ \delta_S(a, b) & \text{otherwise} \end{cases}$$

where ω is any element in $\mathbb{F}_4 \setminus \mathbb{F}_2$, i.e. any element[1] *in $\mathbb{F}_{2^n} \setminus \mathbb{F}_2$ such that $\omega^3 = 1$.*

We then focused on quadratic permutations as these objects present a particular interest for cryptography. As for an even number of variables, APN quadratic permutations do not exist [Nyb94], $\delta_S = 4$ is the lowest differential uniformity that a quadratic permutation can achieve in this case. We proved first that the maximal value in the BCT of differentially 4-uniform quadratic permutations is at most 12.

Proposition 16. *Let S be a quadratic permutation of $\mathbb{F}_2^n$ with differential uniformity 4. Then $\beta_S \leq 12$.*

We next showed that some of the so-called Gold power permutations [Gol68], which are differentially 4-uniform quadratic permutations of $\mathbb{F}_2^n$, have an optimal BCT when $n \equiv 2 \bmod 4$, i.e. they satisfy $\beta_S = 4$.

Proposition 17. *Let $n \equiv 2 \bmod 4$ and let t be an even integer such that $\gcd(t, n) = 2$. Then $S : x \mapsto x^{2^t+1}$ over $\mathbb{F}_{2^n}$ is a differentially 4-uniform permutation and satisfies $\beta_S = 4$.*

As an example of the previous proposition, we get that $S : x \mapsto x^5$ over $\mathbb{F}_{2^n}$ with $n \equiv 2 \bmod 4$ is an S-box with optimal BCT in the sense of [Cid+18] since it satisfies $\delta_S = \beta_S = 4$. It is worth noticing that, for these dimensions, these quadratic differentially 4-uniform permutations have the same behaviour as the inverse mapping, while their DDTs are very different. Indeed, the DDTs of the quadratic mappings in Proposition 17 consist of 0 and 4 only, while the DDT of the inverse mapping has a single 4 in each row.

4.2.4 Conclusion

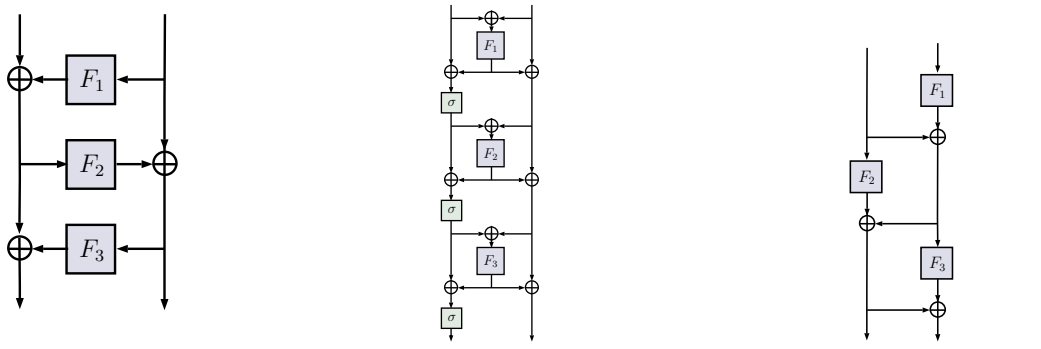
Soon after the seminal paper of Cid et al. [Cid+18] and the exhibition in our paper of the first families of S-boxes in dimension $n \equiv 2 \bmod 4$ with both differential and boomerang uniformity equal to 4, many subsequent works tried to exhibit more such (not APN) S-boxes showing an ideal resistance against the boomerang attack. However, despite the different efforts, only six families of 4-differential uniform permutations with boomerang uniformity 4 are known today [Li+19; MTX20; Li+21; LXZ21; Tu+20]. Moreover, all of them are permutations of $\mathbb{F}_2^n$ with $n \equiv 2 \bmod 4$. Up to now, no permutations

1. It is worth noticing that there are two elements in $\mathbb{F}_4 \setminus \mathbb{F}_2$, ω and $\omega' = \omega \oplus 1$, and both of them obviously lead to the same condition.

with $n \equiv 0 \mod 4$ providing optimal resistance ($\delta_S = \beta_S \in \{2, 4\}$) against boomerang cryptanalysis are known and this seems to be today a quite difficult open problem, related notably to the fact that no APN and only very few 4-uniform permutations are known in these dimensions, for example in $\mathbb{F}_2^8$.

4.3 Boomerang Uniformity of Popular S-box Constructions

A popular way of constructing large S-boxes is by building them using smaller components and some block-based structure. This approach has obvious implementation advantages, as it is much easier and cheaper to implement few layers of a small (e.g. a 4-bit) S-box than a large (e.g. a 8-bit) one without some particular structure. In a joint article with Shizhu Tian and Léo Perrin [TBP20] we studied the boomerang uniformity of some popular S-box constructions of this type. More particularly, we focused on the 3-round Feistel, Lai-Massey and (unbalanced) MISTY structures. These three classical constructions are depicted in Figures 4.1a, 4.1b and 4.1c respectively. We also looked at two 1-round structures: the 1-round SPN and the specific structure used in the `FLY` block cipher [KG16].



(a) 3-round Feistel. (b) 3-round Lai-Massey. (c) 3-round MISTY.

Figure 4.1 – The structures investigated in this paper.

4.3.1 3-Round Feistel, Lai-Massey and MISTY Networks

We studied first the properties of the BCT of 3-round Feistel and Lai-Massey constructions and showed that the boomerang uniformity of both is the worst one possible.

Rounds	S-box Struct.	Cipher	Ref.	δ_S	β_S	Lower bound
3	Feistel	`Scream`	[Gro+14b; CDL15]	8	**256**	256
	MISTY-like	`Fantomas`	[Gro+14a]	16	**160**	64 (∗)
	Lai-Massey	`Fox`	[VJ04]	16	**256**	256
1	SPN	`Midori`	[Ban+15b]	64	**256**	256
	Lai-Massey-like	`FLY`	[KG16]	16	**256**	256 (∗)

Table 4.3 – The boomerang and differential uniformity of various 8-bit S-boxes. (∗) The bound depends on the inner components of these constructions.

In both cases, the results we derived are an inherent property of the structures used: even if the subcomponents are chosen so as to have excellent properties, the boomerang uniformity will be the worst possible.

The MISTY network mimicks a structure used in the `MISTY` cipher [Mat97]. While it might resemble a Feistel network it requires all three inner functions to be bijective in order for the whole function to be a permutation. In our work, we showed that the MISTY structure also differs from the Feistel one via its BCT: the boomerang uniformity of a 3-round MISTY structure *depends* on the specifics of the subfunctions used. Proposition 18 shows a lower bound on the boomerang uniformity of a balanced 3-round MISTY construction.

Proposition 18. *Let $S : (\mathbb{F}_2^n)^2 \to (\mathbb{F}_2^n)^2$ be a 3-round balanced MISTY network and let F_1, F_2 and F_3 be its inner functions as depicted in Fig. 4.1c, with $F_1, F_2, F_3 : \mathbb{F}_2^n \to \mathbb{F}_2^n$ bijective. Then,*

$$\beta_S \geq 2^n \beta_{F_2}.$$

The results for the unbalanced case are similar but need a different treatment.

A natural question regarding Proposition 18 is what is the smallest possible boomerang uniformity for a 3-round MISTY network for some popular choices of n if all functions F_1, F_2 and F_3 are supposed bijective.

$\mathbf{n = 4}$ As we proved in [BC18] (see Section 4.2), the minimal boomerang uniformity for a permutation of $\mathbb{F}_2^4$ is 6. Therefore, by choosing F_2, with $\beta_{F_2} = 6$, we get that $\beta_S \geq 96$. As we will argue later, this is a very high value for an 8-bit S-box. Note here that the bound of Proposition 18 is tight. Indeed, the 8-bit MISTY network with $F_1 = F_2 = F_3 = [\mathtt{8}, \mathtt{0}, \mathtt{1}, \mathtt{12}, \mathtt{15}, \mathtt{5}, \mathtt{6}, \mathtt{7}, \mathtt{4}, \mathtt{3}, \mathtt{10}, \mathtt{11}, \mathtt{9}, \mathtt{13}, \mathtt{14}, \mathtt{2}]$, where $\beta_{F_2} = 6$, has boomerang uniformity 96 and reaches thus this lower bound.

n = 3 APN permutations exist for $n = 3$. Using one as F_2, we obtain $\beta_S \geq 16$.

We showed also that if one of the F_1 or F_3 is linear then the boomerang uniformity gets the worst one possible.

Proposition 19. *Let $S : (\mathbb{F}_2^n)^2 \to (\mathbb{F}_2^n)^2$ be a 3-round MISTY network with inner functions F_1, F_2 and F_3. If F_1 or F_3 is an affine permutation, then $\beta_S = 2^{2n}$.*

4.3.2 Non-Iterative Constructions

We equally analysed 1-round SPN constructions (as used e.g. in `Midori` [Ban+15b]) and at the *ad hoc* Lai-Massey-like structure of the Littlun S-Box inside the block cipher `FLY` [KG16]. It is composed of a single Lai-Massey round followed by an S-box layer (see Fig. 4.2). While `Littlun` is such that $F_1 = F_2 = F_3$, we did not make this assumption in our analysis.

The following straightforward proposition deals with the properties of a 1-round SPN.

Proposition 20. *Let F_1, F_2 be n-bit permutations and let $S : (\mathbb{F}_2^n)^2 \to (\mathbb{F}_2^n)^2$ be such that $S(x_1, x_2) = (F_1(x_1), F_2(x_2))$. Then we have*

$$\beta_S\left((a_1, a_2), (b_1, b_2)\right) \;=\; \beta_{F_1}(a_1, b_1) \times \beta_{F_2}(a_2, b_2)\ ,$$

so that in particular $\beta_S = \beta_S\left((a, 0), (0, b)\right) = 2^{2n}$. □

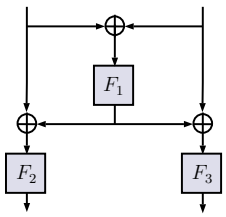


Figure 4.2 – Littlun S-box.

We then considered the Littlun construction (see Figure 4.2). If F_1 is an affine permutation, then the corresponding Littlun-like S-box is a 1-round SPN structure which has thus the worst possible boomerang uniformity. If not, its boomerang uniformity depends on its subcomponents as we proved in the following proposition.

Proposition 21. *Let S be a generalised-Littlun structure with n-bit permutations F_1, F_2 and F_3 (see Fig. 4.2). Then $\beta_S \geq 2^n \max\{\beta_{F_2}, \beta_{F_3}\}$.*

4.3.3 Analysis of the Obtained Results

All the S-boxes we studied were highly structured, it is then not surprising that we found specific artefacts in their BCT. However, it was important for us to understand how the obtained boomerang uniformities compared to the expected value of the boomerang uniformity of a random n-permutation. Based on the analysis of Bonnetain et al. [BPT19] on the expected behaviour of the BCT of a random permutation, we derived a formula to compute the expected boomerang uniformity for an n-bit random permutation, namely:

$$E(\beta_s) \;=\; \sum_{c=0}^{2^n} c \left(\left(\sum_{i=0}^{c} \Pr(\beta_S(a,b)=i) \right)^t - \left(\sum_{i=0}^{c-2} \Pr(\beta_S(a,b)=i) \right)^t \right) \;,$$

where $t = (2^n-1)^2$ and used this formula to build a table containing the expected boomerang uniformity for random permutations from $n = 4$ to 14 (see Table 4.4).

n	4	5	6	7	8	9	10	11	12	13	14
$E(\beta_s)$	11.6	14.2	16.3	18.3	20.2	22.1	23.9	25.7	27.4	29.1	30.8

Table 4.4 – The expected boomerang uniformity for n-bit random permutations.

As we can see, the expected boomerang uniformity increases slowly with n and it is significantly smaller than 2^n. This shows that a maximal boomerang uniformity is an extremely rare event indicative of a very strong structure. Furthermore, the non-maximal but still very high boomerang uniformity of the 3-round MISTY and Littlun structures can also be leveraged to identify such potentially hidden structures. Indeed, for $n = 8$, it holds that the boomerang uniformity of a balanced MISTY structure is at least 96 which is much higher than the expected 20.2.

In parallel, we showed that the coordinates of the BCT coefficients of a permutation S of $\mathbb{F}_2^n$ that are equal to the maximum possible value (namely 2^n) always have a particular structure. Indeed, we proved that the sets

$$\left\{ y \in \mathbb{F}_2^n, \beta_S(y,x) = 2^n \right\} \text{ and } \left\{ y \in \mathbb{F}_2^n, \beta_S(x,y) = 2^n \right\} . \tag{4.3}$$

are vector spaces. These properties are reminiscent of those of the *linear structures* of a Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$, i.e. all elements $a \in \mathbb{F}_2^n$ such that $f(x \oplus a) \oplus f(x)$ is constant for all $x \in \mathbb{F}_2^n$.

This likeness is not a coincidence. Indeed, for a fixed output difference b, the maximum

BCT coefficients correspond to a such that $S_b(x \oplus a) \oplus S_b(x) = a$ for all $x \in \mathbb{F}_2^n$, i.e. essentially to linear structures that are shared by all the coordinates of S_b, where S_b is the mapping $x \mapsto S^{-1}(S(x) + b)$.

These remarks can be used combined with other tools (e.g. [BPT19]), to determine whether a given permutation is affine-equivalent to a 3-round Feistel or Lai-Massey network.

4.3.4 An Algorithm for Inverting a Given BCT

A final problem we treated in [TBP20], was to apply to BCTs a similar approach with what was done with DDTs in Section 4.1. Indeed, we tried to answer the two following questions:

1. What can we say about S-boxes that share the same BCT?
2. How can we compute all S-boxes that share the same BCT?

Finding an algorithm to answer the second question was notably stated as an open problem in [BC18] and [DH19].

To study these problems we introduced the notions of *BCT-equivalence* and *BCT-equivalence class* for permutations that share the same BCT. A first result we showed is that Proposition 9 for DDTs can be entirely adapted to the BCT case and that the BCT-equivalence class of a permutation F of $\mathbb{F}_2^n$ contains at least all the permutations of the form $x \mapsto F(x \oplus c) \oplus d$ with $c, d \in \mathbb{F}_2^n$. We called all permutations of this form *trivially BCT-equivalent*. An obvious remark is that the bound on the size of equivalence classes is the same for both DDT and BCT equivalences. However, it is important to note that despite this, the two classes for a given permutation can be very different. Indeed, for $n = 6$, the DDT-equivalence class of the quadratic permutation $x \mapsto x^5$ contains 2^{12} trivial permutations, while its BCT-equivalence class is much larger.

Another result that we showed to naturally adapt from the DDT to the BCT case concerns the invariance of the sizes of the equivalence classes under affine equivalence. Indeed, it is not complicated to show that the size of the BCT-equivalence classes is preserved under affine equivalence. The importance of this result is principally computational, as it shows that it is sufficient to compute the size of a BCT-equivalence class for only one representative of the affine equivalence class.

Relation Between DDT and BCT-Equivalence Classes

Another natural question was if something interesting could be said about the relation of the DDT and BCT-equivalence classes. To make a step towards this direction, we computed and analyzed all DDT and BCT-equivalence classes for 3-bit permutations. Our experiments showed that for $n = 3$ there are 924 DDT-equivalence classes and 512 BCT-equivalence classes. All DDT-equivalence classes are trivial, but not the BCT-equivalence ones. Most importantly, these computations showed us that two permutations F and G that are BCT-equivalent are not necessarily DDT-equivalent. Moreover, for a permutation F of $\mathbb{F}_2^3$ we always have that the DDT-equivalence class of F is a subset of the BCT-equivalence class of F, with equality occurring for some classes. However, for higher dimensions this inclusion does not always hold. For example, the two functions of Table 4.5 are (non-trivially) DDT-equivalent but have different BCTs. More precisely, the DDT-equivalence class of these two permutations has $7 \cdot 2^{11}$ members inside. These permutations can be partitioned into 28 groups of 2^9 permutations each, each group belonging a different BCT class. Furthermore, each of these BCT classes has also $7 \cdot 2^{11}$ permutations inside, coming here also from 28 different DDT-equivalence classes. This last example shows that the relation between DDT and BCT-equivalences is not trivial and needs further investigation.

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$F(x)$	0	1	2	3	4	5	6	7	8	9	10	11	13	12	15	14
$F'(x)$	0	1	2	3	4	5	6	7	8	9	10	11	13	12	15	14
x	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
$F(x)$	16	17	19	18	20	21	23	22	25	24	26	27	28	29	31	30
$F'(x)$	16	17	19	18	21	20	22	23	24	25	27	26	28	29	31	30

Table 4.5 – Two non-trivially DDT-equivalent permutations with different BCTs.

An Algorithm for Computing BCT-equivalence Classes and Experiments

We finally provided an algorithm that computes and returns the BCT-equivalence class of permutations of a given table B. Our algorithm implements a tree-traversal search where each level i of the tree corresponds to the possible values of $S(i)$. Its structure recalls the one given in Section 4.1.2 for recovering the DDT-equivalence class but the

inner details remain quite different, because of the differences in the nature of these two tables. More precisely, the main difference in the two algorithms is the subroutine that at each level i computes and stores in a list all possible values for $S(i)$ by taking into account values for S computed at a higher level of the search (notably the values $S(0), \ldots, S(i-1)$) and the constraints imposed by the input table B. Indeed, contrary to the DDT, both S and its inverse S^{-1} are needed in the original definition of the BCT. However, by using an equivalent definition of $\beta(a, b)$ that sees this quantity as the number of solutions of a system of equations involving only S and not its inverse, the above problem can be bypassed.

Experiments

Using the above algorithm we were able to compute the BCT-equivalence classes of many known permutations. We notably computed the BCT-equivalence classes for all 4-bit S-boxes. Our algorithm shows that among the 302 affine-equivalence classes (see the classification of [De 07]), 280 are trivial. Among the other 22 non-trivial classes all of them have differential uniformity $\delta_S \geq 8$. This means that normally, all S-boxes used in practice (i.e. having $\delta_S = 4, 6$) have trivial BCT-equivalence classes. Another remark concerns quadratic permutations. These permutations are particularly interesting as many different classes of quadratic permutations were shown to have an optimal boomerang uniformity [BC18; Li+19; MTX20]. There are 6 affine-equivalent quadratic classes for $n = 4$ and all of them have non-trivial BCT-equivalence classes. Finally, we showed that all these non-trivial BCT-equivalence classes are larger or equal than the corresponding DDT-equivalence classes. Finally, we computed the BCT and DDT-equivalence classes for many 8-bit S-boxes from the literature. For all of them we have found both DDT and BCT-equivalence classes to be trivial.

4.3.5 Conclusion

Our results on the BCT and on the boomerang uniformity of permutations with various structures have several consequences. First, by the fact that $\beta_S = 2$ if and only if S is APN, we can immediately see that 3-round Feistel, Lai-Massey and MISTY structures can never be APN.

The consequences in terms of cryptanalysis can also extend further than boomerang attacks. The guess and determine of Biryukov et al. [BLP15] uses a property equivalent to

the fact that the boomerang uniformity of a 3-round Feistel network is always maximal. We can therefore expect the same attack to work against Lai-Massey structures. Interestingly, our results show that it is possible to construct a 3-round MISTY structure immune against the existence of such probability 1 patterns, meaning that they seem to offer some inherent resilience against these attacks. Not only are our results regarding Feistel and Lai-Massey structures very similar, the arguments we used to derive them are also very close. While the similarity between these two structures makes intuitive sense, we find it interesting to see it displayed in such a clear manner.

Another application of our results lies in S-box reverse-engineering [BP15]. In this context, the aim is to recover the hidden structure of an S-box using only its lookup table. If an S-box has non-trivial differential and linear properties but a boomerang uniformity equal to 2^n then we can suspect that it is a 3-round Lai-Massey or Feistel structure. Since the boomerang uniformity is preserved under the composition with an affine permutation, this test would work even if the S-box structure is obfuscated by such permutations—as is the case for instance in the S-box of `ZUC`.

Finally, the initial analysis of the BCT-equivalence classes problem that we provided, gives rise to many open questions. For example, for a permutation F, is the cardinality of a BCT-equivalence class always higher or equal to the size of the corresponding DDT-equivalence class? Further, can we derive any bounds on the size of the BCT-equivalence classes for quadratic permutations? Finally, an interesting direction is to further investigate the relation between the two equivalence notions.

Chapter 5

Conclusion and Perspectives

My research during these last 10 years was centered around understanding the security offered by symmetric primitives, by analyzing classical and more recent attacks and by studying the properties of S-boxes, the small-sized functions whose role is to ensure the non-linearity of the global constructions. I would like to continue working on the security of symmetric primitives by investigating some of the axes described below.

One of my very recent cryptanalysis results was the break of one of the variants of `SPEEDY` [Bou+23] by means of differential cryptanalysis. Our work highlighted the fact that finding an optimal strategy for the key recovery phase of these attacks is a difficult task. In the close future I would like to analyze this phase of differential attacks closer, propose a unified method to treat it and design a generic algorithm to recover the key once a promising differential has been found. A second step would be to provide a tool that would search for the distinguisher leading to the best attack, by incorporating the key recovery algorithm described above.

In another recent work we proposed the differential meet-in-the-middle (MITM) cryptanalysis technique, a new attack combining a differential distinguisher together with a MITM approach to find the secret key. This technique is new and seems very promising. It would be interesting to see if it can be applied in other contexts, for the cryptanalysis of hash functions for example or if the same approach could be combined with other attacks.

The starting point of differential meet-in-the-middle attacks but also of classical differential attacks is the existence of a high-probability differential trail. The search of such trails got boosted in the recent years by the publication of many dedicated algorithms and tool-based approaches. My works on the modelization of SPN ciphers for MILP [BC20] and the search of differential bounds for `Troika` [BFR22] gave me the desire to continue working in this direction by exploring new strategies. From one side I would like to see at which point strategies (e.g. dynamic programming on partial states) other that CP, SAT

or MILP could be applied to classical SPN ciphers such as `AES` or `SKINNY` in the single and related-key context. On the other side I would like to work on a generic strategy for searching for trails for weakly-aligned constructions working on large states, as for example `Keccak`.

Finally, I was always amazed by the field of APN permutations and the combinatorial difficulty of these optimal objects. An idea I would like to explore is to what extent it is possible to combine our knowledge on the mathematical properties of these functions with progresses on tool-based search.

List of Publications

Journal Articles

1. Christina Boura and Daniel Coggia. **Efficient MILP Modelings for Sboxes and Linear Layers of SPN ciphers**. IACR Transactions on Symmetric Cryptology, 2020(3), 327-361.
2. Christina Boura, Nicolas Gama, Mariya Georgieva and Dimitar Jetchev. **CHIMERA: Combining Ring-LWE-based Fully Homomorphic Encryption Schemes**. Journal of Mathematical Cryptology 14(1): 316-338 (2020).
3. Christina Boura, Eirini Chavli and Maria Chlouveraki. **The BMM symmetrising trace conjecture for the exceptional 2-reflection groups of rank 2**. Journal of Algebra 558, Special issue in honor of Michel Broué (2020), 176-198.
4. Christina Boura, Eirini Chavli, Maria Chlouveraki and Konstantinos Karvounis. **The BMM symmetrising trace conjecture for groups G4, G5, G6, G7, G8**. **Journal of Symbolic Computations**. 96: 62-84 (2020).
5. Christina Boura, Anne Canteaut and Daniel Coggia. **A General Proof Framework for Recent AES Distinguishers**. IACR Transactions on Symmetric Cryptology, 2019(1): 170-191.
6. Christina Boura, Anne Canteaut, Jérémy Jean and Valentin Suder. **Two Notions of Differential Equivalence on Sboxes**. Designs, Codes and Cryptography, 87(2), 185-202, 2019.
7. Christina Boura and Anne Canteaut. **On the Boomerang Uniformity of Cryptographic Sboxes**. IACR Transactions on Symmetric Cryptology, 2018(3), 290-310.
8. Christina Boura, Virginie Lallemand, Maria Naya-Plasencia and Valentin Suder. **Making the impossible possible**. Journal of Cryptology 31(1): 101-133, 2018.
9. Christina Boura, Anne Canteaut, Lars R. Knudsen and Gregor Leander. **Reflection ciphers**. Designs, Codes and Cryptography 82(1-2): 3-25, 2017.

10. Christina Boura and Anne Canteaut. **On the influence of the algebraic degree of F^{-1} on the algebraic degree of $G \circ F$**. IEEE Transactions on Information Theory, 59(1): 691-702, 2013.

Conference and Workshop Papers

1. Christina Boura, Nicolas David, Rachelle Heim Boissier and María Naya-Plasencia. **Better Steady than Speedy: Full break of SPEEDY-7-192**. EUROCRYPT 2023, to appear.
2. Christina Boura, Margot Funk and Yann Rotella. **Differential analysis of the ternary hash function Troika**. SAC 2022, to appear.
3. Christina Boura, Rachelle Heim Boissier and Yann Rotella. **Breaking Panther**. AFRICACRYPT 2022, volume 13503 of *Lecture Notes in Computer Science*, pages 176-188, 2022.
4. Christina Boura, Nicolas Gama, Mariya Georgieva and Dimitar Jetchev. **Simulating Homomorphic Evaluation of Deep Learning Predictions**. Cyber Security Cryptography and Machine Learning - Third International Symposium, CSCML 2019, volume 11527 of *Lecture Notes in Computer Science*, pages 212-230, 2019.
5. Christina Boura, Ilaria Chillotti, Nicolas Gama, Dimitar Jetchev, Stanislav Peceny and Alexander Petric. **High-Precision Privacy-Preserving Real-Valued Function Evaluation**. In Financial Cryptography 2018, volume 10957 of *Lecture Notes in Computer Science*, 2019.
6. Christina Boura and Anne Canteaut. **Another view of the division property**. CRYPTO 2016, volume 9814 of *Lecture Notes in Computer Science* (Part I), pages 654-682, 2016.
7. Christina Boura, Avik Chakraborti, Gaëtan Leurent, Goutam Paul, Dhiman Saha, Hadi Soleimany and Valentin Suder. **Key Recovery Attack against 2.5-round π-Cipher**. In FSE 2016, volume 8783 of *Lecture Notes in Computer Science*, pages 535-553, 2016.
8. Christina Boura, María Naya-Plasencia and Valentin Suder. **Scrutinizing and Improving Impossible Differential Attacks: Applications to CLE-**

FIA, Camellia, LBlock and Simon. ASIACRYPT 2014, volume 8873 of *Lecture Notes in Computer Science*, pages 179-199, 2014.

9. Andrey Bogdanov, Christina Boura, Vincent Rijmen, Meiqin Wang, Long Wen and Jingyuan Zhao. **Key Difference Invariant Bias in Block Ciphers**. ASIACRYPT 2013, volume 8269 of *Lecture Notes in Computer Science*, pages 357-376, 2013.

10. Christina Boura and Anne Canteaut. **A new criterion for avoiding the propagation of linear relations through an Sbox.** FSE 2013, volume 8424 of *Lecture Notes in Computer Science*, pages 585-604, 2013.

11. Christina Boura, Sylvain Lévêque and David Vigilant. **Side-Channel Analysis of Grøstl and Skein**. Proceedings of the 2nd International Workshop on Trustworthy Embedded Devices-TrustED 2012, IEEE Symposium on Security and Privacy, pages 6-26, May 2012.

12. Christina Boura, Anne Canteaut and Christophe De Cannière. **Higher-Order Differential Properties of Keccak and Luffa**. FSE 2011, volume 6733 of *Lecture Notes in Computer Science*, pages 252-269, 2011.

13. Christina Boura and Anne Canteaut. **Zero-Sum Distinguishers for Iterated Permutations and Application to Keccak-f and Hamsi-256**. SAC 2010, volume 6544 of *Lecture Notes in Computer Science*, pages 1-17 2010.

14. Christina Boura and Anne Canteaut. **A zero-sum property for the Keccak-f permutation with 18 rounds**. International Symposium on Information Theory-ISIT 2010, pages 2488-2492, IEEE 2010.

Preprints

1. Christina Boura, Nicolas David, Patrick Derbez, Gregor Leander and María Naya-Plasencia. **Differential Meet-In-The-Middle Cryptanalysis**.

BIBLIOGRAPHY

[Abd+17] Ahmed Abdelkhalek, Yu Sasaki, Yosuke Todo, Mohamed Tolba, and Amr M. Youssef, « MILP Modeling for (Large) S-boxes to Optimize Probability of Differential Characteristics », *in*: *IACR Trans. Symmetric Cryptol.* 2017.*4* (2017), pp. 99–129.

[AL13] Hoda AlKhzaimi and Martin M. Lauridsen, « Cryptanalysis of the SIMON Family of Block Ciphers », *in*: *IACR Cryptol. ePrint Arch.* (2013), p. 543.

[AS08] Kazumaro Aoki and Yu Sasaki, « Preimage Attacks on One-Block MD4, 63-Step MD5 and More », *in*: *SAC 2008*, vol. 5381, LNCS, Springer, 2008, pp. 103–119.

[AS09] Kazumaro Aoki and Yu Sasaki, « Meet-in-the-Middle Preimage Attacks Against Reduced SHA-0 and SHA-1 », *in*: *CRYPTO 2009*, vol. 5677, LNCS, Springer, 2009, pp. 70–89.

[Aum+09] Jean-Philippe Aumasson, Itai Dinur, Willi Meier, and Adi Shamir, « Cube Testers and Key Recovery Attacks on Reduced-Round MD6 and Trivium », *in*: *FSE 2009*, vol. 5665, LNCS, Springer, 2009, pp. 1–22.

[Ban+15a] Subhadeep Banik, Andrey Bogdanov, Takanori Isobe, Kyoji Shibutani, Harunaga Hiwatari, Toru Akishita, and Francesco Regazzoni, « Midori: A Block Cipher for Low Energy », *in*: *ASIACRYPT 2015, Part II*, vol. 9453, LNCS, Springer, 2015, pp. 411–436.

[Ban+15b] Subhadeep Banik, Andrey Bogdanov, Takanori Isobe, Kyoji Shibutani, Harunaga Hiwatari, Toru Akishita, and Francesco Regazzoni, « Midori: A Block Cipher for Low Energy », *in*: *ASIACRYPT 2015, Part II*, vol. 9453, LNCS, Springer, 2015, pp. 411–436.

[Ban+17] Subhadeep Banik, Sumit Kumar Pandey, Thomas Peyrin, Yu Sasaki, Siang Meng Sim, and Yosuke Todo, « GIFT: A Small Present - Towards Reaching the Limit of Lightweight Encryption », *in*: *CHES 2017*, vol. 10529, LNCS, Springer, 2017, pp. 321–345.

[Bar+18] Achiya Bar-On, Orr Dunkelman, Nathan Keller, Eyal Ronen, and Adi Shamir, « Improved Key Recovery Attacks on Reduced-Round AES with Practical Data and Memory Complexities », *in*: *CRYPTO 2018, Part II*, vol. 10992, LNCS, Springer, 2018, pp. 185–212.

[BBS99] Eli Biham, Alex Biryukov, and Adi Shamir, « Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials », *in*: *EUROCRYPT 1999*, vol. 1592, LNCS, Springer, 1999, pp. 12–23.

[BC16] Christina Boura and Anne Canteaut, « Another View of the Division Property », *in*: *CRYPTO 2016, Part I*, vol. 9814, LNCS, Springer, 2016, pp. 654–682.

[BC18] Christina Boura and Anne Canteaut, « On the Boomerang Uniformity of Cryptographic Sboxes », *in*: *IACR Trans. Symmetric Cryptol.* 2018.*3* (2018), pp. 290–310.

[BC20] Christina Boura and Daniel Coggia, « Efficient MILP Modelings for Sboxes and Linear Layers of SPN ciphers », *in*: *IACR Trans. Symmetric Cryptol.* 2020.*3* (2020), pp. 327–361.

[BCC19] Christina Boura, Anne Canteaut, and Daniel Coggia, « A General Proof Framework for Recent AES Distinguishers », *in*: *IACR Trans. Symmetric Cryptol.* 2019.*1* (2019), pp. 170–191.

[BDD03] Alex Biryukov, Christophe De Cannière, and Gustaf Dellkrantz, « Cryptanalysis of SAFER++ », *in*: *CRYPTO 2003*, vol. 2729, LNCS, Springer, 2003, pp. 195–211.

[BDK01] Eli Biham, Orr Dunkelman, and Nathan Keller, « The Rectangle Attack - Rectangling the Serpent », *in*: *EUROCRYPT 2001*, vol. 2045, LNCS, Springer, 2001, pp. 340–357.

[BDK02] Eli Biham, Orr Dunkelman, and Nathan Keller, « New Results on Boomerang and Rectangle Attacks », *in*: *FSE 2002*, vol. 2365, LNCS, Springer, 2002, pp. 1–16.

[Bei+16] Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim, « The SKINNY Family of Block Ciphers and Its Low-Latency Variant MANTIS », *in*: *CRYPTO 2016, Part II*, vol. 9815, LNCS, Springer, 2016, pp. 123–153.

[BFR22] Christina Boura, Margot Funk, and Yann Rotella, « Differential analysis of the ternary hash function Troika », *in*: *SAC 2022*, LNCS, To appear, Springer, 2022.

[BG11] Céline Blondeau and Benoıt Gérard, « Multiple Differential Cryptanalysis: Theory and Practice », *in*: *FSE 2011*, vol. 6733, LNCS, Springer, 2011, pp. 35–54.

[BK09] Alex Biryukov and Dmitry Khovratovich, « Related-Key Cryptanalysis of the Full AES-192 and AES-256 », *in*: *ASIACRYPT 2009*, vol. 5912, LNCS, Springer, 2009, pp. 1–18.

[BKR11] Andrey Bogdanov, Dmitry Khovratovich, and Christian Rechberger, « Biclique Cryptanalysis of the Full AES », *in*: *ASIACRYPT 2011*, vol. 7073, LNCS, Springer, 2011, pp. 344–371.

[BL08] Marcus Brinkmann and Gregor Leander, « On the classification of APN functions up to dimension five », *in*: *Des. Codes Cryptogr.* 49.*1-3* (2008), pp. 273–288.

[BLP15] Alex Biryukov, Gaëtan Leurent, and Léo Perrin, « Cryptanalysis of Feistel Networks with Secret Round Functions », *in*: *SAC 2015*, vol. 9566, LNCS, Springer, 2015, pp. 102–121.

[BNS14] Christina Boura, María Naya-Plasencia, and Valentin Suder, « Scrutinizing and Improving Impossible Differential Attacks: Applications to CLEFIA, Camellia, LBlock and Simon », *in*: *ASIACRYPT 2014, Part I*, vol. 8873, LNCS, Springer, 2014, pp. 179–199.

[Bog+07] 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*: *CHES 2007*, vol. 4727, LNCS, Springer, 2007, pp. 450–466.

[Bor+12] Julia Borghoff, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, Miroslav Knezevic, Lars R. Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, Peter Rombouts, Søren S. Thomsen, and Tolga Yalçin, « PRINCE - A Low-Latency Block Cipher for Pervasive Computing Applications - Extended Abstract », *in*: *ASIACRYPT 2012*, vol. 7658, LNCS, Springer, 2012, pp. 208–225.

[Bor+21] Nicolas Bordes, Joan Daemen, Daniël Kuijsters, and Gilles Van Assche, « Thinking Outside the Superbox », *in*: *CRYPTO 2021, Part III*, vol. 12827, LNCS, Springer, 2021, pp. 337–367.

[Bou+14] Christina Boura, Marine Minier, María Naya-Plasencia, and Valentin Suder, « Improved Impossible Differential Attacks against Round-Reduced LBlock », *in*: *IACR Cryptol. ePrint Arch.* (2014), p. 279.

[Bou+18] Christina Boura, Virginie Lallemand, María Naya-Plasencia, and Valentin Suder, « Making the Impossible Possible », *in*: *J. Cryptol.* 31.*1* (2018), pp. 101–133.

[Bou+19] Christina Boura, Anne Canteaut, Jérémy Jean, and Valentin Suder, « Two notions of differential equivalence on Sboxes », *in*: *Des. Codes Cryptogr.* 87.*2-3* (2019), pp. 185–202.

[Bou+22] Christina Boura, Nicolas David, Patrick Derbez, and María Naya-Plasencia, « Differential Meet-In-The-Middle Cryptanalysis », *in*: *IACR Cryptol. ePrint Arch.* (2022), p. 1640.

[Bou+23] Christina Boura, Nicolas David, Rachelle Heim Boissier, and María Naya-Plasencia, « Better Steady than Speedy: Full break of SPEEDY-7-192 », *in*: *EUROCRYPT 2023*, LNCS, to appear, Springer, 2023.

[BP15] Alex Biryukov and Léo Perrin, « On Reverse-Engineering S-Boxes with Hidden Design Criteria or Structure », *in*: *CRYPTO 2015, Part I*, vol. 9215, LNCS, Springer, 2015, pp. 116–140.

[BPT19] Xavier Bonnetain, Léo Perrin, and Shizhu Tian, « Anomalies and Vector Space Search: Tools for S-Box Analysis », *in*: *ASIACRYPT 2019, Part I*, vol. 11921, LNCS, Springer, 2019, pp. 196–223.

[Bra+84] Robert King Brayton, Alberto L. Sangiovanni-Vincentelli, Curtis T. McMullen, and Gary D. Hachtel, « Logic Minimization Algorithms for VLSI Synthesis », *in*: *Kluwer Academic Publishers* (1984).

[Bro+10] K.A. Browning, J.F. Dillon, M.T. McQuistan, and A.J. Wolfe, « An APN permutation in dimension six », *in*: *Finite Fields: Theory and Applications*, vol. 518, Contemporary Mathematics, AMS, 2010, pp. 33–42.

[BS90] Eli Biham and Adi Shamir, « Differential Cryptanalysis of DES-like Cryptosystems », *in*: *CRYPTO '90*, vol. 537, LNCS, Springer, 1990, pp. 2–21.

[CCZ98a] Claude Carlet, Pascale Charpin, and Victor Zinoviev, « Codes, Bent Functions and Permutations Suitable For DES-like Cryptosystems », *in*: *Des. Codes Cryptography* 15.*2* (1998), pp. 125–156.

[CCZ98b] Claude Carlet, Pascale Charpin, and Victor A. Zinoviev, « Codes, Bent Functions and Permutations Suitable For DES-like Cryptosystems », *in*: *Des. Codes Cryptogr.* 15.*2* (1998), pp. 125–156.

[CDL15] Anne Canteaut, Sébastien Duval, and Gaëtan Leurent, « Construction of Lightweight S-Boxes Using Feistel and MISTY Structures », *in*: *SAC 2015*, vol. 9566, LNCS, Springer, 2015, pp. 373–393.

[Cid+18] Carlos Cid, Tao Huang, Thomas Peyrin, Yu Sasaki, and Ling Song, « Boomerang Connectivity Table: A New Cryptanalysis Tool », *in*: *EUROCRYPT 2018, Part II*, vol. 10821, LNCS, Springer, 2018, pp. 683–714.

[CNV13] Anne Canteaut, María Naya-Plasencia, and Bastien Vayssière, « Sieve-in-the-Middle: Improved MITM Attacks », *in*: *CRYPTO 2013, Part I*, vol. 8042, LNCS, Springer, 2013, pp. 222–240.

[CR15] Anne Canteaut and Joëlle Roué, « On the Behaviors of Affine Equivalent Sboxes Regarding Differential and Linear Attacks », *in*: *EUROCRYPT 2015, Part I*, vol. 9056, LNCS, Springer, 2015, pp. 45–74.

[De 07] Christophe De Cannière, *Analysis and Design of Symmetric Encryption Algorithms*, PhD thesis, Katholieke Universiteit Leuven, 2007.

[Del+21] Stéphanie Delaune, Patrick Derbez, Paul Huynh, Marine Minier, Victor Mollimard, and Charles Prud'homme, « Efficient Methods to Search for Best Differential Characteristics on SKINNY », *in*: *ACNS 2021, Part II*, vol. 12727, LNCS, Springer, 2021, pp. 184–207.

[Der16] Patrick Derbez, « Note on Impossible Differential Attacks », *in*: *FSE 2016*, vol. 9783, LNCS, Springer, 2016, pp. 416–427.

[DF13] Patrick Derbez and Pierre-Alain Fouque, « Exhausting Demirci-Selçuk Meet-in-the-Middle Attacks Against Reduced-Round AES », *in*: *FSE 2013*, vol. 8424, LNCS, Springer, 2013, pp. 541–560.

[DFJ13] Patrick Derbez, Pierre-Alain Fouque, and Jérémy Jean, « Improved Key Recovery Attacks on Reduced-Round AES in the Single-Key Setting », *in*: *EUROCRYPT 2013*, vol. 7881, LNCS, Springer, 2013, pp. 371–387.

[DH19] Orr Dunkelman and Senyang Huang, « Reconstructing an S-box from its Difference Distribution Table », *in*: *IACR Trans. Symmetric Cryptol.* 2019.*2* (2019), pp. 193–217.

[DH77] Whitfield Diffie and Martin Hellman, « Special Feature Exhaustive Cryptanalysis of the NBS Data Encryption Standard », *in*: *Computer* 10.*6* (1977), pp. 74–84.

[Din+15] Itai Dinur, Yunwen Liu, Willi Meier, and Qingju Wang, « Optimized Interpolation Attacks on LowMC », *in*: *ASIACRYPT 2015,Part II*, vol. 9453, LNCS, Springer, 2015, pp. 535–560.

[DKS10] Orr Dunkelman, Nathan Keller, and Adi Shamir, « A Practical-Time Related-Key Attack on the KASUMI Cryptosystem Used in GSM and 3G Telephony », *in*: *CRYPTO 2010*, vol. 6223, LNCS, Springer, 2010, pp. 393–410.

[Don+21] Xiaoyang Dong, Jialiang Hua, Siwei Sun, Zheng Li, Xiaoyun Wang, and Lei Hu, « Meet-in-the-Middle Attacks Revisited: Key-Recovery, Collision, and Preimage Attacks », *in*: *CRYPTO 2021, Part III*, vol. 12827, LNCS, Springer, 2021, pp. 278–308.

[DS11] Itai Dinur and Adi Shamir, « Breaking Grain-128 with Dynamic Cube Attacks », *in*: *FSE 2011*, vol. 6733, LNCS, Springer, 2011, pp. 167–187.

[DSP07] Orr Dunkelman, Gautham Sekar, and Bart Preneel, « Improved Meet-in-the-Middle Attacks on Reduced-Round DES », *in*: *INDOCRYPT 2007*, vol. 4859, LNCS, Springer, 2007, pp. 86–100.

[Fei73] Horst Feistel, « Cryptography and computer privacy », *in*: *Scientific american* 228.*5* (1973), pp. 15–23.

[GNL12] Zheng Gong, Svetla Nikova, and Yee Wei Law, « KLEIN: A New Family of Lightweight Block Ciphers », *in*: *RFIDSec 2011*, vol. 7055, LNCS, Springer, June 2012, pp. 1–18.

[Gol68] Robert Gold, « Maximal recursive sequences with 3-valued recursive cross-correlation functions », *in*: *IEEE Trans. Information Theory* 14.*1* (1968), pp. 154–156.

[Gor16] Anastasiya Gorodilova, *On a remarkable property of APN Gold functions*, Cryptology ePrint Archive, Report 2016/286, 2016.

[Gor19] Anastasiya Gorodilova, « On the differential equivalence of APN functions », *in*: *Cryptogr. Commun.* 11.*4* (2019), pp. 793–813.

[Gra18] Lorenzo Grassi, « Mixture Differential Cryptanalysis: a New Approach to Distinguishers and Attacks on round-reduced AES », *in*: *IACR Trans. Symmetric Cryptol.* 2018.*2* (2018), pp. 133–160.

[Gro+14a] Vincent Grosso, Gaëtan Leurent, François-Xavier Standaert, and Kerem Varici, « LS-Designs: Bitslice Encryption for Efficient Masked Software Implementations », *in*: *FSE 2014*, vol. 8540, LNCS, Springer, 2014, pp. 18–37.

[Gro+14b] Vincent Grosso, Gaëtan Leurent, François-Xavier Standaert, Kerem Varici, François Durvaux Anthony Journault, Lubos Gaspar, and Stéphanie Kerckhof, *SCREAM & iSCREAM Side-Channel Resistant Authenticated Encryption with Masking*, Candidate for the CAESAR Competition. See also http://perso.uclouvain.be/fstandae/SCREAM/, 2014.

[GRR16] Lorenzo Grassi, Christian Rechberger, and Sondre Rønjom, « Subspace Trail Cryptanalysis and its Applications to AES », *in*: *IACR Trans. Symmetric Cryptol.* 2016.*2* (2016), pp. 192–225.

[GRR17] Lorenzo Grassi, Christian Rechberger, and Sondre Rønjom, « A New Structural-Differential Property of 5-Round AES », *in*: *EUROCRYPT 2017, Part II*, vol. 10211, LNCS, Springer, 2017, pp. 289–317.

[Guo+10] Jian Guo, San Ling, Christian Rechberger, and Huaxiong Wang, « Advanced Meet-in-the-Middle Preimage Attacks: First Results on Full Tiger, and Improved Results on MD4 and SHA-2 », *in*: *ASIACRYPT 2010*, vol. 6477, LNCS, Springer, 2010, pp. 56–75.

[Guo+11] Jian Guo, Thomas Peyrin, Axel Poschmann, and Matthew J. B. Robshaw, « The LED Block Cipher », *in*: *CHES 2011*, vol. 6917, LNCS, Springer, 2011, pp. 326–341.

[Hao+20] Yonglin Hao, Gregor Leander, Willi Meier, Yosuke Todo, and Qingju Wang, « Modeling for Three-Subset Division Property Without Unknown Subset - Improved Cube Attacks Against Trivium and Grain-128AEAD », *in*: *EUROCRYPT 2020, Part I*, vol. 12105, LNCS, Springer, 2020, pp. 466–495.

[Hao+21] Yonglin Hao, Gregor Leander, Willi Meier, Yosuke Todo, and Qingju Wang, « Modeling for Three-Subset Division Property without Unknown Subset », *in*: *J. Cryptol.* 34.*3* (2021), p. 22.

[Heb+20] Phil Hebborn, Baptiste Lambin, Gregor Leander, and Yosuke Todo, « Lower Bounds on the Degree of Block Ciphers », *in*: *ASIACRYPT 2020, Part I*, vol. 12491, LNCS, Springer, 2020, pp. 537–566.

[HSE22] Hosein Hadipour, Sadegh Sadeghi, and Maria Eichlseder, « Finding the Impossible: Automated Search for Full Impossible Differential, Zero-Correlation, and Integral Attacks (Preliminary Version) », *in*: *IACR Cryptol. ePrint Arch.* (2022), p. 1147.

[IS12] Takanori Isobe and Kyoji Shibutani, « All Subkeys Recovery Attack on Block Ciphers: Extending Meet-in-the-Middle Approach », *in*: *SAC 2012*, vol. 7707, LNCS, Springer, 2012, pp. 202–221.

[KDH12] Ferhat Karakoç, Hüseyin Demirci, and A. Emre Harmanci, « Impossible Differential Cryptanalysis of Reduced-Round LBlock », *in*: *WISTP 2012*, vol. 7322, LNCS, Springer, 2012, pp. 179–188.

[KG16] Pierre Karpman and Benjamin Grégoire, « The LITTLUN S-box and the FLY block cipher », *in*: *Lightweight Cryptography Workshop 2016, October 17-18 (informal proceedings)*, National Institute of Standards and Technology, 2016.

[Kim+04] Jongsung Kim, Seokhie Hong, Sangjin Lee, Junghwan Hwan Song, and Hyungjin Yang, « Truncated Differential Attacks on 8-Round CRYPTON », *in*: *ICISC 2003*, vol. 2971, LNCS, Springer, 2004, pp. 446–456.

[Kim+12] Jongsung Kim, Seokhie Hong, Bart Preneel, Eli Biham, Orr Dunkelman, and Nathan Keller, « Related-Key Boomerang and Rectangle Attacks: Theory and Experimental Analysis », *in*: *IEEE Trans. Inf. Theory* 58.*7* (2012), pp. 4948–4966.

[KKS00] John Kelsey, Tadayoshi Kohno, and Bruce Schneier, « Amplified Boomerang Attacks Against Reduced-Round MARS and Serpent », *in*: *FSE 2000*, vol. 1978, LNCS, Springer, 2000, pp. 75–93.

[Knu94] Lars R. Knudsen, « Truncated and Higher Order Differentials », *in*: *FSE '94*, vol. 1008, LNCS, Springer, 1994, pp. 196–211.

[Knu98] Lars Knudsen, *DEAL – A 128-bit cipher*, Technical Report, Department of Informatics, University of Bergen, Norway, 1998.

[KW02] Lars R. Knudsen and David A. Wagner, « Integral Cryptanalysis », *in*: *FSE 2002*, vol. 2365, LNCS, Springer, 2002, pp. 112–127.

[Lai94] Xuejia Lai, « Higher order derivatives and differential cryptanalysis », *in*: *Proc. "Symposium on Communication, Coding and Cryptography", in honor of J. L. Massey on the occasion of his 60'th birthday*, Kluwer Academic Publishers, 1994.

[LCJ11] Leibo Li, Jiazhe Chen, and Keting Jia, « New Impossible Differential Cryptanalysis of Reduced-Round Camellia », *in*: *CANS 2011*, vol. 7092, LNCS, Springer, 2011, pp. 26–39.

[Lea+21] Gregor Leander, Thorben Moos, Amir Moradi, and Shahram Rasoolzadeh, « The SPEEDY Family of Block Ciphers Engineering an Ultra Low-Latency Cipher from Gate Level for Secure Processor Architectures », *in*: *IACR Trans. Cryptogr. Hardw. Embed. Syst.* 2021.*4* (2021), pp. 510–545.

[Li+08] Ruilin Li, Bing Sun, Peng Zhang, and Chao Li, « New Impossible Differential Cryptanalysis of ARIA », *in*: *IACR Cryptol. ePrint Arch.* (2008), p. 227.

[Li+19] Kangquan Li, Longjiang Qu, Bing Sun, and Chao Li, « New Results About the Boomerang Uniformity of Permutation Polynomials », *in*: *IEEE Trans. Inf. Theory* 65.*11* (2019), pp. 7542–7553.

[Li+21] Kangquan Li, Chunlei Li, Tor Helleseth, and Longjiang Qu, « Cryptographically strong permutations from the butterfly structure », *in*: *Des. Codes Cryptogr.* 89.*4* (2021), pp. 737–761.

[Lim98] Chae Hoon Lim, *CRYPTON: A New 128-bit Block Cipher.* AES Proposal, 1998.

[Liu+11] Zhiqiang Liu, Dawu Gu, Ya Liu, Juanru Li, and Wei Li, « Linear Cryptanalysis of ARIA Block Cipher », *in*: *Information and Communications Security*, vol. 7043, LNCS, Springer, 2011, pp. 242–254.

[LP07] Gregor Leander and Axel Poschmann, « On the Classification of 4 Bit S-Boxes », *in*: *WAIFI 2007*, vol. 4547, LNCS, Springer, 2007, pp. 159–176.

[LS08] Shenhua Li and Chunyan Song, « Improved Impossible Differential Cryptanalysis of ARIA », *in*: *ISA 2008*, 2008, pp. 129–132.

[LS22] Ting Li and Yao Sun, « SuperBall: A New Approach for MILP 2 Modelings of Boolean Functions », *in*: *IACR Trans. Symmetric Cryptol.* 2022.*3* (2022).

[Lu+08a] Jiqiang Lu, Orr Dunkelman, Nathan Keller, and Jongsung Kim, « New Impossible Differential Attacks on AES », *in*: *INDOCRYPT 2008*, vol. 5365, LNCS, Springer, 2008, pp. 279–293.

[Lu+08b] Jiqiang Lu, Jongsung Kim, Nathan Keller, and Orr Dunkelman, « Improving the Efficiency of Impossible Differential Cryptanalysis of Reduced Camellia and MISTY1 », *in*: *CT-RSA 2008*, vol. 4964, LNCS, Springer, 2008, pp. 370–386.

[LXZ21] Nian Li, Maosheng Xiong, and Xiangyong Zeng, « On Permutation Quadrinomials and 4-Uniform BCT », *in*: *IEEE Trans. Inf. Theory* 67.*7* (2021), pp. 4845–4855.

[Mal+10] Hamid Mala, Mohammad Dakhilalian, Vincent Rijmen, and Mahmoud Modarres-Hashemi, « Improved Impossible Differential Cryptanalysis of 7-Round AES-128 », *in*: *INDOCRYPT 2010*, vol. 6498, LNCS, Springer, 2010, pp. 282–291.

[Mal14] Hamid Mala, *Private communication*, 2014.

[Mat97] Mitsuru Matsui, « New Block Encryption Algorithm MISTY », *in*: *FSE '97*, vol. 1267, LNCS, Springer, 1997, pp. 54–68.

[McC56] Edward J. McCluskey, « Minimization of Boolean functions », *in*: *The Bell SystemTechnical Journal* 35.*6* (1956), pp. 1417–1444.

[MDS11] Hamid Mala, Mohammad Dakhilalian, and Mohsen Shakiba, « Impossible Differential Attacks on 13-Round CLEFIA-128 », *in*: *J. Comput. Sci. Technol.* 26.*4* (2011), pp. 744–750.

[Men+09] Florian Mendel, Christian Rechberger, Martin Schläffer, and Søren S. Thomsen, « The Rebound Attack: Cryptanalysis of Reduced Whirlpool and Grøstl », *in*: *FSE 2009*, vol. 5665, LNCS, Springer, 2009, pp. 260–276.

[MN12] Marine Minier and María Naya-Plasencia, « A related key impossible differential attack against 22 rounds of the lightweight block cipher LBlock », *in*: *Inf. Process. Lett.* 112.*16* (2012), pp. 624–629.

[Mou+11] Nicky Mouha, Qingju Wang, Dawu Gu, and Bart Preneel, « Differential and Linear Cryptanalysis Using Mixed-Integer Linear Programming », *in*: *Inscrypt 2011*, vol. 7537, LNCS, Springer, 2011, pp. 57–76.

[MSD10] Hamid Mala, Mohsen Shakiba, and Mohammad Dakhilalian, « New impossible differential attacks on reduced-round Crypton », *in*: *Comput. Stand. Interfaces* 32.*4* (2010), pp. 222–227.

[MTX20] Sihem Mesnager, Chunming Tang, and Maosheng Xiong, « On the boomerang uniformity of quadratic permutations », *in*: *Des. Codes Cryptogr.* 88.*10* (2020), pp. 2233–2246.

[Mur11] Sean Murphy, « The Return of the Cryptographic Boomerang », *in*: *IEEE Trans. Inf. Theory* 57.*4* (2011), pp. 2517–2521.

[Nay11] María Naya-Plasencia, « How to Improve Rebound Attacks », *in*: *CRYPTO 2011*, vol. 6841, LNCS, Springer, 2011, pp. 188–205.

[NIS01] NIST, *Federal Information Processing Standards Publication (FIPS 197). Advanced Encryption Standard (AES)*, 2001.

[NK92] Kaisa Nyberg and Lars R. Knudsen, « Provable Security Against Differential Cryptanalysis », *in*: *CRYPTO '92*, vol. 740, LNCS, Springer, 1992, pp. 566–574.

[Nyb93] Kaisa Nyberg, « Differentially Uniform Mappings for Cryptography », *in*: *EUROCRYPT '93*, vol. 765, LNCS, Springer, 1993, pp. 55–64.

[Nyb94] Kaisa Nyberg, « S-boxes and Round Functions with Controllable Linearity and Differential Uniformity », *in*: *FSE '94*, vol. 1008, LNCS, Springer, 1994, pp. 111–130.

[Par+03] Sangwoo Park, Soo Hak Sung, Sangjin Lee, and Jongin Lim, « Improving the Upper Bound on the Maximum Differential and the Maximum Linear Hull Probability for SPN Structures and AES », *in*: *FSE 2003*, vol. 2887, LNCS, Springer, 2003, pp. 247–260.

[PFL16] Charles Prud'homme, Jean-Guillaume Fages, and Xavier Lorca, *Choco Solver Documentation*, TASC, INRIA Rennes, LINA CNRS UMR 6241, COSLING S.A.S., 2016.

[Qui52] Willard Van Orman Quine, « The Problem of Simplifying Truth Functions », *in*: *The American Mathematical Monthly* 59.*8* (1952), pp. 521–531.

[Qui55] Willard Van Orman Quine, « A Way to Simplify Truth Functions », *in*: *The American Mathematical Monthly* 62.*9* (1955), pp. 627–631.

[RBH17] Sondre Rønjom, Navid Ghaedi Bardeh, and Tor Helleseth, « Yoyo Tricks with AES », *in*: *ASIACRYPT 2017, Part I*, vol. 10624, LNCS, Springer, 2017, pp. 217–243.

[Shi+18] Danping Shi, Siwei Sun, Patrick Derbez, Yosuke Todo, Bing Sun, and Lei Hu, « Programming the Demirci-Selçuk Meet-in-the-Middle Attack with Constraints », *in*: *ASIACRYPT 2018, Part II*, vol. 11273, LNCS, Springer, 2018, pp. 3–34.

[Sun+14a] Siwei Sun, Lei Hu, Meiqin Wang, Peng Wang, Kexin Qiao, Xiaoshuang Ma, Danping Shi, Ling Song, and Kai Fu, *Towards Finding the Best Characteristics of Some Bit-oriented Block Ciphers and Automatic Enumeration of (Related-key) Differential and Linear Characteristics with Predefined Properties*, Cryptology ePrint Archive, Report 2014/747, 2014.

[Sun+14b] Siwei Sun, Lei Hu, Peng Wang, Kexin Qiao, Xiaoshuang Ma, and Ling Song, « Automatic Security Evaluation and (Related-key) Differential Characteristic Search: Application to SIMON, PRESENT, LBlock, DES(L) and Other Bit-Oriented Block Ciphers », *in*: *ASIACRYPT 2014, Part I*, vol. 8873, LNCS, Springer, 2014, pp. 158–178.

[Sun+16] Bing Sun, Meicheng Liu, Jian Guo, Longjiang Qu, and Vincent Rijmen, « New Insights on AES-Like SPN Ciphers », *in*: *CRYPTO 2016, Part I*, vol. 9814, LNCS, Springer, 2016, pp. 605–624.

[TAY17] Mohamed Tolba, Ahmed Abdelkhalek, and Amr M. Youssef, « Impossible Differential Cryptanalysis of Reduced-Round SKINNY », *in*: *AFRICACRYPT 2017*, vol. 10239, LNCS, 2017, pp. 117–134.

[TBP20] Shizhu Tian, Christina Boura, and Léo Perrin, « Boomerang uniformity of popular S-box constructions », *in*: *Des. Codes Cryptogr.* 88.*9* (2020), pp. 1959–1989.

[The20] The Sage Developers, *SageMath, the Sage Mathematics Software System (Version 9.1)*, `https://www.sagemath.org`, 2020.

[TM16] Yosuke Todo and Masakatu Morii, « Bit-Based Division Property and Application to Simon Family », *in*: *FSE 2016*, ed. by Thomas Peyrin, vol. 9783, LNCS, Springer, 2016, pp. 357–377.

[Tod15a] Yosuke Todo, « Integral Cryptanalysis on Full MISTY1 », *in*: *CRYPTO 2015, Part I*, vol. 9215, LNCS, Springer, 2015, pp. 413–432.

[Tod15b] Yosuke Todo, « Structural Evaluation by Generalized Integral Property », *in*: *EUROCRYPT 2015, Part I*, vol. 9056, LNCS, Springer, 2015, pp. 287–314.

[Tsu+08] Yukiyasu Tsunoo, Etsuko Tsujihara, Maki Shigeri, Tomoyasu Suzaki, and Takeshi Kawabata, « Cryptanalysis of CLEFIA using multiple impossible differentials », *in*: *ISITA 2008*, 2008, pp. 1–6.

[Tu+20] Ziran Tu, Nian Li, Xiangyong Zeng, and Junchao Zhou, « A Class of Quadrinomial Permutations With Boomerang Uniformity Four », *in*: *IEEE Trans. Inf. Theory* 66.*6* (2020), pp. 3753–3765.

[Udo21] Aleksei Udovenko, « MILP modeling of Boolean functions by minimum number of inequalities », *in*: *IACR Cryptol. ePrint Arch.* (2021), p. 1099.

[VJ04] Serge Vaudenay and Pascal Junod, *Device and method for encrypting and decrypting a block of data*, United States Patent (20040247117), see also "Fox, a New Family of Block Ciphers" http://crypto.junod.info/sac04a.pdf, 2004.

[Wag99] David A. Wagner, « The Boomerang Attack », *in*: *FSE '99*, vol. 1636, LNCS, Springer, 1999, pp. 156–170.

[WW11] Shengbao Wu and Mingsheng Wang, « Security Evaluation against Differential Cryptanalysis for Block Cipher Structures », *in*: *IACR Cryptol. ePrint Arch.* 2011 (2011), p. 551.

[WZF07a] Wen-Ling Wu, Wen-Tao Zhang, and Deng-Guo Feng, « Impossible Differential Cryptanalysis of Reduced-Round ARIA and Camellia », *in*: *J. Comput. Sci. Technol.* 22.*3* (2007), pp. 449–456.

[WZF07b] Wenling Wu, Wentao Zhang, and Dengguo Feng, « Impossible Differential Cryptanalysis of Reduced-Round ARIA and Camellia », *in*: *J. Comput. Sci. Technol.* 22.*3* (2007), pp. 449–456.

[WZZ08] Wenling Wu, Lei Zhang, and Wentao Zhang, « Improved Impossible Differential Cryptanalysis of Reduced-Round Camellia », *in*: *SAC 2008*, vol. 5381, LNCS, Springer, 2008, pp. 442–456.

[Xia+16] Zejun Xiang, Wentao Zhang, Zhenzhen Bao, and Dongdai Lin, « Applying MILP Method to Searching Integral Distinguishers Based on Division Property for 6 Lightweight Block Ciphers », *in*: *ASIACRYPT 2016, Part I*, vol. 10031, LNCS, 2016, pp. 648–678.

[YQC17] Dong Yang, Wen-Feng Qi, and Hua-Jin Chen, « Impossible differential attacks on the SKINNY family of block ciphers », *in*: *IET Inf. Secur.* 11.*6* (2017), pp. 377–385.

[ZH08] Wenying Zhang and Jing Han, « Impossible Differential Analysis of Reduced Round CLEFIA », *in*: *Inscrypt*, vol. 5487, LNCS, Springer, 2008, pp. 181–191.

[ZWF07] Wentao Zhang, Wenling Wu, and Dengguo Feng, « New Results on Impossible Differential Cryptanalysis of Reduced AES », *in*: *ICISC 2007*, vol. 4817, LNCS, Springer, 2007, pp. 239–250.

université
PARIS-SACLAY

Title : On the security of symmetric primitives and the properties of their inner components

Keywords : symmetric cryptography, block ciphers, security, cryptanalysis

Abstract : Symmetric cryptography is a central discipline for ensuring information security. Indeed, the use of cryptographic algorithms allows to protect sensitive information when stored on some physical device and permits two parties to communicate safely via an unreliable communication channel. Among these algorithms, those known as symmetric are the only ones that can guarantee a good performance in terms of speed or circuit size for most applications.

This manuscript is centred around the understanding of the security offered by symmetric primitives. This understanding can be achieved via different processes and some of these processes are described in this thesis. The first part of this document presents a careful study and analysis of some existing cryptanalysis techniques and permitted notably to generalize and improve some well-known families of attacks, such as differential and impossible differential ones. A new cryptanalysis technique that combines differential and meet-in-the-middle (MITM) attacks is then presented followed with new heuristic modeling approaches for the research of distinguishers with MILP solvers. The second part of this thesis focuses on the analysis of some newly proposed cryptanalysis techniques and proposes an alternative approach to two of them, namely the division property and the multiple-of-n distinguishers. Finally, the last chapter presents results on the resistance of symmetric algorithms against differential and boomerang attacks by studying the mathematical properties of the underlying S-boxes.