Commutative Cryptanalysis Made Practical

Jules Baudrin, Patrick Felke, Gregor Leander, Patrick Neumann, Léo Perrin, Lukas Stennes

Commutative Cryptanalysis Made Practical

Jules Baudrin[1], Patrick Felke[2], Gregor Leander[3], Patrick Neumann[3], Léo Perrin[1] and Lukas Stennes[3]

[1] Inria, Paris, France, {jules.baudrin,leo.perrin}@inria.fr
[2] University of Applied Sciences Emden/Leer, patrick.felke@hs-emden-leer.de
[3] Ruhr University Bochum, {gregor.leander,patrick.neumann,lukas.stennes}@rub.de

Abstract. About 20 years ago, Wagner showed that most of the (then) known techniques used in the cryptanalysis of block ciphers were particular cases of what he called *commutative diagram cryptanalysis*. However, to the best of our knowledge, this general framework has not yet been leveraged to find concrete attacks.
In this paper, we focus on a particular case of this framework and develop *commutative cryptanalysis*, whereby an attacker targeting a primitive E constructs affine permutations A and B such that $E \circ A = B \circ E$ with a high probability, possibly for some weak keys. We develop the tools needed for the practical use of this technique: first, we generalize differential uniformity into "A-uniformity" and differential trails into "commutative trails", and second we investigate the commutative behaviour of S-box layers, matrix multiplications, and key additions.
Equipped with these new techniques, we find probability-one distinguishers using only two chosen plaintexts for large classes of weak keys in both a modified Midori and in Scream. For the same weak keys, we deduce high probability truncated differentials that can cover an arbitrary number of rounds, but which do not correspond to any high probability differential trails. Similarly, we show the existence of a trade-off in our variant of Midori whereby the probability of the commutative trail can be decreased in order to increase the weak key density. We also show some statistical patterns in the AES super S-box that have a much higher probability than the best differentials, and which hold for a class of weak keys of density about $2^{-4.5}$.

Keywords: block cipher · commutative cryptanalysis · Midori · Scream · differential cryptanalysis · rotational cryptanalysis

1 Introduction

Symmetric cryptographic primitives protect a large fraction of our sensitive and personal data. We have at hand a strong set of algorithms that have resisted an impressive amount of cryptanalysis. However, as our understanding increases, the field of cryptanalysis becomes more and more scattered leading to a situation where it becomes harder to keep track of all possible attacks to consider when analysing, or designing, a new primitive. Thus, as more and more advanced, sophisticated, and numerous attacks are being developed, it is important to unify attacks and security arguments. By doing so, we can hope to keep the security analysis to a manageable level but also, and more importantly in the long run, to improve our fundamental understanding of modern ciphers.

However, while generalizing and unifying attacks is certainly important, it has to be done to such an extent that it still leads to meaningful results, i.e., the results have to be applicable. This in particular means that it should be possible to check if the attack vectors apply to a given cipher. Otherwise, conceptually nice ideas do not allow to be populated with non-trivial examples and remain of limited interest despite the potential large variety of properties covered. To give a concrete example, the partitioning attack

[HM97], while being a very nice and elegant framework, seems, at least for now, too general to be ever falsifiable entirely. That is, we are far from giving security arguments against partitioning attacks in its general form, and even more, concrete examples of ciphers broken by partitioning cryptanalysis are covered by very special cases of the general framework.

In our work, we show how to mount attacks exploiting that, for a given cipher E, there exist *affine* bijective mappings A and B such that there exists (a lot of) keys k for which

$$E_k(A(x)) = B(E_k(x))$$

is verified with high probability (taken over x), compared to the case where E is replaced with a random permutation. We focus on the case where E is a substitution-permutation construction. For those constructions, we feel that this property, and attacks based on it, provide the right level of generality: enabling A and B to be *any* function, as in Wagner's work on *commutative diagram cryptanalysis* [Wag04], casts too wide a net for practical use. On the other hand, as we outline next, this set-up unifies several important attack vectors, allows to construct new attacks and it is possible, in the case of a very-high probability, to be detected algorithmically in its general form. The latter point is particularly important for security arguments against those attack vectors.

Generality. As pointed out in [Wag04], the framework of what we will refer to as *commutative cryptanalysis* captures important classes of attacks as special cases. Most prominently, differential attacks correspond to the case where both A and B are simple translations. As extensions to this, it also includes rotational cryptanalysis, as introduced in [KN10] and generalized to capture differences on top of rotations in [AL16]. In the former A and B are rotations, while in the latter they are allowed to be rotations composed with translations.

Focusing on a round-based structure, i.e. when E is the composition of several simpler permutations R_i, allows to discuss *commutative trails* where we have a chain of affine mappings such that each of the round functions R_i fulfills

$$R_i(A_i(x)) = A_{i+1}(R_i(x))$$

with high probability. An interesting special case of this is when translations for A_i are interleaved with non-translations. This in particularly captures and puts into a better perspective a recent example at CRYPTO 2023 [BFL+23] of a toy cipher, where a probability-one differential over two rounds has been constructed such that it does not stem from a probability-one characteristic.

Applicability. On the other hand, in the case of an iterated cipher, and more particularly in the case of a SPN cipher, tracing a commutative trail through its parts becomes feasible, at least in the case of a high probability. As we will detail, each of the usual components, the S-box and linear layers, and the key or constant addition, allows for a rather rich theory and comes with its own insights to the full picture of commutative cryptanalysis.

For an S-box, the probability-one case of the equation $S(A(x)) = B(S(x))$ can be algorithmically solved for all instances of interest by known algorithms, in particular [BDBP03] and [Din18]. In practice, several S-boxes from the literature have such a property, in particular Midori [BBI+15] (both the 64- and the 128-bit versions), and Scream [GLS+15].

For the linear layer, the behaviour of commutation can be captured with the rich structure of linear and affine mappings. Here, in particular, the case of an S-box-aligned block diagonal matrix that commutes with the linear layer is studied. Interestingly, in some cases, commutation with the linear layer is the only probabilistic step, all others happening with probability one.

For the round keys, or constant additions, it is not hard to see that the probability-one case corresponds to keys or constants being fixed points of the linear part of A (or B).

This also implies that a commutative trail that commutes with the addition of *any* key must be such that all keys are fixed points of the linear part, i.e. the traditional case of differential cryptanalysis. This simple observation has thus significant consequences for excluding the existence of probability-one trails. At the same time, allowing lower probabilities significantly increases the reach of commutative cryptanalysis, allowing it to be applied even to the AES super S-box.

Using all those insights, that we develop in detail in Section 4, in particular allows to algorithmically check for any probability-one commutative trails in reasonable time for a large number of ciphers. We do so in Section 5. We can also explain in Section 6 some astonishing properties of variants of Midori, in particular the existence of very high-probability truncated differentials ($p = 2^{-16}$) that, for a large number of weak keys, can cover an arbitrary number of rounds, but do not correspond to any high-probability differential trail. Moreover, the "real" Scream exhibits a similar property. The existence of such differentials highlights the limits of the current security arguments.

Related Work. The key insight behind our work is that several attacks can be seen as particular cases of the commutation with affine mappings. This was first put forward by Wagner in [Wag04] when he introduced *commutative diagram cryptanalysis* (which we shortened into *commutative cryptanalysis*). His approach is even more general as he does not require the mappings A and B to be permutations, nor do they need to be affine. His framework then also captures for instance linear attacks and non-linear invariant subspace attacks. However, to the best of our knowledge, this framework has never been used directly to find a distinguisher.

In [BM22], Bellini and Makarim re-used the framework laid out by Wagner and renamed it *functional cryptanalysis*. Their main result is an attack against round-reduced Xoodoo [DHP+18] that is of a different nature than the ones we present in this paper. Indeed, they consider an object akin to a commutative trail where the first rounds correspond to a simple differential trail, but where the last transition is of the shape $\sigma \circ F = F \circ T_\alpha$, where T_α is the translation by a well chosen difference α, and where σ is a non-linear function. Such transitions are always possible (we can simply set $\sigma = F \circ T_\alpha \circ F^{-1}$), and it seems at this stage hard to chain non-linear commutative patterns as it is a priori hard to describe their commutation with each operation. Still, it remains an interesting instantiation of the framework of Wagner that deserves further investigation.

Outline. We present the notations in Section 2, and then describe the known attack techniques covered by the commutative approach in Section 3. In Section 4, we investigate how to mount a commutative attack in practice, and in particular how to find mappings that commute with the various building blocks of a Substitution-Permutation Network (SPN). Using those insights, in Section 5, we present an algorithm that can exhaustively search for probability-one commutative trails for SPN ciphers. We can thus prove the absence of probability-one commutative trails for several ciphers and to automatically discover ones for Midori and Scream round functions. Naturally, we then explore those cases further. We present new attacks against variants of Midori (with a slightly-modified key schedule), and against the *genuine* Scream in Section 6. In particular, we provide extensive experimental results on the behaviour of probabilistic commutative trails. Section 7 concludes the paper.

2 Preliminaries

2.1 Notations

Let κ, d, s be positive integers. Let n, m, ℓ be positive integers such that $n = m \times \ell$. Cardinality of a set S is denoted $|S|$.

Finite fields and vector spaces. Let $\mathbb{F}_2$ denote the finite field with two elements, $\mathbb{F}_2^d$ the vector space of dimension d over $\mathbb{F}_2$ and $\mathbb{F}$ a generic finite vector space over $\mathbb{F}_2$. Given two bits (elements of $\mathbb{F}_2$) or binary vectors (elements of $\mathbb{F}_2^n$), we denote $+$ the bit-wise addition (XOR). We use the usual vector space isomorphism $\mathbb{F}_2^n \simeq (\mathbb{F}_2^m)^\ell$ and refer to the former as the *state* point of view, and to the latter as the *cell* point of view. When it is necessary to distinguish between both, we reserve caligraphic letters to applications applied on the whole state: $\mathcal{F}\colon \mathbb{F}_2^n \to \mathbb{F}_2^n$, and capital letters to functions applied on a cell: $F\colon \mathbb{F}_2^m \to \mathbb{F}_2^m$. For $a \in \mathbb{F}_2^m$, $\widehat{a}$ denotes the vector whose ℓ cells are all equal to a: $\widehat{a} := (a, \cdots, a)$. $a^{\times s} := (a, \cdots, a) \in \mathbb{F}_2^s$ might be used to emphasize the sizing.

We denote the subspace spanned by a tuple of vectors $(v_1, \ldots, v_s) \in \mathbb{F}^s$ by $\langle v_1, \ldots, v_s \rangle := \{a_1 v_1 + \cdots + a_s v_s, (a_1, \ldots, a_s) \in \mathbb{F}_2^s\}$. We denote generic sub-spaces with V.

If not explicitly stated otherwise, when linearity (resp. affinity) is mentioned, we always refer to $\mathbb{F}_2$-linearity (resp. $\mathbb{F}_2$-affinity). Given an affine mapping $A\colon \mathbb{F} \to \mathbb{F}$, we denote $c_A := A(0)$ its constant term and $L_A := A + c_A$ the linear part of it.

We represent binary vectors often as integers, either written in decimal, hexadecimal or binary notation. The explicit relationship between integers and vectors we use throughout the paper is as follows:

$$\sum_{i=0}^{n-1} a_i 2^i \simeq (a_{n-1}, \ldots, a_0) \simeq \begin{pmatrix} a_{n-1} \\ \vdots \\ a_0 \end{pmatrix}.$$

Matrix spaces. We denote by $\mathbf{M}(n, \mathbb{F})$ (respectively $GL(n, \mathbb{F})$) the space of square (resp. invertible square) matrices of size n with coordinates in $\mathbb{F}$. I denotes the identity matrix. When dealing with block matrices, we always consider a matrix of size $n \times n$, where all the sub-matrices are square matrices of size $m \times m$. Given ℓ matrices of $\mathbf{M}(m, \mathbb{F}_2)$, namely, $L_1, \cdots, L_\ell$, we denote $\mathrm{Diag}(L_1, \cdots, L_\ell)$ (resp. $\mathrm{Diag}(L_1)$) the diagonal block matrix whose i-th diagonal block is L_i (resp. is L_1). While defining a matrix (or vector), the value 0 can be replaced by a dot to make the reading easier.

Parallel applications. Abusing the previous notation, in case of generic mappings $(G, G_1, \cdots, G_\ell)$, the parallel application of G_i on the i-th cell will be denoted $\mathrm{Diag}(G_1, \cdots, G_\ell)$, and $\mathrm{Diag}(G)$ will refer to the parallel application of G on each of the ℓ cells. Finally, *if* mappings F, G, A, S *are defined* from $\mathbb{F}_2^m$ to itself, $\mathcal{F}, \mathcal{G}, \mathcal{A}, \mathcal{S}$ always refer to $\mathrm{Diag}(F), \mathrm{Diag}(G), \mathrm{Diag}(A), \mathrm{Diag}(S)$.

Commutation and conjugation. Let $F_1, F_2, F_3\colon \mathbb{F} \to \mathbb{F}$ be functions and $G\colon \mathbb{F} \to \mathbb{F}$ be a permutation. By the *conjugate of* F_1 *by* G, we mean $G \circ F_1 \circ G^{-1}$ and we denote it F_1^G. By F_1 *commutes* with F_2 we mean that F_1 and F_2 verify the following relation $F_1 \circ F_2 = F_2 \circ F_1$. The *center* of F_1 is defined as the set of all maps commuting with F_1 and is denoted $Z(F_1) := \{F\colon \mathbb{F} \to \mathbb{F}, F \circ F_1 = F_1 \circ F\}$ (Zentrum) abusing standard labeling, by A and B *commute through* F we mean that A, B, F verify $F \circ A = B \circ F$. Such a behavior will be denoted using arrows, as in standard linear or differential trail studies: either as $A \xrightarrow{F} B$ or $A \to B$ if the context is clear. Finally, we denote $\mathrm{Fix}(F)$ the set of fixed points of F: $\mathrm{Fix}(F) := \{x \mid x = F(x)\}$.

2.2 Toy Cipher Families: Vert and Grün

Later in this paper, we will use variants of the block cipher Midori [BBI+15] to illustrate our approach. Like Midori, they are named after the word "green" in different languages.

Block cipher. Let us first introduce general notations common to all block ciphers we investigate. Let $\mathcal{E}_k$ be an n-bit key-alternating block cipher parameterized by a κ-bit key k, a bijective non-linear S-box $S : \mathbb{F}_2^m \to \mathbb{F}_2^m$ and a linear layer $\mathcal{L} : \mathbb{F}_2^n \to \mathbb{F}_2^n$. Let $c \in \mathbb{F}$. We denote the translation by c as $T_c \colon \mathbb{F} \to \mathbb{F}, x \mapsto x + c$. If $\mathbb{F} = \mathbb{F}_2^n$, we use $\mathcal{T}_c := T_c$ instead to emphasize that it is applied on the full state. We denote sk^i the n-bit string (derived from k) that is added to the state at the start of round i, so that the r rounds of $\mathcal{E}_k$ can be written

$$\mathcal{E}_k = \mathcal{T}_{\mathsf{sk}^{r+1}} \circ \mathcal{L} \circ \mathcal{S} \circ \mathcal{T}_{\mathsf{sk}^r} \circ \ldots \circ \mathcal{L} \circ \mathcal{S} \circ \mathcal{T}_{\mathsf{sk}^1} \ .$$

Round functions are denoted using $\mathcal{R}_i := \mathcal{L} \circ \mathcal{S} \circ \mathcal{T}_{\mathsf{sk}^i}$ and $\bar{\mathcal{R}}_i := \mathcal{T}_{\mathsf{sk}^{i+1}} \circ \mathcal{L} \circ \mathcal{S}$.

Midori64. The AES-like block cipher Midori [BBI+15] attracted many third-party analysis [TLS19, Bey18, Bey21, LW17, BCC19]. We describe here the 64-bit-state version, namely Midori64. The very-similar 128-bit-state version is presented in Supplementary Material B.1.

First, the S-box layer of Midori64 uses a single involutive 4-bit S-box which is applied on all nibbles. Then, the MixColumn operation uses a binary quasi-MDS matrix M that is applied on each column: the i-th output cell is the XOR of the three input cells of index different from i. We denote $\mathcal{M}$ the parallel application of M on the 4 columns of the state.

$$M = \begin{bmatrix} . & 1 & 1 & 1 \\ 1 & . & 1 & 1 \\ 1 & 1 & . & 1 \\ 1 & 1 & 1 & . \end{bmatrix}$$

The ShuffleCell operation consists in a reorganization (permutation) of the cells. Finally a round key is XORed at the end of each round. It is derived from the 128-bit master key $K = K_0||K_1$: K_0 is used for even rounds, and K_1 for odd rounds. Sparse round constants that belong to $\{\texttt{0x0}, \texttt{0x1}\}^{16}$ are also added at each round.

From now on, *if not explicitly stated otherwise*, Midori will always refer to Midori64.

Vert. Vert is a family of 64-bit-state and 128-bit-key ciphers which are heavily based on Midori64: all of its subroutines are almost identical to the ones used in the latter.

Vert uses the same S-box and MixColumn layers as Midori64. The permutation of cells can be chosen to be either the genuine Midori ShuffleCell or the AES ShiftRows. Finally, the key-schedule is identical, except that the round constants added in each nibble can take any value in $\{0, c\}$ throughout the encryption (Midori uses $c = 1$). The cells permutation is denoted using subscripts, and the choice of c using superscripts: $\mathsf{Vert}^c_{\mathrm{SR}}$ and $\mathsf{Vert}^c_{\mathrm{SC}}$.

Midori thus corresponds to $\mathsf{Vert}^1_{\mathrm{SC}}$. Such modified versions of Midori64 have already been studied, namely $\mathsf{Vert}^1_{\mathrm{SR}}$ in the original Midori paper [BBI+15] (where bounds on differential active S-boxes are given), or in previous cryptanalysis papers, $\mathsf{Vert}^c_{\mathrm{SC}},\ c \in \langle 2, 8\rangle$ in [Bey18, Bey20] and $\mathsf{Vert}^5_{\mathrm{SC}}$ in [TLS19].

Grün. Grün is a modified-constants version of Midori128. It is identical to Midori128 except for the constants that lie in $\{\texttt{0x00}, \texttt{0x11}\}$ rather than in the genuine $\{\texttt{0x00}, \texttt{0x01}\}$.

3 Commutative Cryptanalysis as a Unifying Framework

While the concept of a commutative distinguisher may seem abstract, we illustrate in this section a claim of Wagner, namely that it is in fact a convenient tool which captures a wide range of attacks, from differential cryptanalysis to rotational attacks. To this end, we

first setup some concepts that will be used throughout the paper, and then argue that multiple attacks fit them.

As a side-note, in Appendix A we explain that the framework can also be used to understand what has previously been discussed as conjugate ciphers, whereby a round function R is not studied directly, but instead a conjugate version $G \circ R \circ G^{-1}$ is, for an auxiliary permutation G. Though G itself may be non-linear, the differential behaviour of $G \circ R \circ G^{-1}$ is best explained by commutative trails.

3.1 Basic Tools and Definitions

Definition 1 (A-Uniformity)**.** Given an S-box $S : \mathbb{F}_2^m \to \mathbb{F}_2^m$ and two bijective affine mappings $A, B : \mathbb{F}_2^m \to \mathbb{F}_2^m$ we define

$$\Gamma_S(A, B) := |\{x \mid S \circ A(x) = B \circ S(x)\}| \quad \text{and} \quad \Gamma_S := \max_{\substack{A,B \\ I \notin \{A,B\}}} \Gamma_S(A, B).$$

The maximum Γ_S is referred as the *a(ffine)-uniformity* of S.

This naturally generalizes the well-known notion of differential uniformity as introduced by Nyberg in [Nyb94]. Unfortunately, at this stage, there is no efficient algorithm to compute this quantity. Brute-forcing A and B is doable for m up to 4, but not beyond this small value. In fact, we consider that an algorithm able to solve this problem would be a very interesting scientific contribution.

Lemma 1. *If $S : \mathbb{F}_2^m \to \mathbb{F}_2^m$ is a permutation, and A and B are affine permutations with $\Gamma_S(A, B) = 2^m$, i.e. S is self-equivalent, then:*

1. *A and B are conjugates (and thus have the same cycle type and the same order),*
2. *$\Gamma_S(A, B) = \Gamma_S(A^i, B^i)$ for any integer $i > 0$,*
3. *$\Gamma_{C \circ S \circ D}(A, B) = \Gamma_S(D \circ A \circ D^{-1}, C^{-1} \circ B \circ C)$, provided that C and D are affine permutation, i.e. A-uniformity is invariant under affine equivalence, and*
4. *$S(\mathrm{Fix}_A) = \mathrm{Fix}_B$.*

Proof. First, by multiplying on the left both sides of $S \circ A = B \circ S$ by S^{-1} (S is bijective), we observe that $A = S^{-1} \circ B \circ S$, *i.e.* $A = B^{S^{-1}}$.

The second point is obtained using a straightforward induction: if $S \circ A^i = B^i \circ S$, then we obtain that $S \circ A^{i+1} = B^{i+1} \circ S$ using that $S \circ A = B \circ S$.

The third point is proved by simply writing down the definition of $\Gamma_{C \circ S \circ D}(A, B)$ and simplifying the resulting equations.

Finally, for the fourth point, let $x \in \mathrm{Fix}_A$ be a fixed point of A. Then it holds that $S(x) = S \circ A(x) = B \circ S(x)$. In other words, $S(x)$ is a fixed point of B and $S(\mathrm{Fix}_A) \subseteq \mathrm{Fix}_B$. Equality follows from switching the roles of A and B and replacing S by S^{-1}. □

It is possible to leverage such commutative patterns existing at the level of a subfunction to develop what we call *commutative cryptanalysis*. It can be seen as a generalization of differential cryptanalysis: for "regular" differential cryptanalysis, we study patterns of the type $S \circ A(x) = B \circ S(x)$, where $A(x) = x + \alpha$ and $B(x) = x + \beta$. Furthermore, much like differential cryptanalysis, this attack can be investigated first at the round level and then adapted to multiple rounds using a *commutative trail*. This principle is summarized in the diagram Figure 1. As we show below, this framework captures several types of attacks.

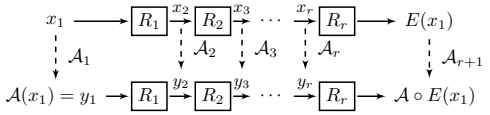

Figure 1: Overview of a commutative trail built layer by layer.

3.2 Differential Attack (and Some Variants)

As we explained above, the classical differential attack corresponds to the case where $A_i := T_{\alpha_i} : x \mapsto x + \alpha_i$, for any i. In that case, the commutative trail exactly corresponds to a classical differential trail.

***c*-differential.** The concept of *c-differential* was introduced in [EFR+20] to generalize "regular" differentials.

Definition 2 (*c*-derivative [EFR+20])**.** Let p be a prime number and n, m positive integers. Let $\mathbb{F}_{p^n}$ (resp. $\mathbb{F}_{p^m}$) be the field with p^n (resp. p^m) elements. Given a p-ary (n, m)-function $F : \mathbb{F}_{p^n} \to \mathbb{F}_{p^m}$, and $c \in \mathbb{F}_{p^m}$, the (multiplicative) c-derivative of F with respect to $a \in \mathbb{F}_{p^n}$ is the function ${}_cD_aF$ defined as ${}_cD_aF(x) = F(x + a) - cF(x)$, for all $x \in \mathbb{F}_{p^n}$.
For a fixed $c \in \mathbb{F}_{p^m}$ let k be the maximal number of solutions of ${}_cD_aF(x) = b$, where the maximum is taken over $b \in \mathbb{F}_{p^m}, a \in \mathbb{F}_{p^n}$ ($a \in \mathbb{F}_{p^n}^*$, if $c = 1$). Then k is called the *c-differential uniformity* of F.

In the binary case ($p = 2$), the definition of ${}_cD_aF$ can thus be reformulated as ${}_cD_aF = F \circ T_a + M_c \circ F$; where $M_c := x \mapsto cx$. Stated otherwise, the c-derivative with respect to a estimates how much T_a and M_c commute through F. While c-differential uniformity has been extensively studied on its own, e.g. in [EFR+20, ERST21, HPS22, MRS+21, SGG+22], we are not aware of any cryptanalysis leveraging it at this stage. It seems to be hard to find an exploitable invariant. Indeed, as we will see later, as long as $c \neq 1$ any non-zero constant (or key) addition destroys a commutative trail involving M_c. At the same time, c-differential uniformity is a lower bound on A-uniformity.

Rotational(-XOR). Let ρ be the rotation of a word by one bit to the left. In a rotational distinguisher [KN10], an attacker tries to find pairs of rotations ρ^i and ρ^j such that $\rho^i \circ F = F \circ \rho^j$. This is a simple example of commutative pattern where the affine permutations correspond to the rotations. Furthermore, this patterns are built iteratively (round by round) in a way which corresponds exactly to a commutative trail.

This attack was more recently generalized into *Rotational-XOR (RX)* cryptanalysis [AL16] where the rotations can be composed with constant additions. The goal is to track pairs of inputs of the form $(x + a, \rho^i(x) + b)$, and see if they retain a relationship of this shape after each subfunction R. Concretely, the aim is to find (a, a', b, b') and (i, i') such that $R(x + a) + a' = R\left(\rho^i(x) + b\right) + b'$ with a high probability. This equivalently means $R(y) + a' + b' = R\left(\rho^i(y + a) + b\right)$, where $y = x + a$, which is a particular case of $A \circ R = R \circ B$, where $A(x) = x + a' + b'$, and $B(x) = \rho^i(x + a) + b$.

3.3 Self-Similarity, Linear Commutants & Invariant Subspaces

In a paper of Leander, Minaud & Rønjom [LMR15], and in more depth in Minaud's thesis [Min16], the case of linear maps commuting with the round function is addressed. It corresponds to the case where A_i is linear rather than affine. As explained in [LMR15], the existence of such linear commutants can be restated as a particular case of self-similarity.

Definition 3 (Self-similarity in a block-cipher [BB02, BDLF10])**.** For a fixed block cipher E, a self-similarity relation is given by invertible and efficently computable mappings ϕ, ψ, θ such that: $\forall\ K, x, \quad \theta \circ E_K(x) = E_{\psi(K)} \circ \phi(x)$.

In Section 6, we will present several such relations. Moreover, as indicated in [LMR15], linear commutants always imply (possibly trivial) invariant subspaces. A similar implication holds when considering affine commutants, as the fixed points of an affine mapping form an affine subspace in case they exist.

Finally, the linear maps investigated in these previous works lie under the so-called "S-box independent setting": they act as a permutation of the cells.[1] Stated otherwise, they can be viewed as block matrices in which a single block of each row and each column is the identity matrix, while all the others are the zero matrix. This choice avoids taking into account the S-box and rather focus on the linear layer. The approach of Section 4 is complementary: the affine commutants are built using a small affine mapping A which is "S-box dependent". On the contrary, it almost avoids taking the linear layer into account.

Cryptanalysis of NORX v2.0 [CFG+17]. NORX v2.0 is an ARX permutation-based AEAD cipher which was a third-round candidate of the CAESAR competition [CAE14]. In a paper from Chaigneau, Fuhr, Gilbert, Jean, & Reinhard [CFG+17], a ciphertext-only forgery attack on full NORX v2.0 is described. The cornerstone of this attack is the observation that the permutation P, that is applied to a square state, commutes with column rotations. Denoting ρ the rotation of a square state by one column to the left: we obtain $\rho \circ P = P \circ \rho$. This is proved by following a trail through two half-rounds as ρ commutes with the subcomponents $G_{\text{col}}, G_{\text{diag}}$ of P. Finally, using the second point of Lemma 1, P commutes with the left column-rotation by *any* number of columns.

3.4 Probability-One Differential Over Two Rounds

In their recent work, Beierle *et al.* [BFL+23] present a cipher that, for some weak keys, exhibits a probability-one differential over two rounds. This corresponds to a differential version of the backdoored cipher Boomslang which was proposed in [BBFL22]. The S-box layer $\mathcal{S}$ consists of the threefold parallel application of an $m = 5$-bit S-box S. For the exact definitions of the S-box S and the linear layer $\mathcal{L}$, we refer to [BFL+23]. Here, we just state that the probability-one differential, which does *not* consists of a *single* differential trail, can be understood as a *single* probability-one *commutative trail*.

More precisely, there is an affine map A, and a difference δ (both given in [BFL+23]), for which it holds that translating the input of the S-box S by δ is the same as applying the affine map A to its output, and vice versa. That is, $S \circ T_\delta = A \circ S$ and $T_\delta \circ S = S \circ A$. Of course, the same holds if we consider the full S-box layer $\mathcal{S}$. Translating the inputs of all three S-boxes by $\Delta = \delta^{\times 3}$ is the same as applying $\mathcal{A} = A^{\times 3}$ to the outputs, and again vice versa. Furthermore, $\mathcal{A}$ commutes with the linear layer, i.e., $\mathcal{A} \circ \mathcal{L} = \mathcal{L} \circ \mathcal{A}$. Now, for an arbitrary key k and a weak key k' (we will discuss the properties of weak keys in Section 4.1), combining the aforementioned properties gives an iterative probability-one *commutative trail* over the two rounds $\mathcal{L} \circ \mathcal{S} \circ \mathcal{T}_{k'} \circ \mathcal{L} \circ \mathcal{S} \circ \mathcal{T}_k$:

$$\mathcal{T}_\Delta \xrightarrow{\mathcal{T}_k} \mathcal{T}_\Delta \xrightarrow{\mathcal{S}} \mathcal{A} \xrightarrow{\mathcal{L}} \mathcal{A} \xrightarrow{\mathcal{T}_{k'}} \mathcal{A} \xrightarrow{\mathcal{S}} \mathcal{T}_\Delta \xrightarrow{\mathcal{L}} \mathcal{T}_\Delta.$$

4 Commuting with Basic Building Blocks

The easiest way to find commutants for any iterative construction is to find compatible ones for each building block, and then chain those to form *trails*. Thus, we investigate

[1] The definition is actually more general than this one, to cope with partial S-box layers where the S-box is only applied to *some* of the cells. Yet, we will not consider this case in this study.

each of the layers of a traditional SPN block cipher (as specified in Section 2.2) separately.

4.1 Commuting with the Key Addition

The subkey addition has a non-trivial interaction with commutation. We indeed need to distinguish the probability of a commutation for a fixed key, and the probability that a given key enables a commutation. To better discuss these probabilities, we introduce the following concept.

Definition 4. Consider a commutation $A \xrightarrow{T_k} B$. We say that k is *p-weak* for it if $B(x+k) = A(x)+k$ holds with probability p, i.e. if $|\{x \in \mathbb{F}_2^n | B(x+k) = A(x)+k\}| = p2^n$. If $p = 0$, we simply say that k is *strong*.

The idea is that a key is weak if it enables the commutation, and is weaker if it allows it with a higher probability. For example, in the differential case where $A = B = T_a$, we have that all keys are 1-weak (i.e. the worst possible situation from a security standpoint). On the other hand, if $A = T_a$ and $B = T_b$ with $a \neq b$, then all keys are strong. The following proposition allows us to predict these behaviours in the general case.

Proposition 1. *Let $T_k, A, B \colon \mathbb{F}_2^n \to \mathbb{F}_2^n$ be the translation by k and two affine permutations. Then, k is either strong or $2^{-\operatorname{rank}(L_A+L_B)}$-weak, and the number of $2^{-\operatorname{rank}(L_A+L_B)}$-weak keys is given by*

$$|\operatorname{Im}(I + L_B) \cap \operatorname{Im}(A + B)| \times |\ker(I + L_B)| \ .$$

Proof. The core equation is $B \circ T_k(x) = T_k \circ A(x)$. It can equivalently be written $L_B(x + k) + c_B = L_A(x) + c_A + k$, and thus as

$$(A + B)(x) = (I + L_B)(k) \ . \tag{1}$$

As a consequence, in order for the equation to have solutions (x, k), it is necessary and sufficient that $\operatorname{Im}(A + B)$ and $\operatorname{Im}(I + L_B)$ have a non-empty intersection.

For a fixed k, the number of solutions x of the equation is then either 0 if $(I+L_B)(k) \notin \operatorname{Im}(A + B)$, or given by the size of the kernel of $L_A + L_B$, namely $2^{n-\operatorname{rank}(L_A+L_B)}$. This indeed yields a probability of $2^{-\operatorname{rank}(L_A+L_B)}$.

Regarding weak keys, they can be enumerated first by fixing a value $v \in \operatorname{Im}(A + B) \cap \operatorname{Im}(I + L_B)$, and then finding the $|\ker(I + L_B)|$ keys that verifies $(I + L_B)(k) = v$. So there are $|\operatorname{Im}(I + L_A) \cap \operatorname{Im}(A + B)| \times |\ker(I + L_B)|$ of them. □

If $L_A = L_B$, this can be greatly simplified as Equation 1 becomes $(I+L_B)(k) = c_A+c_B$.

Corollary 1. *If $L_A = L_B$, then a key k is 1-weak if $(I + L_B)(k) = c_A + c_B$, and strong otherwise. The first case occurs for weak keys living in a space of dimension $n - \operatorname{rank}(I + L_B)$, and is possible if and only if $c_A + c_B \in \operatorname{Im}(I + L_B)$. This is trivially true if $c_A = c_B$.*

Again, this corollary is coherent with the usual differential attack: in that case $L_A = L_B = I$, and the transition has probability 1 if $c_A = c_B$ (0 otherwise).

4.2 Commuting with S-box Layers

Finding (all) affine permutations A and B such that $S \circ A = B \circ S$ amounts to the well-known problem of affine equivalence. Hence, we can use the algorithm of Dinur [Din18] if the degree of S is maximum (i.e., $m-1$), or otherwise the one of Biryukov *et al.* [BDBP03]. While less time efficient, the latter works for any permutation, regardless of its degree. We

tweaked this algorithm to exhaustively list all pairs (A, B^{-1}) of affine permutations such that $B^{-1} \circ S \circ A = S$.[2]

While a random permutation (of sufficient size) is not expected to be (non-trivially) affine equivalent to itself [Hou06], the S-boxes used in practice are usually highly structured, either because they correspond to a simple Boolean circuit for an efficient implementation, or because they have a strong mathematical structure, e.g., because they are affine-equivalent to a finite field monomial. Related to that, all known APN permutations admit a non-trivial self-equivalence and it has actually been conjectured to be true for all APN permutations in [BBL21]. For a discussion on this, we also refer to [BDBP03]. For 4-bit S-boxes, we checked all equivalence classes and found that 137 out of all 302 classes are (non-trivially) affine self-equivalent. The case of $\Gamma_S(A, B) < 2^m$, i.e., the case of (A, B) not implying maximal A-uniformity, is covered by [BDBP03, Section 4.3].

Now consider the whole S-box layer $\mathcal{S}$. If $\mathcal{S}$ consists of S-boxes S with non-trivial linearity or differential uniformity then [RP20, Theorem 1] implies that there exist affine permutations $\mathcal{A}, \mathcal{B}$ such that $\mathcal{S} \circ \mathcal{A} = \mathcal{B} \circ \mathcal{S}$ if and only if there exist families of affine permutations $\{A_i\}_i$, $\{B_i\}_i$ such that $S \circ A_i = B_i \circ S$ and $\mathcal{A}$ and $\mathcal{B}$ are (up to a permutation of the S-boxes) the same as $\mathrm{Diag}(A_1, \ldots, A_\ell)$ and $\mathrm{Diag}(B_1, \ldots, B_\ell)$ respectively. In other words, finding $\mathcal{A}$ and $\mathcal{B}$ can be reduced to finding A and B such that $S \circ A = B \circ S$ and combining them accordingly. Notice that this also includes linear mappings that merely permute the full inputs of the S-boxes, such as the ShiftRows operation of the AES. Much like in a differential cryptanalysis, such trails imply a notion of *active* S-boxes. We call *active* an S-box that is expected to commute with an affine mapping that is *not* the identity. When restricting commutative cryptanalysis to the differential case, the definitions match. Perhaps counter-intuitively however, an active S-box in a commutative trail does not necessarily decrease its probability.

4.3 Commuting with Linear Layers

First, we recall that for two affine permutations $A, B \colon \mathbb{F}_2^n \to \mathbb{F}_2^n$ to commute with the linear layer $\mathcal{L}$ with probability $2^{-n}\Gamma_{\mathcal{L}}(A, B)$ it has to hold that

$$\mathcal{L} \circ A(x) = B \circ \mathcal{L}(x) \quad \iff \quad (\mathcal{L} \circ L_A + L_B \circ \mathcal{L})(x) = \mathcal{L}(c_A) + c_B \tag{2}$$

for $\Gamma_{\mathcal{L}}(A, B)$ values of x. In other words, $\Gamma_{\mathcal{L}}(A, B)$ is the number of solutions of the right hand side of Eq. (2), which is either zero if $\mathcal{L}(c_A) + c_B \notin \mathrm{Im}\,(\mathcal{L} \circ L_A + L_B \circ \mathcal{L})$, or the size of the kernel of $\mathcal{L} \circ L_A + L_B \circ \mathcal{L}$.

In case A and B are $\mathrm{Diag}((A_i)_i)$ and $\mathrm{Diag}((B_i)_i)$ with $S \circ A_i = B_i \circ S$ respectively, meaning that L_A and L_B are block diagonal matrices, and if we denote by $\mathcal{L}_{i,j}$ the blocks of size $m \times m$ of $\mathcal{L}$, then $(x_1, \ldots, x_\ell) \in \ker(\mathcal{L} \circ L_A + L_B \circ \mathcal{L})$ implies $L_{i,i} \circ L_{A_i}(x_i) = L_{B_i} \circ L_{i,i}(x_i)$ for any i. Hence, the kernel of $\mathcal{L} \circ L_A + L_B \circ \mathcal{L}$ is a subset of the Cartesian product (over i) of $\ker(L_{i,i} \circ L_{A_i} + L_{B_i} \circ L_{i,i})$.

If we now require the commutation to happen with probability one[3] (i.e. for all x), then the right hand side of Eq. (2) directly implies $\mathcal{L}(c_A) = c_B$ (by using $x = 0$) and $\mathcal{L} \circ L_A = L_B \circ \mathcal{L}$. If additionally $A = B$, then L_A lies in the center of $\mathcal{L}$, denoted by $Z(\mathcal{L})$. Interestingly, the center of any linear map is an algebra and has therefore a lot of structure. Even more can be said if A is the parallel application of the same cell-size mapping A', denoted by $\mathcal{A}$. Firstly, any cells permutation layer commutes with any such (affine) permutation applied in parallel on all cells. But we can also fully classify all $\mathcal{A}$ with $\mathcal{L} \circ \mathcal{A} = \mathcal{A} \circ \mathcal{L}$ for arbitrary $\mathcal{L}$.

[2] We will make our implementation public after acceptance.

[3] The probabilistic commutation is harder to generically treat, but an example is given in Section 6.2.

Theorem 1. *Let $\mathcal{L} = (\mathcal{L}_{ij})$ be a linear permutation expressed as a $\ell \times \ell$ block matrix whose blocks are of size $m \times m$. Let $\mathcal{A} = \mathrm{Diag}(A)$ for an affine permutation A. Then $\mathcal{L} \circ \mathcal{A} = \mathcal{A} \circ \mathcal{L}$ if and only if $\mathcal{L}_{ij} \in Z(L_A)$ for all $i, j \in \{1, \cdots, \ell\}$, and $c_{\mathcal{A}} \in \mathrm{Fix}(\mathcal{L})$.*

Proof. From the discussion above we already know that $\mathcal{L} \circ \mathcal{A} = \mathcal{A} \circ \mathcal{L}$ if and only if $L_{\mathcal{A}} \in Z(\mathcal{L})$ and $c_{\mathcal{A}} \in \mathrm{Fix}(\mathcal{L})$. The proof now follows from noting that $L_{\mathcal{A}} \in Z(\mathcal{L})$ if and only if $\mathcal{L}_{ij} \in Z(L_A)$ for all $i, j \in \{1, \cdots, \ell\}$, as $L_{\mathcal{A}}$ is a block diagonal matrix whose blocks are aligned with the those of $\mathcal{L}$. □

As we can see, Condition 1 of Theorem 1 can be reformulated as: $L_A \in \bigcap_{i,j} Z(\mathcal{L}_{ij})$. This reformulation is very convenient when investigating a fixed linear layer: in that case, the main objective is the description of the intersection of the centers of all sub-blocks. In particular, the case of matrices whose sub-blocks are either the null matrix or the identity matrix is very simple to handle. Indeed, any matrix commutes with both of them. In that case, Condition 1 *does not* constrain the choice of the linear part of A. As simple as this example might seem, it is actually very enlightening, as a lot of linear layers from the literature are built in this way. This is the case of binary MixColumn layers (as in Midori and many other ciphers), but also the case of the linear layers of LS designs [GLSV15]. One should thus be careful when defining a cipher using a self-affine-equivalent S-box together with a binary linear layer, as the cases of Vert and Scream in Section 6 will highlight.

5 Searching for Probability-One Commutative Trails

By the discussion in Section 4.2, if the S-box S has non-trivial linearity or differential uniformity, then all $\mathcal{A}$ and $\mathcal{B}$ that commute through the S-box layer $\mathcal{S}$ are (up to a permutation of the S-boxes) exactly the direct products of $A_1, \ldots, A_\ell$ and $B_1, \ldots, B_\ell$ such that (A_i, B_i) commute through the S-box S, and finding all (A_i, B_i) that commute through S is an already-solved problem. Hence, we can reduce the problem of finding probability-one commutative trails to (efficiently) checking if there exists an arrangement of the (A_i, B_i) such that $\mathcal{L} \circ \mathrm{Diag}(A_{i_1}, \ldots, A_{i_\ell}) = \mathrm{Diag}(B_{i_1}, \ldots, B_{i_\ell}) \circ \mathcal{L}$. As we have already seen in Section 4.3, this is equivalent to

$$\mathcal{L} \circ \mathrm{Diag}(L_{A_{i_1}}, \ldots, L_{A_{i_\ell}}) = \mathrm{Diag}(L_{B_{i_1}}, \ldots, L_{B_{i_\ell}}) \circ \mathcal{L} \quad \text{and}$$

$$\mathcal{L}(c_{A_{i_1}}, \ldots, c_{A_{i_\ell}}) = (c_{B_{i_1}}, \ldots, c_{B_{i_\ell}}).$$

To prevent iterating all choices of $\mathrm{Diag}(A_{i_1}, \ldots, A_{i_\ell})$, we note that for any $L_{i,j}, M_l, M'_k \in \mathbf{M}(n', \mathbb{F}_2)$

$$\begin{bmatrix} L_{1,1} & \ldots & L_{1,d} \\ \vdots & \ddots & \vdots \\ L_{d,1} & \ldots & L_{d,d} \end{bmatrix} \mathrm{Diag}(M_1, \ldots, M_d) = \mathrm{Diag}(M'_1, \ldots, M'_d) \begin{bmatrix} L_{1,1} & \ldots & L_{1,d} \\ \vdots & \ddots & \vdots \\ L_{d,1} & \ldots & L_{d,d} \end{bmatrix}$$

implies $L_{i,i} M_i = M'_i L_{i,i}$, which allows us to filter A_i and B_i using a divide-and-conquer approach, see Algorithm 1. Furthermore, if two rounds of the cipher operate on independent parts, which are often referred to as superboxes, then we can analyze those parts independently, as long as they themselves do have non-trivial linearity and differential uniformity.

Note that we see any constant addition as part of the key schedule here, and that the trails *only* hold *if* the round keys are within the class of weak keys. Additionally, knowing all possible two round trails enables us to combine them and exhaustively list all possible trails for any given number of rounds. We would like to mention that for most of the ciphers a sagemath[The23] implementation[4] of Algorithm 1 takes less than a second to

[4]The implementation will be made public after acceptance.

Algorithm 1 Searching Two Round Commutative Trails

Require: S-box S of size m, Linear Layer $\mathcal{L}$, number of S-boxes $\ell = 2^d$
Ensure: All trails T_1 over $\mathcal{L}$ that can be extended to trails over $\mathcal{S} \circ \mathcal{L} \circ \mathcal{S}$
Compute $E \leftarrow \{(A, B) | S \circ A = B \circ S\}$
For every i let $T_i \leftarrow \{(B, A') | (A, B), (A', B') \in E\}$ ▷ $T_1 \times \cdots \times T_\ell$ represents a superset of all trails over $\mathcal{L}$ that can be extended to ones over $\mathcal{S} \circ \mathcal{L} \circ \mathcal{S}$
for $b = 0, \ldots, d$ **do** ▷ Consider blocks that contain 2^b S-boxes
  Split $\mathcal{L}$ into the $m \cdot 2^b \times m \cdot 2^b$ blocks $L_{i,j}$
  for $i = 1, \ldots, 2^{d-b}$ **do** ▷ For each (diagonal) block
    for $(B, A') \in T_i$ **do** ▷ Filter trails block wise
      if $L_{i,i} L_B \neq L_{A'} L_{i,i}$ **then**
        Remove (B, A') from T_i
      end if
    end for
  end for
  if $b \neq d$ **then** ▷ Double block size covered by each T_i, if possible
    $T_i \leftarrow \{(\mathrm{Diag}(B_1, B_2), \mathrm{Diag}(A'_1, A'_2)) | (B_1, A'_1) \in T_{2i-1}, (B_2, A'_2) \in T_{2i}\}$ for $i = 1, \ldots, 2^{d-b-1}$
  end if
end for ▷ It is now ensured that $\mathcal{L} L_B = L_{A'} \mathcal{L}$ for every $(B, A') \in T_1$
for $(B, A') \in T_1$ **do** ▷ Filter out by constants
  if $\mathcal{L} c_B \neq c_{A'}$ **then**
    Remove (B, A') from T_1
  end if
end for
return T_1

finish on a personal laptop (ignoring some pre-computation for setting up the linear-layer) .[5] For the others, it still takes less than two minutes to calculate all possible commutative trails after the affine self equivalences are found (which we do using the algorithm of Biryukov *et al.* [BDBP03]). The only outlier here is Rectangle, where the three non-trivial self equivalences are found immediately, but the algorithm still takes around 30 minutes to finish. Nevertheless, the highest total execution time we have seen is for AES, which is still below 40 minutes.

We tested our algorithm on AES, Ascon, Boomslang, Craft, Gift-$\{64, 128\}$, iScream, Kuznyechik, LED, Mantis, Midori64, Pride, Prince, Present, Rectangle, Scream, Skinny-$\{64, 128\}$ and Streebog. Out of the tested ciphers, we only find (non-trivial) trails over at least two rounds for Scream, as well as Mantis and Midori, both of which use the same 4-bit S-box and the same linear layer. These trails (over the superboxes) for the later two are trails of the form $\mathcal{A} \xrightarrow{\mathcal{S}} \mathcal{B} \xrightarrow{\mathcal{L}} \mathcal{B}$. The first trail is such that $A = B = A_1$ where

$$A_1 := x \mapsto \begin{bmatrix} 1 & . & . & . \\ . & . & . & 1 \\ . & . & 1 & . \\ . & 1 & . & . \end{bmatrix} \cdot x + \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

We will discuss it further, first in Section 6.1.1, and Supplementary Material A.3.4. The second kind of trail is such that $(A, B) \in \{(A_2, A_3), (A_3, A_2)\}$ where

$$A_2 := x \mapsto \begin{bmatrix} 1 & 1 & . & 1 \\ . & 1 & . & . \\ . & . & 1 & . \\ . & . & . & 1 \end{bmatrix} \cdot x + \begin{bmatrix} . \\ 1 \\ . \\ 1 \end{bmatrix}, A_3 := x \mapsto \begin{bmatrix} 1 & 1 & . & 1 \\ . & . & . & 1 \\ . & . & 1 & . \\ . & 1 & . & . \end{bmatrix} \cdot x + \begin{bmatrix} 1 \\ . \\ 1 \\ . \end{bmatrix}.$$

Note that all trails have the same class of weak keys. However, because the key schedule of Mantis is heavy (especially its dense round constants), the weak key space seems incompatible with the genuine scheduling. Thus, we focus our efforts on Midori/Vert whose constants are sparser. In the case of Scream, we show that the one (non-trivial) trail found (see Section 6.1.2) is compatible with its key schedule, and can propagate over an arbitrary number of rounds for 2^{80} out of the 2^{128} possible keys.

6 Applications

As we just established, Midori and Scream seem like promising targets for attacks leveraging commutative behaviours. For Midori, the commutative patterns briefly mentioned above and presented in Section 6.1.1 are in line with the observations made on Vert using a

[5] The laptop is equipped with an Intel Core i7-1165G7 @ 2.80GHz × 4

different framework, which are depicted in Supplementary Material A.1. Naturally, in this section, we study attacks based on commutative trails for Midori/Vert and Scream in more detail. Our effort is here focused on distinguishers.

More specifically, in the case of Vert, we can compare our results to the complexity estimates obtained using classical wide-trail strategy arguments, as given in the specification paper of Midori. Those arguments do not take the constants into account, and would therefore hold for any Vert variants. Our results show one of the limits of such an argument. Indeed, despite the correctness of this argument, we will establish the following properties:

- for 2^{96} out of 2^{128} keys, there exists a probability-1 commutative trail covering an arbitrary number of rounds in $\mathsf{Vert}^2_{\mathrm{SC}}$ (and $\mathsf{Vert}^2_{\mathrm{SR}}$),
- for the same keys, this pattern implies a truncated differential covering an arbitrary number of rounds with probability 2^{-16}, and
- for 2^{120} out of 2^{128} keys, there exists for $\mathsf{Vert}^2_{\mathrm{SR}}$ a commutative trail with probability 2^{-4r} over r rounds, which is essentially the square root of the wide trail bound.

These properties are summarized in Figure 2. We first present in Section 6.1 the probability-one results for Vert, as well as for Scream. A complementary probability-one distinguisher for Grün is presented in Supplementary Material B.3. Then the probabilistic behaviors are presented in Sections 6.2 and 6.3.

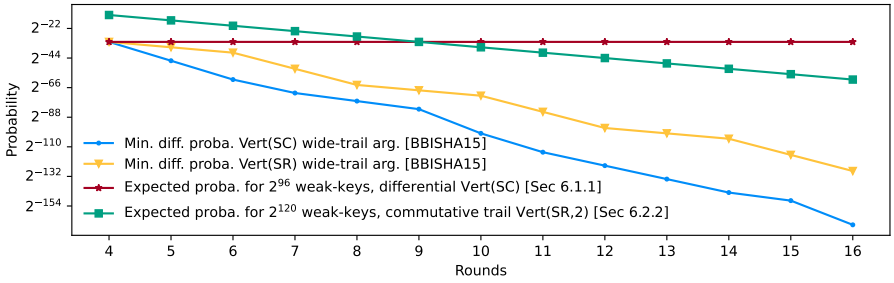

Figure 2: Comparison of the complexities of our attacks with wide-trail argument bounds.

6.1 Probability-One Commutative Trails

6.1.1 Probability-one Trail in Vert

We consider here A_1 (and $\mathcal{A}_1$), where A_1 is the mapping introduced in Section 5 and detail the distinguisher we obtain from it.

First, one can easily verify that $A_1 \circ S = S \circ A_1$.[6] Then, according to Section 4.3, Condition 1 of Theorem 1, is already satisfied by the binary Midori MixColumn. So is Condition 2 because *any* vector (c, c, c, c), with $c \in \mathbb{F}_2^4$ is a fixed point of M. This immediately implies that $\widehat{c_{A_1}} \in \mathrm{Fix}(\mathcal{L}_{A_1})$, and thus $\mathcal{L} \circ \mathcal{A}_1 = \mathcal{A}_1 \circ \mathcal{L}$. Finally, according to Corollary 1, because $A = B = A_1$ here, there exist 1-weak keys and they correspond to the fixed points of L_{A_1}: $\mathrm{Fix}(L_{A_1}) = \{x \in \mathbb{F}_2^4 \mid x_1 = x_3\} = \langle \mathtt{0x2}, \mathtt{0x5}, \mathtt{0x8} \rangle =: \mathtt{V}$. Overall, with a one-bit condition for 16 nibbles of both half of the key (i.e. for 2^{96} keys out of 2^{128}), we get a distinguishing self-similarity property for $\mathsf{Vert}^2_{\mathrm{SC}}$ (and $\mathsf{Vert}^2_{\mathrm{SR}}$) because $2 \in V$:

$$\forall\, K \in V^{32}, \forall\, x \in \mathbb{F}_2^{128}, \quad \mathcal{A}_1 \circ \mathsf{Vert}^2_{\mathrm{SC}}\,(K, x) = \mathsf{Vert}^2_{\mathrm{SC}}\,(K, \mathcal{A}_1\,(x))\,.$$

This weak-key space is, to be the best of our knowledge, new. Yet it is striking to observe the similiaries with weak-key spaces already present in the literature. Indeed,

[6] A script used to verify such statements is provided with the submission.

the non-linear invariant attack from Todo, Leander & Sasaki [TLS19] on $\mathsf{Vert}^5_{\mathrm{SC}}$ also works on $\mathsf{Vert}^2_{\mathrm{SC}}$, and has 2^{64} weak keys: $\langle \mathtt{0x2}, \mathtt{0x5} \rangle^{32}$. The same holds for the non-linear invariant presented by Beyne [Bey18, Bey20] which works for $\mathsf{Vert}^2_{\mathrm{SC}}$, given that $K \in \{K_0 || K_1 \mid K_0 \in \langle \mathtt{0x2}, \mathtt{0x8} \rangle^{16}$ or $\mathtt{K_1} \in \langle \mathtt{0x2}, \mathtt{0x8} \rangle^{16}\}$. This naturally opens the question of how to establish a unified framework to look at all this sets as one.

However, it is also important to note that A_1, and thus $\mathcal{A}_1$, have no fixed point. In that case, the invariant subspace obtained from the fourth item of Lemma 1 is empty. With this in mind, and compared to invariant subspaces distinguishers, such an affine-self-similarity relation appears to be *fundamentally* different, and in this particular case, stronger.

Remark 1. Regarding A_2 and A_3 defined in Section 5, they verify $A_2 \circ S = S \circ A_3$ (as well as $A_1 = A_2 \circ A_3 = A_3 \circ A_2$). We further get $\mathrm{Fix}(L_{A_1}) = \mathrm{Fix}(L_{A_2}) = \mathrm{Fix}(L_{A_3})$, so the same distinguisher applies in that case, except that the trail will be an alternating one: $\mathcal{A}_2 \to \mathcal{A}_3 \to \cdots \to \mathcal{A}_2$.

6.1.2 Weak Keys in Scream

Scream [GLS+15] is a 128-bit-state and 128-bit-key tweakable block cipher of the LS-design category. Its 128-bit state can be viewed as an 8×16 matrix. The S-box layer consists in applying a unique 8-bit S-box in parallel on each column, while the linear layer consists in applying a unique 16-bit linear permutation (called L-box) on each row. At each round, round constants are added to the first row of the state, the key is added to the state (and the tweak, that we consider to be equal to 0 here, is added on the first 4 rows). For further details, we refer to the CAESAR competition [CAE14] submission document.

Using our tweaked version of the algorithm of Biryukov *et al.*, we found out that the 8-bit affine permutation A_4, that is defined below, commutes with the Scream S-box.

$$A_4 := x \mapsto \begin{bmatrix} 1 & . & . & . & 1 & . & 1 & . \\ . & 1 & . & . & 1 & . & . & . \\ . & . & 1 & . & . & . & . & . \\ . & . & . & 1 & 1 & 1 & 1 & . \\ . & . & . & . & 1 & . & . & . \\ . & . & . & . & . & 1 & . & . \\ . & . & . & . & . & . & 1 & . \\ . & . & . & . & . & . & . & 1 \end{bmatrix} \cdot x + \begin{bmatrix} . \\ . \\ 1 \\ . \\ . \\ . \\ . \\ 1 \end{bmatrix}.$$

We further observe that $\mathrm{Fix}(L_{A_4}) = \langle \mathtt{0x01}, \mathtt{0x10}, \mathtt{0x20}, \mathtt{0x40}, \mathtt{0x80} \rangle$. As the round constants are added on the least significant row and as $\mathtt{0x01}$ (and $\mathtt{0x00}$) belongs to $\mathrm{Fix}\,(L_{A_4})$, we immediately observe that the round constants belong to $\mathrm{Fix}(L_{\mathcal{A}_4})$, and thus are 1-weak constants, according to Corollary 1. Then, as observed in Section 4.3, Condition 1 of Theorem 1 is verified for $\mathcal{A}_4$, because $\mathcal{L}$ is only made of null and identity matrix blocks. Finally, $c_{A_4} = \mathtt{0b00100001}$, so $\widehat{c_{A_4}}$ is composed of two all-1 rows and six all-0 rows. But one can easily verify that the all-1 vector (and obviously the all-0 one) is a fixed point of the L-box of Scream as the columns of its matrix given in [GLS+15] all have an odd Hamming weight. This means that $\widehat{c_{A_4}}$ is a fixed point of $\mathcal{L}$. Applying the same reasoning as for Vert^2_{SC}, we obtain a probability-one distinguisher for $2^{128-3\times 16} = 2^{80}$ keys because of the sixteen 3-bit conditions.

Unlike our attacks against Vert, this one can be applied to the "real" primitive without modifying its key schedule. However, the weak keys we obtain are a strict subset of those obtained in [TLS19], where the non-linear invariant attack they mount works provided that 2 rows of the key are constrained (while we need to constrain one more).

6.1.3 From Commutative Patterns to Very High Probability Differentials

Let us take a step back to Vert and look more carefully at A_1. We can see that for any $x \in V$, we get $A_1(x) = x + \mathtt{0xf}$ and for any $x \in \mathbb{F}_2^4 \setminus V$, $A_1(x) = x + \mathtt{0xa}$. This means that $x + A_1(x) \in U$, where $U := \{\mathtt{0xa}, \mathtt{0xf}\}$, each equality holding with a probability of $1/2$. As a consequence, by picking a random state $x \in \mathbb{F}_2^{64}$ and looking at the pair $(x, x + \mathtt{0xf}^{\times 16})$, we actually have a pair of the form $(x, \mathcal{A}_1(x))$ with a probability of 2^{-16}. However, as we

established, $\mathcal{A}_1$ commutes with any number of rounds of Vert (provided that the constant and key nibbles are all in V). This means that the final difference is necessarily of the form $y + \mathcal{A}_1(y)$, and thus has to lie in the small set U^{16}. As a consequence, for weak keys there exists a truncated differential of the form $\texttt{0xf}^{16} \to \texttt{U}^{16}$ that has probability 2^{-16} over an *arbitrary* amount of rounds! We have experimentally verified this surprising property.

This observation raises multiple striking points. First of all, the cost function of such a truncated differential is remarkable: it is independent of the number of internal rounds and only depends on the cost of the first round[7]. Moreover, as we also know that any internal difference on any nibble has the form $x + A(x) \in U$, we are also assured that *all the S-boxes of each round will be differentialy active* because $0 \notin U$. The discrepancy between the bound on the differential probability obtained via a wide-trail argument and our result is illustrated in Figure 2. It is not only yet another example of how much the fixed-key behavior can deviate from the expected average computed with standard wide-trail strategy arguments, but even more a high differential independent of the number of rounds.

The same actually happens when the genuine Scream is used with a 1-weak key. Indeed, $I + A_4$ can take only eight values; we denote U' this set of values, and let $\alpha \in U'^{16}$. In that case, for a random plaintext x, the pair $(x, x + \alpha)$ has a probability of $8^{-16} = 2^{-48}$ to coincide with $(x, \mathcal{A}_4(x))$; the corresponding truncated differential $\alpha \to U'$ has thus a probability of 2^{-48}, independently of the number of rounds. Because $0 \notin U'^{16}$, we are again assured that all the S-boxes will be differentially active.

6.2 Probabilistic Trails: a Probability/Number of Weak Keys Trade-off

6.2.1 Probabilistic Trails through the Linear Layer

As we have seen, the size of the weak-key space is driven by the number of *commutatively* active S-boxes. It is tempting to try to use a partial $\mathcal{A}$ instead of the full one, as we did for now. Indeed, decreasing the number of active S-boxes will limit the number of constraints on the key and thus increase the number of weak keys. However, this change imposes a counter-intuitive concept: that of *active linear layers*. Indeed, such a partial affine permutation do not commute with matrix multiplications with probability one.

Let $i_1, \cdots, i_4, j_1, \cdots, j_4 \in \{0, 1\}$. We denote $A^0 := I$ and $A^1 = A$ and define $\widetilde{\mathcal{A}}_{i_1,i_2,i_3,i_4} := \mathrm{Diag}(A^{i_1}, A^{i_2}, A^{i_3}, A^{i_4})$. We study $M \circ \widetilde{\mathcal{A}}_{i_1,i_2,i_3,i_4}(x) = \widetilde{\mathcal{A}}_{j_1,j_2,j_3,j_4} \circ M(x)$ by developing it, and observe that it is equivalent to the following equality:

$$\underbrace{\begin{bmatrix} \cdot & \delta_{j_1,i_2}\widetilde{L_A} & \delta_{j_1,i_3}\widetilde{L_A} & \delta_{j_1,i_4}\widetilde{L_A} \\ \delta_{j_2,i_1}\widetilde{L_A} & \cdot & \delta_{j_2,i_3}L_A & \delta_{j_2,i_4}\widetilde{L_A} \\ \delta_{j_3,i_1}\widetilde{L_A} & \delta_{j_3,i_2}\widetilde{L_A} & \cdot & \delta_{j_3,i_4}L_A \\ \delta_{j_4,i_1}L_A & \delta_{j_4,i_2}L_A & \delta_{j_4,i_3}\widetilde{L_A} & \cdot \end{bmatrix}}_{B(i,j)} x = c_A \left(M \begin{bmatrix} i_1 \\ i_2 \\ i_3 \\ i_4 \end{bmatrix} + \begin{bmatrix} j_1 \\ j_2 \\ j_3 \\ j_4 \end{bmatrix} \right),$$

where $\widetilde{L_A} = L_A + I$ and $\delta_{i,j}$ is the Kronecker delta: $\delta_{i,j} = 1$ if $i = j$ and 0 otherwise.

As we can see, should the right-hand side not be in the image of $B(i, j)$, then the transition $\widetilde{\mathcal{A}}_{i_1,i_2,i_3,i_4} \xrightarrow{M} \widetilde{\mathcal{A}}_{j_1,j_2,j_3,j_4}$ would be impossible. Otherwise, the number of x that satisfy this relation is given by the size of the kernel of $B(i, j)$, and can be deduced from the size of its image. The dimension of this image is given in Table 1.

6.2.2 Application to $\mathsf{Vert}^2_{\mathbf{SR}}$

The counter-part of choosing a partial affine layer is the necessary handling of the cells permutation, which could be ignored beforehand. Hence, because the ShuffleCell permutation

[7] Actually, any differential $\alpha \to \beta$ where $\alpha, \beta \in U^{16}$ has a probability of $2^{-16\times 2} \times 2^{-16} = 2^{-32}$ and this probability only depends on the cost of the first and final round.

Table 1: Dimension of $\mathrm{Im}(B(i,j))$. All entries must be multiplied by $\dim \mathrm{Im}(\widetilde{L_A})$. When written in bold, the commutation holds for all c_A, otherwise, we need $c_A \in \mathrm{Im}(I + L_A)$.

	1	2	4	8	3	5	6	9	a	c	7	b	d	e	f
1	2	2	2	2	3	3	3	3	3	3	4	4	4	**2**	3
2	2	2	2	2	3	3	3	3	3	3	4	4	**2**	4	3
4	2	2	2	2	3	3	3	3	3	3	4	**2**	4	4	3
8	2	2	2	2	3	3	3	3	3	3	**2**	4	4	4	3
3	3	3	3	3	**2**	4	4	4	4	4	3	3	3	3	2
5	3	3	3	3	4	**2**	4	4	4	4	3	3	3	3	2
6	3	3	3	3	4	4	**2**	4	4	4	3	3	3	3	2
9	3	3	3	3	4	4	4	**2**	4	4	3	3	3	3	2
a	3	3	3	3	4	4	4	4	**2**	4	3	3	3	3	2
c	3	3	3	3	4	4	4	4	4	**2**	3	3	3	3	2
7	4	4	4	**2**	3	3	3	3	3	3	2	2	2	2	1
b	4	4	**2**	4	3	3	3	3	3	3	2	2	2	2	1
d	4	**2**	4	4	3	3	3	3	3	3	2	2	2	2	1
e	**2**	4	4	4	3	3	3	3	3	3	2	2	2	2	1
f	3	3	3	3	2	2	2	2	2	2	1	1	1	1	**0**

is stronger than the ShiftRows permutation, what follows only applies to $\mathsf{Vert}^2_{\mathrm{SR}}$.

The square activity pattern (see Equation 3), which has already been used for instance against PRINCE [CFG+15]), is preserved by the classical ShiftRows.

$$\begin{bmatrix} \mathrm{x} & . & \mathrm{x} & . \\ . & . & . & . \\ \mathrm{x} & . & \mathrm{x} & . \\ . & . & . & . \end{bmatrix} \tag{3}$$

As a consequence, if we consider commutants that are inactive everywhere (the identity mapping) except on these nibbles which are all activated with the same mapping A, then we can build iterated commutative trails. As just established, commutation with the ShiftRows operation holds with probability one. In order to go through the S-box layer with probability one, we use the already-introduced affine permutation A_1 on the active nibbles, and denote $\widetilde{\mathcal{A}_1}$ the corresponding partially-active mapping. Then, in order to study the transition $\widetilde{\mathcal{A}_1} \xrightarrow{\mathcal{M}} \widetilde{\mathcal{A}_1}$, we first examine, thanks to Table 1, the transition of a single partially-active column the MixColumn operation M. This activity pattern corresponds to the case $i = j = 5$ and $\dim(\mathrm{Im}(\widetilde{L_A})) = 1$: in that case $\mathrm{Im}(B(i,j))$ has dimension 2. Then, $\widetilde{\mathcal{A}_1} \xrightarrow{\mathcal{M}} \widetilde{\mathcal{A}_1}$ occurs with probability $(2^{-2})^2 = 2^{-4}$ because of the two active columns.

Thus, *assuming independence of the rounds*, $\mathsf{Vert}^2_{\mathrm{SR}}$ has commutative trails with probability 2^{-4r}, where r is the number of full rounds (rounds with $\mathcal{M}$ involved). This probabilistic behaviour is counter-balanced by the size of the weak-key space. Indeed, because the square activity pattern only involves four active nibbles, only the corresponding nibbles of the key need to be constrained[8](they must belong to $\mathrm{Fix}(A_1)$). A significantly bigger weak-key space is thus obtained: with four 1-bit constraints on each half of the key (K_0 and K_1), the number of 2^{-4r}-weak keys becomes $2^{128-4\times 2} = 2^{120}$.

6.2.3 Experimental Results

These high probabilities enable to thoroughly test experimentally our distinguishers.

Experiment 1 We picked uniformly at random (weak key, plaintext) pairs (K, x) and verified whether $\widetilde{\mathcal{A}} \circ \mathcal{E}_K(x) = \mathcal{E}_K \circ \widetilde{\mathcal{A}}(x)$, where $\mathcal{E}$ stands for a round-reduced $\mathsf{Vert}^2_{\mathrm{SC}}$ or $\mathsf{Vert}^0_{\mathrm{SC}}$. This estimates the probability of the hull but not the one of the trail, which is the one mentioned in the previous section. As experimenters with full-access, we also focus on the trail by studying $\bar{\mathcal{R}}_i \circ \cdots \circ \bar{\mathcal{R}}_1 \circ \widetilde{\mathcal{A}}(x) \stackrel{?}{=} \widetilde{\mathcal{A}} \circ \bar{\mathcal{R}}_i \circ \cdots \circ \bar{\mathcal{R}}_1(x)$ for all $i \in \{1, \cdots, r\}$. We repeated the draw (1 key *and* 1 plaintext) 2^{36} times, expecting an average number of solutions of $2^{36-4(r-1)}$ for the r-round version.

[8]Yet, we keep weak round constants for all nibbles, to stay as close as possible from the genuine Midori.

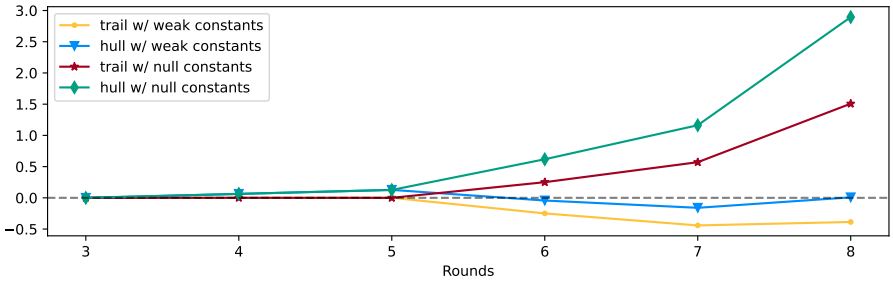

Figure 3: Evolution of the experimental average throughout the rounds. $r \mapsto (\mathrm{EA}(r) - \mathrm{TA}(r))/\mathrm{TA}(r)$, where EA, TA stands for experimental and theoretical average.

As we can see on Figure 3, the behaviour of the experimental average is more intricate as one could first think. In the weak-constant setting and as the rounds go, the experimental average of the trail seems to slowly decrease compared to the theoretical average. However, the behavior of the hull stays really close to the theoretical average for the trail. This seems to indicate the not-so-surprising fact that the round independence hypothesis is probably too strong in some cases at the trail level. However, even if the dominance of the trail among the hull slowly vanishes, the hull effect becomes stronger and compensates this drop. The difference between the null constants (a particular class of weak constants) and the weak ones is also pointed out by Figure 3. With no round constant, no asymmetry are input at each round. This could be the reason why in that scenario, if a pair $(x, \widetilde{\mathcal{A}}(x))$ goes through a few rounds, it has a higher probability of continuing going further.

Experiment 2 We also studied the fixed-weak-key setting, for multiple round-reduced versions. We picked uniformly at random a weak key and a *set* of plaintexts $(K, \mathbb{X})$, and observed whether $\widetilde{\mathcal{A}} \circ \mathsf{Vert}^2_{\mathrm{SC}}(K, x) \stackrel{?}{=} \mathsf{Vert}^2_{\mathrm{SC}}\left(K, \widetilde{\mathcal{A}}(x)\right)$ or $\bar{\mathcal{R}}_i \circ \cdots \circ \bar{\mathcal{R}}_1 \circ \widetilde{\mathcal{A}}(x) \stackrel{?}{=} \widetilde{\mathcal{A}} \circ \bar{\mathcal{R}}_i \circ \cdots \circ \bar{\mathcal{R}}_1(x)$ for all $i \in \{1, \cdots, r\}$ occured. For a fixed weak key, we drew $2^{4(r-1)+6}$ plaintexts, hoping for an average of 2^6 solutions. We repeated the experiment for 10000 weak keys, except for $r = 7$ for which we used 6000 weak keys.

Naturally, the average from Experiment 2 goes in the same direction as Experiment 1: as the rounds increase, the trail average moves away from the theoretical one while the hull average stays much closer. What really appears in Figure 4, is the fact that the average case taken over "all" weak keys and "all" plaintexts for the trail is not as representative as we could expect: the probability $p = 2^{-4r}$ seems appropriate for $r = 3$, however as r grows it seems that p-weak keys are rather p'-weak keys where p' can take a palette of values. For the hull, the distribution of p' seems to flatten as r grows and tends to a uniform distribution over $[0, 1]$. In particular, it is unclear why about half of the tested weak keys appears to be actually strong, while some others are weaker than expected.

An experiment studying the strongest keys is provided in Supplementary Material C. As just shown, the basic model seems to work well-enough to estimate the average probability and effectiveness of our distinguisher. However, it fails at explaining precisely the all-$\widetilde{\mathcal{A}}$ trail. Explaining the observed clustering, and understanding the sub-classes among the weak keys are two of the many open questions raised by our experimentation.

6.3 Probabilistic Trails through the S-box Layer

For 4-bit S-boxes, it is possible to study probabilistic relations of the form $A \circ S = S \circ B$ holding with high-but-not-1 probability where A, B are affine permutations, as they can be brute-forced. In the case of the S-box of Midori, the second highest value of $\Gamma_S(A, B)$

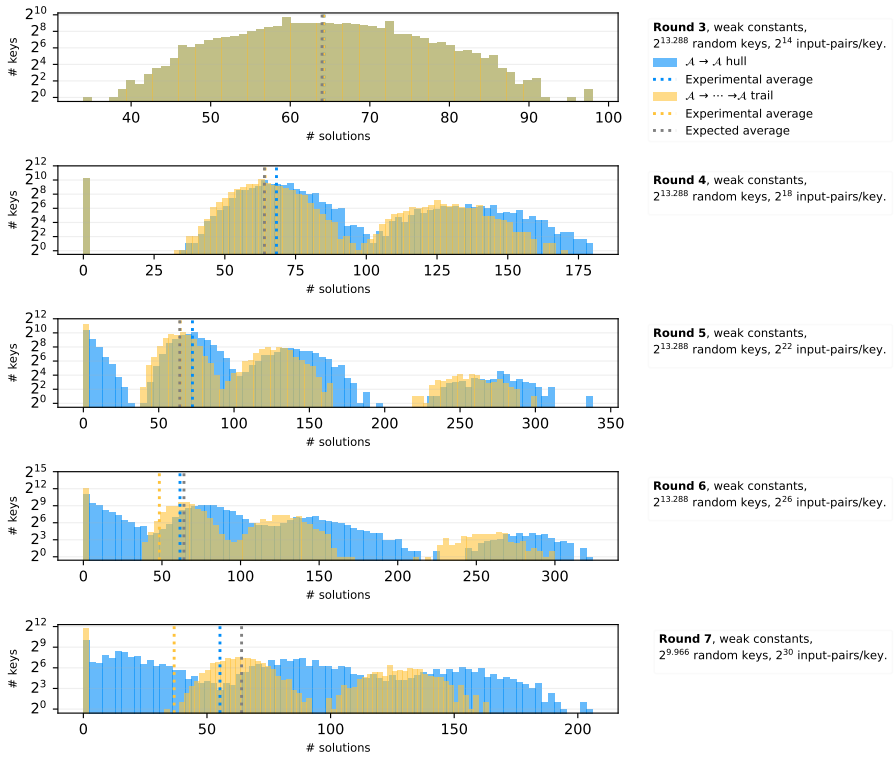

Figure 4: Fixed-key study: Estimation of the p-weakness through the numbers of solutions $\left(x, \widetilde{\mathcal{A}}(x)\right)$ following the trail/hull. The expected average is 2^6 for every number of rounds.

(after 16) we can find is 12. The corresponding mappings are $A = A_6, B = A_7$, where

$$A_6 := \{6, \mathtt{f}, 4, \mathtt{d}, 2, \mathtt{b}, 0, 9, \mathtt{e}, 7, \mathtt{c}, 5, \mathtt{a}, 3, 8, 1\},\ \mathtt{A_7} := \{1, 0, 3, 2, 8, 9, \mathtt{a}, \mathtt{b}, 6, 7, 4, 5, \mathtt{f}, \mathtt{e}, \mathtt{d}, \mathtt{c}\}.$$

Regarding the fixed points, L_{A_6} has 8 and L_{A_7} has 4. It also holds that $\Gamma_S(A_7, A_6) = 12$. We can reuse our previous framework to mount an alternating commutative trail based on the square pattern, as well as the corresponding distinguisher. In that case, the probability of going through the S-box layer is estimated as $2^{4\log_2\left(\frac{12}{16}\right)}$ as 4 S-boxes are activated. The probability of going through a full round should thus be $2^{4\left(\log_2\left(\frac{12}{16}\right)-1\right)}$, and the theoretical average mentioned in Figure 5 is computed as $2^{4\left(\log_2\left(\frac{12}{16}\right)-1\right)r+4\log_2\left(\frac{12}{16}\right)}$, because of the final round where the linear layer is omitted.

However, there is a significant divergence between our initial estimate and our experimental results: the probability of commuting is higher than expected under the assumption that S-box and linear layer transitions are independent. To get a better picture, we computed the probability of having $\widetilde{\mathcal{A}_i} \circ M \circ \mathcal{S} = M \circ \mathcal{S} \circ \widetilde{\mathcal{A}_j}$, where $\widetilde{\mathcal{A}_r}$ is a partial layer involving A_r, where $\{i, j\} = \{6, 7\}$ and $r \in \{i, j\}$. Checking all 2^{16} possible inputs, we have that the two transitions happen with probabilities $2^{-5.6}$ (which is coherent with our estimate), and $2^{-8.8}$ (instead of $2^{-9.6}$). Understanding the rest of the difference is an interesting open problem, but we expect dependencies between the round probabilities.

We also found $A_8 = \{4, 5, 6, 7, 0, 1, 2, 3, \mathtt{a}, \mathtt{b}, 8, 9, \mathtt{e}, \mathtt{f}, \mathtt{c}, \mathtt{d}\}$ which is such that $A_8 \circ S(x) = S \circ A_8(x)$ holds with probability 10/16. L_{A_8} has 8 fix points, but in that case 1 belong to them, so the genuine constants of Midori can be used.

In both of these cases, the probability of going through a round-reduced version of $\mathsf{Vert}_{\mathrm{SR}}$ is smaller than in Section 6.2.2, because of the new cost imposed by the probabilistic

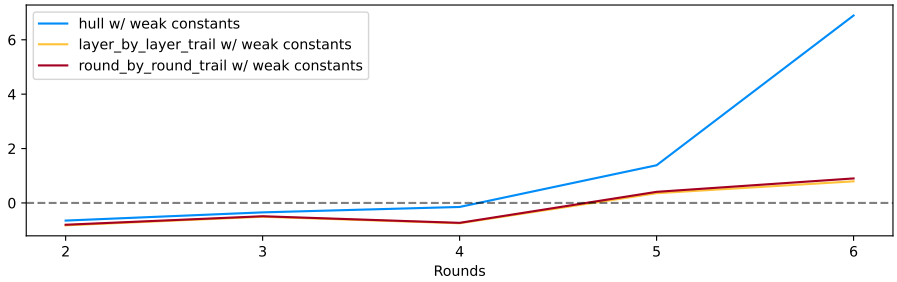

Figure 5: Experimental evaluations of $(\mathrm{EA}(r) - \mathrm{TA}(r))/\mathrm{TA}(r)$, where EA, TA stands for experimental and theoretical average respectively, for $r \in \{2, ..., 6\}$.

crossing of the S-box layer. The weak-key space, in the first example, is also smaller. While a priori less impressive than the results based on probability-one S-box transitions, this probabilistic case opens some very interesting open problems. Indeed, while we can quickly rule out the applicability of the probability-one case by ensuring the absence of non-trivial commuters for the S-box, there is no way at this stage to efficiently compute the A-uniformity of an S-box operating on more than 4 bits. Thus, we cannot be sure that primitives using 8-bit S-boxes are safe from non-probability-one commutative cryptanalysis.

6.4 High Probability Commutants in the AES Super-S-Box

The AES [AES01] is a 128-bit block cipher, arguably the most important primitive in symmetric cryptography due to its wide use, and well-trusted security. As shown by Gilbert and Peyrin in [GP10], two rounds of this primitive can be seen as the application of a layer of *Super S-boxes* followed by an affine layer. Here, we focus on said *Super S-boxes*: they are permutation of $(\mathbb{F}_2^8)^4$ obtained by composing:

1. a layer of four parallel S-boxes $S : \mathbb{F}_2^8 \to \mathbb{F}_2^8, x \mapsto C \circ \mathsf{Inv}$, where C is an affine permutation (we shorten $c := c_C$) and Inv is the multiplicative inversion in $\mathbb{F}_{256}$,
2. a key addition, [9]
3. a multiplication of the internal state by the MC matrix operating on $\mathbb{F}_{256}^4$, and
4. another application of the S-box layer.

We denote Mult_λ the multiplication by λ in $\mathbb{F}_{256}$, and recall that F^G denotes $G \circ F \circ G^{-1}$. Since S is essentially a monomial, it is tempting to investigate its commutative behaviour with a multiplication as input.[10] This behaviour is predicted by the following lemma.

Lemma 2. *Let $G = B \circ F \circ A$ be a permutation of $\mathbb{F}_{2^m}$, where A and B are affine permutations, and where $F : x \mapsto x^d$ is a power permutation of $\mathbb{F}_{2^m}$. Then, for any $\lambda \in \mathbb{F}_{2^m}$ such that $\lambda \neq 0$, we have $G \circ \mathsf{Mult}_\lambda^{A^{-1}} = \mathsf{Mult}_{\lambda^d}^B \circ G$.*

Proof. First, $G \circ \mathsf{Mult}_\lambda^{A^{-1}} = B \circ F \circ A \circ A^{-1} \circ \mathsf{Mult}_\lambda \circ A$, which we rewrite using that $F \circ \mathsf{Mult}_\lambda = \mathsf{Mult}_{\lambda^d} \circ F$ to obtain that $G \circ \mathsf{Mult}_\lambda^{A^{-1}} = B \circ \mathsf{Mult}_{\lambda^d} \circ F \circ A$. We then decompose the right-hand side into $B \circ \mathsf{Mult}_{\lambda^d} \circ B^{-1} \circ B \circ F \circ A$, and deduce the lemma. □

For the AES S-box, we obtain that $S \circ \mathsf{Mult}_\lambda = \mathsf{Mult}_{1/\lambda}^C \circ S$, for any $\lambda \in \mathbb{F}_{256} \setminus \{0\}$. Since the linear layer is a simple matrix multiplication, it commutes with four Mult_λ

[9] Normally, the key addition should be after the linear layer. However, they can be safely swapped (provided that the key is replaced by its image through the inverse linear layer).

[10] It is well known that a "naive" AES variant where C is removed from the S-box can have sophisticated interaction with e.g. the Frobenius automorphisms. Our analysis does not rely on such a simplification.

applied in parallel. As a consequence, we investigate commutative trails of the form

$$\mathsf{Mult}_{1/\lambda} \xrightarrow{\mathcal{S}} \mathsf{Mult}^C_\lambda \xrightarrow{T_k} \mathsf{Mult}_\mu \xrightarrow{\mathsf{MC}} \mathsf{Mult}_\mu \xrightarrow{\mathcal{S}} \mathsf{Mult}^C_{1/\mu} \ ,$$

where all transitions have probability 1 except for $\mathsf{Mult}^C_\lambda \xrightarrow{T_k} \mathsf{Mult}_\mu$. We consider that $\lambda \neq 1$, otherwise we only get a trivial result. Applying Proposition 1 we deduce that the set of p-weak keys is of size $|\ker(I + \mathsf{Mult}_\mu)| \times \left|\mathrm{Im}(I + \mathsf{Mult}_\mu) \cap \mathrm{Im}(\mathsf{Mult}^C_\lambda + \mathsf{Mult}_\mu)\right|$. The set $\ker(I + \mathsf{Mult}_\mu)$ is trivial because $\mu x = x$ is equivalent to $x = 0$ as $\mu \neq 1$. This implies that $\mathrm{Im}(I + \mathsf{Mult}_\mu)$ is the full field. We deduce that the number of p-weak keys is equal to $|\mathrm{Im}(\mathsf{Mult}^C_\lambda + \mathsf{Mult}_\mu)| = |\mathrm{Im}(\mathsf{Mult}^{L_C}_\lambda + \mathsf{Mult}_\mu)|$, and that $p = 1/|\mathrm{Im}(\mathsf{Mult}^{L_C}_\lambda + \mathsf{Mult}_\mu)|$.

We then have a trade-off: if $|\mathrm{Im}(\mathsf{Mult}^{L_c}_\lambda + \mathsf{Mult}_\mu)|$ is large, then the number of p-weak keys is large but they are not very weak as p is small. On the other hand, if $\mathrm{Im}(\mathsf{Mult}^{L_c}_\lambda + \mathsf{Mult}_\mu)$ is small, then there are very few p-weak keys, but they are very weak.

Experiment 4 For any pair (λ, μ) of $\mathbb{F}_{256}$, it is easy to compute if there exists any p-weak keys for the transition $\mathsf{Mult}^C_\lambda \xrightarrow{T_k} \mathsf{Mult}_\lambda$, and if so what the value of p is. For all $\lambda \notin \{0, 1\}$, there exists at least one μ such that keys that are either 2^{-5}-, 2^{-4}- or 2^{-2}-weak for the corresponding commutation. More precisely, there are 68 values of λ for which the weakest keys are 2^{-5}-weak, 180 for which they are 2^{-4}-weak, and 6 for which they are 2^{-2}-weak.

We can look at this property from an other angle: what is the probability that a random 32-bit key is at least p-weak for at least one commutation, thus yielding a commutation for the full super S-box? This requires that all bytes of the key be weak for the same pair (λ, μ). To estimate this probability, we sampled 2^{22} uniformly random 32-bit keys K, and then checked if there exists a pair (λ, μ) such that all its 8-bit cells are p-weak for it. Out of the 2^{22} keys K we looked at, there existed a pair (λ, μ) such that K is at least 2^{-16}-weak in 6430 cases, meaning a density of $2^{-9.35}$. We observed 2^{-20}-weakness in 196 089 cases, hence a density of about $2^{4.35}$ for this weak-key space. We also computed directly the number of 2^{-8}-weak keys: there are 6 transitions for which such keys exist, and 4 keys allow each; hence, there are $6 \times 4^4 \approx 2^{10.58}$ very weak keys, *i.e.* a $2^{21.42}$ density.

These high probabilities show that commutative cryptanalysis is a powerful technique. Indeed, as established by Keliher & Sui [KS07], the maximum expected differential probability for the AES super S-box is $53/2^{32} \approx 2^{-26.27}$, which is much lower than the worst probabilities we considered. However commutative cryptanalysis only works for weak keys, but, as we established, this set can be of much higher density than we might expect.

7 Conclusion and Future Work

Our revisiting of commutative cryptanalysis, an idea that dates back almost 20 years, provides a rich structure and solid foundations for understanding and applying this approach concretely. While it allows to explain interesting phenomena in a compact and unifying way, it also leaves many open questions, possibilities for improvements and paths to follow.

Beyond the high probability patterns we found, the existence of weak-key truncated differentials in a Midori variant and in Scream whose probability is independent from the number of rounds, and does not correspond to any trail, challenges how usual security arguments are built. Indeed, a differential pattern over r rounds is *not* necessarily obtained by concatenating several high probability patterns over fewer rounds. This implicit but very common assumption should be used with care.

From an application point of view, as mentioned above, it is most important (and challenging) to develop algorithms that allow to compute the A-uniformity in cases where it is close to maximal. This would then potentially allow to find good, but not probability-one, commutative trails for large classes of ciphers – likely discovering new attacks, at least in

weak-key settings. A closely related but twisted question is if it is possible to hide the existence of a highly-probable commutative trail, potentially in larger than usual S-boxes. This might be a way of putting backdoors in otherwise secure ciphers that are very hard to find – even so the general principle would be known. Finally, similarly to the case of c-differentials, and even better motivated by attacks, studying A-uniformity from a Boolean function point of view seems rewarding. Concrete questions include the discussion on bounds on the A-uniformity and the construction of families of permutations with either maximal or minimal A-uniformity.

References

[AES01] Advanced Encryption Standard (AES). National Institute of Standards and Technology, NIST FIPS PUB 197, U.S. Department of Commerce, November 2001.

[AK19] Ralph Ankele and Stefan Kölbl. Mind the gap - A closer look at the security of block ciphers against differential cryptanalysis. In Carlos Cid and Michael J. Jacobson Jr:, editors, *SAC 2018: 25th Annual International Workshop on Selected Areas in Cryptography*, volume 11349 of *Lecture Notes in Computer Science*, pages 163–190. Springer, Heidelberg, August 2019.

[AL16] Tomer Ashur and Yunwen Liu. Rotational cryptanalysis in the presence of constants. *IACR Transactions on Symmetric Cryptology*, 2016(1):57–70, 2016. https://tosc.iacr.org/index.php/ToSC/article/view/535.

[BB02] Elad Barkan and Eli Biham. In how many ways can you write Rijndael? In Yuliang Zheng, editor, *Advances in Cryptology – ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 160–175. Springer, Heidelberg, December 2002.

[BBFL22] Christof Beierle, Tim Beyne, Patrick Felke, and Gregor Leander. Constructing and deconstructing intentional weaknesses in symmetric ciphers. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology – CRYPTO 2022, Part III*, volume 13509 of *Lecture Notes in Computer Science*, pages 748–778. Springer, Heidelberg, August 2022.

[BBI+15] Subhadeep Banik, Andrey Bogdanov, Takanori Isobe, Kyoji Shibutani, Harunaga Hiwatari, Toru Akishita, and Francesco Regazzoni. Midori: A block cipher for low energy. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology – ASIACRYPT 2015, Part II*, volume 9453 of *Lecture Notes in Computer Science*, pages 411–436. Springer, Heidelberg, November / December 2015.

[BBL21] Christof Beierle, Marcus Brinkmann, and Gregor Leander. Linearly self-equivalent APN permutations in small dimension. *IEEE Trans. Inf. Theory*, 67(7):4863–4875, 2021.

[BCC19] Christina Boura, Anne Canteaut, and Daniel Coggia. A general proof framework for recent AES distinguishers. *IACR Transactions on Symmetric Cryptology*, 2019(1):170–191, 2019.

[BCL18] Christof Beierle, Anne Canteaut, and Gregor Leander. Nonlinear approximations in cryptanalysis revisited. *IACR Transactions on Symmetric Cryptology*, 2018(4):80–101, 2018.

[BDBP03] Alex Biryukov, Christophe De Canniére, An Braeken, and Bart Preneel. A toolbox for cryptanalysis: Linear and affine equivalence algorithms. In Eli Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 33–50. Springer, Heidelberg, May 2003.

[BDLF10] Charles Bouillaguet, Orr Dunkelman, Gaëtan Leurent, and Pierre-Alain Fouque. Another look at complementation properties. In Seokhie Hong and Tetsu Iwata, editors, *Fast Software Encryption – FSE 2010*, volume 6147 of *Lecture Notes in Computer Science*, pages 347–364. Springer, Heidelberg, February 2010.

[Bey18] Tim Beyne. Block cipher invariants as eigenvectors of correlation matrices. In Thomas Peyrin and Steven Galbraith, editors, *Advances in Cryptology – ASIACRYPT 2018, Part I*, volume 11272 of *Lecture Notes in Computer Science*, pages 3–31. Springer, Heidelberg, December 2018.

[Bey20] Tim Beyne. Block cipher invariants as eigenvectors of correlation matrices. *Journal of Cryptology*, 33(3):1156–1183, July 2020.

[Bey21] Tim Beyne. A geometric approach to linear cryptanalysis. In Mehdi Tibouchi and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2021, Part I*, volume 13090 of *Lecture Notes in Computer Science*, pages 36–66. Springer, Heidelberg, December 2021.

[BFL+23] Christof Beierle, Patrick Felke, Gregor Leander, Patrick Neumann, and Lukas Stennes. On perfect linear approximations and differentials over two-round spns. Cryptology ePrint Archive, Paper 2023/725, 2023. https://eprint.iacr.org/2023/725.

[BM22] Emanuele Bellini and Rusydi H. Makarim. Functional cryptanalysis: Application to reduced-round xoodoo. Cryptology ePrint Archive, Report 2022/134, 2022. https://eprint.iacr.org/2022/134.

[BR22] Tim Beyne and Vincent Rijmen. Differential cryptanalysis in the fixed-key model. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology – CRYPTO 2022, Part III*, volume 13509 of *Lecture Notes in Computer Science*, pages 687–716. Springer, Heidelberg, August 2022.

[CAE14] CAESAR: Competition for Authenticated Encryption: Security, Applicability, and Robustness, March 2014. https://competitions.cr.yp.to/caesar.html.

[Car20] Claude Carlet. *Boolean Functions for Cryptography and Coding Theory*. Cambridge University Press, 11 2020. https://www.math.univ-paris13.fr/~carlet/book-fcts-Bool-vect-crypt-codes.pdf.

[CFG+15] Anne Canteaut, Thomas Fuhr, Henri Gilbert, María Naya-Plasencia, and Jean-René Reinhard. Multiple differential cryptanalysis of round-reduced PRINCE. In Carlos Cid and Christian Rechberger, editors, *Fast Software Encryption – FSE 2014*, volume 8540 of *Lecture Notes in Computer Science*, pages 591–610. Springer, Heidelberg, March 2015.

[CFG+17] Colin Chaigneau, Thomas Fuhr, Henri Gilbert, Jérémy Jean, and Jean-René Reinhard. Cryptanalysis of NORX v2.0. *IACR Transactions on Symmetric Cryptology*, 2017(1):156–174, 2017.

[DHP+18] Joan Daemen, Seth Hoffert, Michaël Peeters, Gilles Van Assche, and Ronny Van Keer. Xoodoo cookbook. Cryptology ePrint Archive, Paper 2018/767, 2018. https://eprint.iacr.org/2018/767.

[Din18] Itai Dinur. An improved affine equivalence algorithm for random permutations. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018, Part I*, volume 10820 of *Lecture Notes in Computer Science*, pages 413–442. Springer, Heidelberg, April / May 2018.

[EFR+20] Pål Ellingsen, Patrick Felke, Constanza Riera, Pantelimon Stănică, and Anton Tkachenko. C-differentials, multiplicative uniformity, and (almost) perfect c-nonlinearity. *IEEE Transactions on Information Theory*, 66(9):5781–5789, 2020.

[ERST21] Pål Ellingsen, Constanza Riera, Pantelimon Stănică, and Anton Tkachenko. An extension of the avalanche criterion in the context of c-differentials. In *Proceedings of the 18th International Conference on Security and Cryptography - Volume 1: SECRYPT,*, pages 460–467. INSTICC, SciTePress, 2021.

[GLS+15] Vincent Grosso, Gaëtan Leurent, François-Xavier Standaert, Kerem Varici, Anthony Journault, François Durvaux, Lubos Gaspar, and Stéphanie Kerckhof. SCREAM v3. Submission to CAESAR Competition, 2015.

[GLSV15] Vincent Grosso, Gaëtan Leurent, François-Xavier Standaert, and Kerem Varici. LS-designs: Bitslice encryption for efficient masked software implementations. In Carlos Cid and Christian Rechberger, editors, *Fast Software Encryption – FSE 2014*, volume 8540 of *Lecture Notes in Computer Science*, pages 18–37. Springer, Heidelberg, March 2015.

[GP10] Henri Gilbert and Thomas Peyrin. Super-sbox cryptanalysis: Improved attacks for AES-like permutations. In Seokhie Hong and Tetsu Iwata, editors, *Fast Software Encryption – FSE 2010*, volume 6147 of *Lecture Notes in Computer Science*, pages 365–383. Springer, Heidelberg, February 2010.

[HM97] Carlo Harpes and James L. Massey. Partitioning cryptanalysis. In Eli Biham, editor, *Fast Software Encryption – FSE'97*, volume 1267 of *Lecture Notes in Computer Science*, pages 13–27. Springer, Heidelberg, January 1997.

[Hou06] Xiang-dong Hou. Affinity of permutations of $p_2{}^n$. *Discret. Appl. Math.*, 154(2):313–325, 2006.

[HPS22] Sartaj Ul Hasan, Mohit Pal, and Pantelimon Stănică. The c-differential uniformity and boomerang uniformity of two classes of permutation polynomials. *IEEE Transactions on Information Theory*, 68(1):679–691, 2022.

[KN10] Dmitry Khovratovich and Ivica Nikolic. Rotational cryptanalysis of ARX. In Seokhie Hong and Tetsu Iwata, editors, *Fast Software Encryption – FSE 2010*, volume 6147 of *Lecture Notes in Computer Science*, pages 333–346. Springer, Heidelberg, February 2010.

[KS07] Liam Keliher and Jiayuan Sui. Exact maximum expected differential and linear probability for two-round advanced encryption standard. *IET Inf. Secur.*, 1(2):53–57, 2007.

[LMR15] Gregor Leander, Brice Minaud, and Sondre Rønjom. A generic approach to invariant subspace attacks: Cryptanalysis of robin, iSCREAM and Zorro. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology –*

EUROCRYPT 2015, Part I, volume 9056 of *Lecture Notes in Computer Science*, pages 254–283. Springer, Heidelberg, April 2015.

[LW17] Li Lin and Wenling Wu. Meet-in-the-middle attacks on reduced-round Midori64. *IACR Transactions on Symmetric Cryptology*, 2017(1):215–239, 2017.

[Min16] Brice Minaud. *Analyse de primitives cryptographiques récentes*. PhD thesis, Université de Rennes 1, 2016.

[MRS+21] Sihem Mesnager, Constanza Riera, Pantelimon Stănică, Haode Yan, and Zhengchun Zhou. Investigations on c-(almost) perfect nonlinear functions. *IEEE Transactions on Information Theory*, 67(10):6916–6925, 2021.

[Nyb94] Kaisa Nyberg. Differentially uniform mappings for cryptography. In Tor Helleseth, editor, *Advances in Cryptology – EUROCRYPT'93*, volume 765 of *Lecture Notes in Computer Science*, pages 55–64. Springer, Heidelberg, May 1994.

[PT22] Thomas Peyrin and Quan Quan Tan. Mind your path: On (key) dependencies in differential characteristics. *IACR Transactions on Symmetric Cryptology*, 2022(4):179–207, 2022.

[RP20] Adrián Ranea and Bart Preneel. On self-equivalence encodings in white-box implementations. In Orr Dunkelman, Michael J. Jacobson Jr., and Colin O'Flynn, editors, *SAC 2020: 27th Annual International Workshop on Selected Areas in Cryptography*, volume 12804 of *Lecture Notes in Computer Science*, pages 639–669. Springer, Heidelberg, October 2020.

[SGG+22] Pantelimon Stănică, Sugata Gangopadhyay, Aaron Geary, Constanza Riera, and Anton Tkachenko. C-differential bent functions and perfect nonlinearity. *Discrete Applied Mathematics*, 307:160–171, 2022.

[The23] The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 9.8)*, 2023. https://www.sagemath.org.

[TLS19] Yosuke Todo, Gregor Leander, and Yu Sasaki. Nonlinear invariant attack: Practical attack on full SCREAM, iSCREAM, and Midori64. *Journal of Cryptology*, 32(4):1383–1422, October 2019.

[Wag04] David Wagner. Towards a unifying view of block cipher cryptanalysis. In Bimal K. Roy and Willi Meier, editors, *Fast Software Encryption – FSE 2004*, volume 3017 of *Lecture Notes in Computer Science*, pages 16–33. Springer, Heidelberg, February 2004.

A Connexions Between Commutative Cryptanalysis and Conjugation

A.1 Non-linear Changes of Variables

Let $F\colon \mathbb{F} \to \mathbb{F}$ be a function and $G\colon \mathbb{F} \to \mathbb{F}$ be a permutation. We recall that, by the *conjugate of F by G*, we mean $G \circ F \circ G^{-1}$ and we denote it F^G. Let $\alpha = (\alpha_n, \cdots, \alpha_1) \in \mathbb{F}_2^n$ and $F\colon \mathbb{F}_2^n \to \mathbb{F}_2^n$, where its coordinates are denoted $F = (f_n, \cdots, f_1)$. By the *α-component of F* we mean $\alpha \cdot F := \sum_{i=1}^{n} \alpha_i f_i$.

In [BCL18], Beierle, Canteaut & Leander investigated the linear cryptanalysis of non-linear conjugates of Midori. They showed that the existence of a balanced-non linear

invariant[11] of a function $F\colon \mathbb{F}_2^n \to \mathbb{F}_2^n$ could be interpreted as (and is actually equivalent) to the existence of a linear approximation $\alpha \to \alpha$ with absolute correlation 1 for a conjugate $G \circ F \circ G^{-1}$ of F, where G is non-linear. They then used this insight to reinterpret previous distinguishers from the literature [TLS19]: the only constraint for such G, α, g to exist is that $\alpha \cdot G = g$. It is pointed out in [BCL18], and later adressed by Beyne [Bey21], that $G \circ F_K \circ G^{-1}$ may also have other highly-undesirable properties, such as almost-one linear approximations for some weak keys.

If previous works have thoroughly studied the linear case, the differential one has been (to the best of our knowledge) left out until now. As we show in the next section, the differential properties of some particular conjugates of a cipher can be expressed in the commutative framework. The experimental results that follow, while being a little bit more general, assess the soundness of such a relationship between commutation and conjugation.

A.2 Relationship with Commutative Cryptanalysis.

For a permutation $G : \mathbb{F} \to \mathbb{F}$, we have that $F^G(x+\alpha) + F^G(x) = \beta$ if and only if

$$G \circ F \circ G^{-1}(x+\alpha) = G \circ F \circ G^{-1}(x) + \beta \ .$$

If we let $y = G^{-1}(x)$, this is equivalent to $G \circ F \circ G^{-1}\big(G(y)+\alpha\big) = G \circ F(y) + \beta$, which has as many solutions as $F \circ G^{-1}\big(G(y)+\alpha\big) = G^{-1}\big(G \circ F(y) + \beta\big)$. We rewrite it as:

$$F \circ T_\alpha^{(G^{-1})} = T_\beta^{(G^{-1})} \circ F. \tag{4}$$

Under the assumption that G is a permutation such that $T_\alpha^{(G^{-1})}$ and $T_\beta^{(G^{-1})}$ are affine, we thus obtain:

$$F^G(x+\alpha) + F^G(x) = \beta \quad \iff \quad F \circ A(x) = B \circ F(x), \tag{5}$$

where $A := T_\alpha^{(G^{-1})}, B := T_\beta^{(G^{-1})}$. In particular, $\alpha \to \beta$ holds with probability one through F^G if and only if A and B commute through F. A particular case of the situation we just described actually happens for Vert, as we will show below.

A.3 Conjugating Vert.

A.3.1 The conjugation framework

Let us consider a composition $F = R_r \circ ... \circ R_1$, where all functions R_i map $\mathbb{F}$ to itself. A conjugate of F can be obtained by composing conjugates of its subfunctions for the same permutation. Indeed, by interleaving $G^{-1} \circ G$ within the computation of F^G, we can rewrite $F^G = G \circ R_r \circ R_{r-1} \circ ... \circ R_2 \circ R_1 \circ G^{-1}$ into

$$F^G \ = \ \underbrace{G \circ R_r \circ G^{-1}}_{R_r^G} \circ \underbrace{G \circ R_{r-1} \circ G^{-1}}_{R_{r-1}^G} \circ G \circ ... \circ R_2 \circ G^{-1} \circ \underbrace{G \circ R_1 \circ G^{-1}}_{R_1^G} \ .$$

We investigate the 4 main operations in Midori (and Vert). First, a permutation G applied in parallel at the S-box level commutes with the cells permutation, be it ShuffleCell or ShiftRows. Thus, if we denote $\mathcal{P}$ a cells permutation, we have that $\mathcal{G} \circ \mathcal{P} \circ \mathcal{G}^{-1} = \mathcal{P}$.

Recall that the MixColumn layer $\mathcal{M}$ consists of the parallel application of M over the 4 four columns of the state. Thus, it does not provide any intra-nibble mixing and simplifications can be hoped to occur in $\mathcal{G} \circ \mathcal{M} \circ \mathcal{G}^{-1}$ provided that G (and thus $\mathcal{G}$) is sparse enough—which is indeed the case, as we will see.

[11] A balanced non-linear invariant is a balanced Boolean function $g : \mathbb{F} \to \mathbb{F}_2$ such that $g(x) = g \circ F(x)$ for all x.

Usually ignored[12] in statistical attacks, the key addition plays an important role here. Indeed, studying the differential behavior through T_c^G is not as simple as through a standard key addition because the key dependency within it can be non-linear! In order to handle transitions through T_c^G, it is important to keep it as simple as possible. Finally, regarding the S-box layer, we can restrict our investigation to the nibble level, which allows us to brute-force a rather large space of candidates.

A.3.2 Our space of conjugates.

Given the analysis sketched above and the necessity of G to be simple and sparse enough, we investigate the conjugates of the S-box S of Midori through change of variables containing a single quadratic coordinate, as they are the simplest non-linear change of variables one can think of. Because a balanced quadratic Boolean function in m variables is linearly-equivalent to a function of the form $g(x_{m-1}, \cdots, x_2) + x_1$ (see Propositions 55 and 28 of [Car20]) we chose at first to study Feistel-like permutations (in fact, involutions) of the form:

$$G_g(x_m, ..., x_1) := \big(x_m, x_{m-1}, \cdots, x_1 + g(x_m, \cdots, x_2)\big) \ ,$$

where g is a quadratic Boolean function. This however induces a restriction in our search space: only the 1-component can be non-linear. To solve this, we compose our change of variables with a linear permutation which enables to move the 1-component into the a-component, for any value of $a \in \mathbb{F}_2^n$. The only constraint for such a linear permutation L_a is that $L_a(1) = a$. We thus look at conjugates of the form $G_g \circ L_a^{-1} \circ F \circ L_a \circ G_g$, where, in our case, we deterministically built L_a starting from a chosen a: once the image of 1 is fixed, only the images of $2, 4, 8$ need to be chosen (as $(1, 2, 4, 8)$ is the standard basis of $\mathbb{F}_2^n \simeq [0, \cdots, 2^n - 1]$ and L_a is linear). So starting from 2 to 8, we selected as image the smallest integer such that the rank of the partial list of images is increased by one, until obtaining the images for the full basis.

All in all, we focused on the class of permutations $G_{a,g} := G_g \circ L_a^{-1}$. Regarding the practical search, this space is sufficiently constrained to be efficiently explored in practice as it consists of $2^m - 1$ choices for a multiplied by $2^{2^{m-1}}$ Boolean functions mapping $m-1$ bits to 1 bit, meaning about 2^{12} possibilities (in our case $m = 4$).

Yet being quite small, this class contains interesting conjugates. Indeed, when used in parallel, such sparse permutations yield simple conjugates for M. Furthermore, the conjugates of constant additions have the intended very simple shape:

$$T_c^{G_{a,g}^{-1}} \ = \ G_{a,g}^{-1} \circ T_c \circ G_{a,g} = L_a \circ G_g \circ T_c \circ G_g \circ L_a^{-1}. \tag{6}$$

We further see that

$$\begin{aligned} G_g \circ T_c \circ G_g(x) &= G_g\big(x_m + c_m, \cdots, x_1 + c_1 + g(x_m, \cdots, x_2)\big) \\ &= \big(x_m + c_m, \cdots, x_1 + c_1 + g(x_m, \cdots, x_2) + g(x_m + c_m, \ldots, x_2 + c_2)\big) \\ &= T_c \circ G_{\Delta_{c'} g}(x), \end{aligned} \tag{7}$$

where $c' = (c_m, \cdots, c_2)$ and $\Delta_{c'} g$ denotes the (first-order) derivative of g toward c'. Finally, we obtain

$$T_c^{G_{a,g}^{-1}} \ = \ L_a \circ T_c \circ G_{\Delta_{c'} g} \circ L_a^{-1},$$

and as g is quadratic, we observe that $G_{\Delta_{c'} g}$ and thus $T_c^{G_{a,g}^{-1}}$ are affine. This corresponds to the assumption we gave for Equation 5 to hold; it will be needed for the commutative interpretation in Supplementary Material A.3.4.

[12] A crucial property of statistical attacks is that their behaviour is usually assumed to be key-independent, which allows the construction of key-independent distinguishers that are then used in key-recovery attacks. However, this folklore wisdom has been challenged recently as a string of papers [BR22, AK19, PT22] points out strong key dependencies e.g. in differential attacks.

In the same way we observe that, $T_c^{G_{a,g}} = G_g \circ L_a^{-1} \circ T_c \circ L_a \circ G_g$, but as $L_a^{-1} \circ T_c \circ L_a(x) = L_a^{-1}(L_a(x) + c) = x + L_a^{-1}(c) = T_{L_a^{-1}(c)}(x)$, we can deduce from Equation 7 that $T_c^{G_{a,g}} = G_g \circ T_{L_a^{-1}(c)} \circ G_g$ is also affine. This corresponds to our desire to keep the conjugate of key addition as simple as possible to enable an easier study of the key dependency.

Let us now present a differential study of a *conjugate* cipher, built from such permutations $G_{a,g}$.

A.3.3 Probability-One Differential Trails for Conjugates.

Within the class of permutations $G_{a,g}$ presented above, we found that many of them induce very weak conjugates of the S-box S of Midori. More precisely, we identified permutations G such that S^G has a $\alpha \to \alpha$ probability-one differential. For example, we can consider $G_{a,g}$ where $g(x_4, x_3, x_2) = x_2 + x_2x_4$ and $a = 5$. For easier verifications, look-up tables of the change of variables and of the conjugate are given below:

$$G_{5,x_2+x_2x_4} = \{\texttt{0x0}, \texttt{0x3}, \texttt{0x4}, \texttt{0x7}, \texttt{0x2}, \texttt{0x1}, \texttt{0x6}, \texttt{0x5}, \texttt{0x8}, \texttt{0xa}, \texttt{0xc}, \texttt{0xe}, \texttt{0xb}, \texttt{0x9}, \texttt{0xf}, \texttt{0xd}\} ,$$

$$S^{G_{5,x_2+x_2x_4}} = \{\texttt{0xb}, \texttt{0xe}, \texttt{0xf}, \texttt{0xc}, \texttt{0x9}, \texttt{0x5}, \texttt{0xd}, \texttt{0x7}, \texttt{0x8}, \texttt{0x4}, \texttt{0xa}, \texttt{0x0}, \texttt{0x3}, \texttt{0x6}, \texttt{0x1}, \texttt{0x2}\} .$$

From the remainder of this section, we denote $G = G_{5,x_2+x_2x_4}$ for lighter notation. We also denote $S' := S^G$.

First, we can easily observe that $\texttt{0xd} \to \texttt{0xd}$ holds with probability one through S' (for instance $S'(\texttt{0x0}) + S'(\texttt{0xd}) = \texttt{0xb} + \texttt{0x6} = \texttt{0xd}$ and so on). It immediately yields a probability-one $\gamma \to \gamma$ transition through the conjugate of the S-box layer $\mathcal{S}' := S'^{\times 16}$, where $\gamma := \texttt{0xdddddddddddddddd}$. Furthermore, we observe[13] that $\texttt{0xdddd} \to \texttt{0xdddd}$ holds with probability one through $M^{G^{\times 4}}$, which ultimately leads to the probability-one transition $\gamma \to \gamma$ through $\mathcal{M}^{\mathcal{G}}$, the conjugate of the full MixColumn layer.

Perhaps counter-intuitively, while this differential goes through both the S-box and the linear layer with probability one, its interaction with the key addition is more sophisticated. As expected with such a choice for $G = G_{a,g}$ (the Boolean function g is a quadratic function), T_k^G is affine, so once a key is fixed, the derivative in any direction is constant. This means that, depending on the key k, any transition $\alpha \to \beta$ either holds with probability 0 or 1 through T_k^G. We can thus easily establish the set V of key nibbles for which $\alpha \to \beta$ holds with probability one, by looking at the equation for $x = 0$, namely:

$$\begin{aligned} V &= \{k \in \mathbb{F}_2^4 \mid \Delta_\alpha T_k^G(x) = \beta \ \forall\ x \in \mathbb{F}_2^4\} \\ &= \{k \in \mathbb{F}_2^4 \mid \Delta_\alpha T_k^G(0) = \beta\}. \end{aligned}$$

As $T_k^G = T_{L_a^{-1}(k)}^{G_g}$, we first look at $\Delta_\alpha T_c^{G_g}(0)$. Using Equation 7, we get that the i-th coordinate of $\Delta_\alpha T_c^{G_g}(0) = T_c^{G_g}(0) + T_c^{G_g}(\alpha)$ for $i \in \{2, \cdots, m\}$, is equal to $c_i + (c_i + \alpha_i) = \alpha_i$, while the first one is equal to $\alpha_1 + g(0) + g(c') + g(\alpha') + g(c' + \alpha')$, where c', α' correspond to c and α where the first coordinate is omitted. Replacing c by $L_a^{-1}(k_4, k_3, k_2, k_1) = (k_4, k_2, k_1 + k_3, k_3)$, and using that $g(x) = x_2 + x_2x_4$, we thus obtain that

$$\Delta_\alpha T_k^G(0) = \begin{pmatrix} \alpha_4 \\ \alpha_3 \\ \alpha_2 \\ \alpha_1 + k_4\alpha_2 + (k_1 + k_3)\alpha_4 \end{pmatrix} \tag{8}$$

[13] A small Sage script is provided as supplementary material. It presents the basic propreties of $G_{x_2+x_2x_4,5}$ and its interaction with Vert.

Finally, we are able to determine the set of keys V for which $\texttt{0xd} \to \texttt{0xd}$ holds with probability one through T_k^G, by solving

$$\begin{aligned} 1 &= 1 \\ 1 &= 1 \\ 0 &= 0 \\ 1 + k_4 \times 0 + (k_1 + k_3) &= 1. \end{aligned}$$

We easily obtain $V = \{k \in \mathbb{F}_2^4, k_1 + k_3 = 0\} = \langle \texttt{0x2, 0x5, 0x8} \rangle$.

To conclude, provided that the round key lies in V^{16}, there exists a probability-one differential $\gamma \to \gamma$ for the *conjugate* of one round of Vert^2_{SC} or Vert^2_{SR}. This is true for any number of rounds if both K_0 and K_1 are in V^{16}. It is also true for any variant of Vert where each constant nibble is taken in $\langle \texttt{0x2}, \texttt{0x5}, \texttt{0x8} \rangle$, *i.e.* each nibble can take 8 of the 16 values.

Remark 2. This does not apply to the genuine Midori because $1 \notin V$.

Any such change of variables G (resp. conjugate cipher) can be used as distinguisher: given an oracle access to Vert^2_{SC} or Vert^2_{SR}, it is sufficient to choose, $p_1 = G^{-1}(p)$, $p_2 = G^{-1}(p + \gamma)$, to ask for the corresponding ciphertexts c_1, c_2 and to verify whether $G(c_1)$ is equal to $G(c_2) + \gamma$.

More importantly, as we see in the next section, the differential behaviour of this conjugate of Vert is best explained by commutative trails.

A.3.4 Commutative Interpretation

The differential behaviour of the non-linear conjugate of Vert we exhibited here can be explained by the iterative commutative trail described in Section 6.1.1. Indeed, as mentioned in Supplementary Material A.2, Equation 5 links the existence of a probability-one differential $\alpha \to \beta$ for F^G to the existence of an affine self-similarity behavior for F. However, this holds *under the assumption* that $T_\alpha^{G^{-1}}$ and $T_\beta^{G^{-1}}$ are affine.

In the case of $G_{a,g}$, where g is quadratic, we already proved in the last section that $T_c^{G_{a,g}^{-1}}$ was indeed affine.

Let us look at the case where $a = 5$ and $g = x_2 + x_2x_4$ more carefully. In that case, from the design of L_a, we know that L_a verifies $L_a(1) = 5, L_a(2) = 1, L_a(4) = 2, L_a(8) = 8$. We thus deduce the ANF of L_a[14]:

$$L_a(x) = \begin{pmatrix} x_4 \\ x_1 \\ x_3 \\ x_1 + x_2 \end{pmatrix}.$$

From the ANF of L_a, L_a^{-1} and g, we can deduce the ANF of $T_k^{G_{a,g}^{-1}}$.

$$T_k^{{G_{a,g}}^{-1}}(x) = \begin{pmatrix} x_4 + k_4 \\ x_1k_4 + x_3k_4 + x_3 + x_4k_2 + k_1 + k_2k_4 + k_2 \\ x_2 + k_3 \\ x_1k_4 + x_1 + x_3k_4 + x_4k_2 + k_1 + k_2k_4 \end{pmatrix}$$

By substituting $k = \texttt{0xd}$, we thus get $T_{\texttt{0xd}}^{{G_{a,g}}^{-1}}(x) = (x_4 + 1, x_1 + 1, x_2 + 1, x_3 + 1)$. As a matter of fact, we can observe that $T_{\texttt{0xd}}^{{G_{a,g}}^{-1}}$ coincides with the definition of A_1 given in Section 5. The differential behavior of this non-linear conjugate of Vert can thus be equivalently explained by the commutative framework.

[14] We can also verify that it matches the ANF of L_a^{-1} used earlier.

B The Toy Cipher Grün and its Probability-one Trail

In this section, we present Midori128, the 128-bit state version of Midori, the toy cipher Grün which is based on it, as well as a probability-one self-similarity distinguisher for Grün based on the same commutative cryptanalysis techniques that we presented in Section 6.1.

B.1 Midori128

The 128-bit-state Midori128 operates on a 4×4 square state of bytes. The S-box layer uses four distinct involutive 8-bit S-boxes: one of them is applied on each byte. These four 8-bit S-boxes are built by linearly conjugating a parallel call to a single 4-bit S-box. We denote them $SSb_i := SS^{L_i}$. They are depicted in Figure 6.

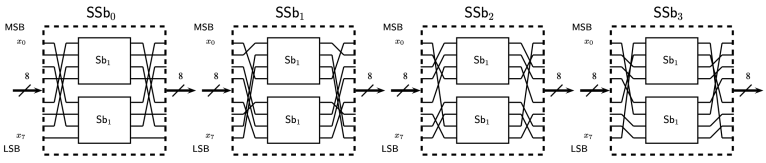

Figure 6: Midori128 S-boxes, extracted from [BBI+15].

The other layers (MixColumn and ShuffleCell) are identical to the ones of Midori64. Finally Midori128 uses K as a round key for every round. Sparse round constants belonging to $\{\texttt{0x0}, \texttt{0x1}\}^{16}$ are also added at each round.

B.2 Grün

Let us recall that Grün is a modified-constants version of Midori128. It is identical to Midori128 except for the constants that lie in $\{\texttt{0x00}, \texttt{0x11}\}$ rather than in the genuine $\{\texttt{0x00}, \texttt{0x01}\}$.

B.3 A Third Example of Probability-one Commutative Trail

Let us apply the same methodology as in Section 6.1 to the case of Grün. As we can see in Figure 6, each of the four S-boxes used in Grün has an internal symmetry: swapping the two nibbles in the input amounts to swapping the two nibbles in the output. The four of them thus commutes with the linear map $A_5 \colon \mathbb{F}_2^4 \times \mathbb{F}_2^4 \to \mathbb{F}_2^4 \times \mathbb{F}_2^4, (x, y) \to (y, x)$ and the full S-box layer thus commutes with $\mathcal{A}_5$. Then, as already explained, Condition 1 of Theorem 1 is immediately verified for Midori MixColumn. So is Condition 2, because here, A_5 is linear (i.e. $c_{A_5} = 0$). Thus, $\mathcal{A}_5 \circ \mathcal{M} = \mathcal{M} \circ \mathcal{A}_5$ holds. The ShuffleCell layer naturally commutes with $\mathcal{A}_5$. Finally, we observe that $\mathrm{Fix}(L_{A_5}) = \{(x, x)\} \subset (\mathbb{F}_2^4)^2 \simeq \mathbb{F}_2^8$. Because the constants of Grün lie in $\mathrm{Fix}(L_{A_5})$, we obtain from the 16 4-bit conditions, a set of $2^{128-4\times16} = 2^{64}$ weak keys.

C A Complementary Experiment

Experiment 3. We studied more precisely the peak of the strongest keys among the weak ones. To do so, we repeated Experiment 2, but this time, we fixed the round number to $r = 4$ and let the size of $\mathbb{X}$ grow.

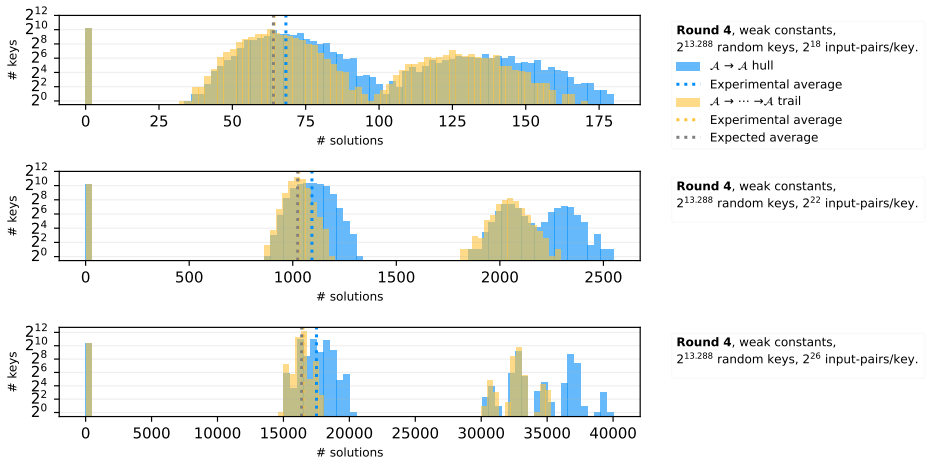

Figure 7: Fixed-key study for 4-round version: Evolution of the distribution of the numbers of $\left(x, \widetilde{\mathcal{A}}(x)\right)$ pairs following the trail/hull, as $\mathbb{X}$ grows.

Interpretation. The idea behind such an experiment was to observe whether a Gaussian bell could be hidden behind the peak for 0 solution: a key for which the actual number of solutions is very low, could appear as a key with no solution because of a lack of data. According to Figure 7, the peak stays the same as data increase. Either the peak does not collapse, or still, more data is needed. However, for the other weak keys, a multitude of sub-classes appear among the Gaussian bells, as data increases.