IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2023, No. 4, pp. 270–298. DOI:10.46586/tosc.v2023.i4.270-298

Propagation of Subspaces in Primitives with Monomial Sboxes: Applications to Rescue and Variants of the AES

Aurélien Boeuf, Anne Canteaut and Léo Perrin

Inria, Paris, France
{aurelien.boeuf,anne.canteaut,leo.perrin}@inria.fr

Abstract. Motivated by progress in the field of zero-knowledge proofs, so-called *Arithmetization-Oriented (AO)* symmetric primitives have started to appear in the literature, such as MiMC, Poseidon or Rescue. Due to the design constraints implied by this setting, these algorithms are defined using simple operations over large (possibly prime) fields. In particular, many rely on simple low-degree monomials for their non-linear layers, essentially using $x \mapsto x^3$ as an S-box.

In this paper, we show that the structure of the material injected in each round (be it subkeys in a block cipher or round constants in a public permutation) could allow a specific pattern, whereby a well-defined affine space is mapped to another by the round function, and then to another, etc. Such chains of one-dimensional subspaces always exist over 2 rounds, and they can be extended to an arbitrary number of rounds, for any linear layer, provided that the round-constants are well chosen.

As a consequence, for several ciphers like Rescue, or a variant of AES with a monomial Sbox, there exist some round-key sequences for which the cipher has an abnormally high differential uniformity, exceeding the size of the Sbox alphabet.

Well-known security arguments, in particular based on the wide-trail strategy, have been reused in the AO setting by many designers. Unfortunately, our results show that such a traditional study may not be sufficient to guarantee security. To illustrate this, we present two new primitives (the tweakable block cipher Snare and the permutation-based hash function Stir) that are built using state-of-the-art security arguments, but which are actually deeply flawed. Indeed, the key schedule of Snare ensures the presence of a subspace chain that significantly simplifies an algebraic attack against it, and the round constants of Stir force the presence of a subspace chain aligned with the rate and capacity of the permutation. This in turns implies the existence of many easy-to-find solutions to the so-called CICO problem.

Keywords: Arithmetization-Oriented Primitives · Rescue · Invariant subspace · Differential uniformity

1 Introduction

The emergence of cryptographic protocols with advanced functionalities, such as fully homomorphic encryption, multi-party computation and new types of proof systems, is asking for new symmetric primitives offering better performances than the standard algorithms in the context of these specific applications. Most notably, the use of primitives that are defined over finite fields of odd characteristic, in particular over large prime fields, is desirable in many such applications. For example, STARKs [BBHR18] are most efficient over fields $\mathbb{F}_p$ such that $p-1$ is a multiple of a large power of 2, while other schemes rely on pairing-friendly elliptic curves and need primitives defined over the corresponding

Received: 2023-09-01 Accepted: 2023-11-01 Published: 2023-12-08

fields [GW20]. Furthermore, in the context of side-channel resistance, primitives operating on a prime field have been shown to be well-suited for efficient masking [MMMS23].

Several new ciphers and hash functions operating over fields of odd characteristic have then been proposed recently, including MiMC [AGR+16] and its variants [AGP+19], Poseidon [GKR+21], Ciminion [DGGK21], ReinforcedConcrete [GKL+22], Griffin [GHR+23], AES-Prime [MMMS23], Anemoi [BBC+23], or the *arithmetization-oriented* functions from the Marvellous family, especially Rescue [AAB+20] and its different versions [SAD20, AKM+22].

However, while there are many new designs, this area suffers from a lack of cryptanalysis. Some design principles governing the choice of the Sbox or of the linear layer in an SPN can certainly be directly transposed to any finite field, but some properties inherent to arithmetization-orientation may introduce new weaknesses in the primitives:

- While this might look like a minor change at first glance, the size and the characteristic of the underlying field may strongly affect the security, as was exhibited in [BCD+20]. For instance, it was shown that the complexities of integral attacks depend on the sizes of multiplicative subgroups of the involved finite field [EGL+20, BCP23, LAW+23].
- While the number of rounds of these primitives is usually decided by the complexity of attacks based on solving polynomial systems, estimating this complexity is harder than it looks. Indeed, it is tied to the specific model used to encode the evaluation of the primitive as an equation, or as a system of multi-variate equations, and this model is not unique. It has been shown for instance in [ACG+19] and [BBLP22], that some of these primitives were much more vulnerable to such attacks than anticipated because of a clever re-writing of the involved equations.

An in-depth security evaluation of these primitives and of their design principles is therefore necessary to achieve the same level of confidence as the one we have in primitives defined over binary fields, after decades of research in cryptanalysis. As we are now faced with several proposals of arithmetization-oriented primitives, there is a clear need to analyze their respective security levels and potential weaknesses.

This need is all the more crucial as arithmetization-oriented primitives are defined for a very broad number of different parameters. While there are only 3 versions of the AES, most arithmetization-oriented primitives are defined for any prime p, and for any block size (i.e. they operate on $\mathbb{F}_p^m$, where m can be as low as 2 or higher than 10). Even for a given set of parameters, authors sometimes give implementers the freedom to choose better components for their use case, e.g. to use an MDS matrix that is more efficient according to some metrics. In fact, the authors of Rescue explicitly stated in [AKM+22, Section 4.1] that "Rescue-Prime is secure when instantiated with *any* MDS matrix". This substantially increases the attack surface, and one can wonder whether all these choices provide the same security. For example, it was found in [BCD+20] that some matrices could be weak if used in Poseidon, a direction much more thoroughly explored in [KR21].

In this context, our work investigates the potential weaknesses coming from the use of monomial mappings, $x \mapsto x^\alpha$ over a finite field, as Sboxes. Many primitives, like MiMC or Rescue, use as Sboxes "pure" monomial functions because they operate on a prime field. This differs from the AES Sbox [AES01], which consists of a monomial mapping, i.e., $x \mapsto x^\alpha$ over $\mathbb{F}_{2^8}$, composed with an $\mathbb{F}_2$-linear affine function which has very little influence on the resistance to statistical attacks (see e.g. [CR15]), but thwarts potential attacks exploiting a simple algebraic representation of the Sbox [DR02]. However, when the Sbox operates on a prime field, this simple method cannot be used anymore to make the univariate polynomial representation of the Sbox more complex.

Our contributions. Our study therefore focuses on *Monomial-Based SPNs* (shortened into *monomial-based SPNs*), i.e., families of permutations whose round function follows

the SPN construction with a monomial Sbox over $\mathbb{F}_q$ and a linear layer defined by a matrix with entries in the same field. This includes, for example, Rescue and its variants [AAB+20, SAD20, AKM+22], but also a variant of the AES where the $\mathbb{F}_2$-linear affine function composed with the inverse mapping in $\mathbb{F}_{2^8}$ is removed. We show that, for any such primitive, there exist some round-constants (or round-keys) such that, at the end of each round, the image of an $\mathbb{F}_q$-affine subspace of dimension 1 is still an affine subspace. Even if this particular property involves a subspace of dimension 1 only, it is worth noting that, in the context of arithmetization-oriented primitives, the Sbox alphabet is large, typically of size 2^{32} or more, which implies that the property affects a large number of inputs.

Moreover, for some of these round-constants, any even number of rounds of the primitive is an affine function on this affine subspace, in the particular cases where the Sbox is an involution (e.g. the inversion in $\mathbb{F}_q$) or where two consecutive rounds of the primitive use Sboxes which are the inverse of each other, like in Rescue, where the Sbox in all even rounds corresponds to $x \mapsto x^3$ and the Sbox in all odd rounds to its inverse.

The proportion of such weak round-constants (resp. round-keys) is very small. This implies that the property does not provide an attack on real instances of the primitive, when the round-constants are randomly chosen or when the round-keys are derived from a well-defined key-schedule. However, it points out that the security level offered by this type of primitives highly depends on the choice of the round-keys, which appears to be very problematic since their security mainly relies on arguments derived from analyses on average over all round-key sequences. As an illustration of this, we exhibit some examples where the fixed-key maximal differential probability of the permutation (aka differential uniformity) is much higher than expected from the maximal expected differential probability. This result generalizes and explains the observations made in [BCL+20] on Rescue: for instance, [BCL+20] enlightens "the oddity of the behaviour" of an instance of the cipher whose differential uniformity significantly increases between two rounds.

The existence of weak round-keys usually does not threaten the security of well-specified primitives. However, it raises a worrying issue since weak round-constants could be chosen intentionally, in order to insert a backdoor in the primitive. Indeed, we exhibit two such examples, a hash function and a tweakable block cipher, in which a hidden backdoor makes some attack feasible. For the hash function, we target its inner permutation and ensure the existence of solutions for the so-called CICO problem that are trivial to find for the malicious designer. The tweakable block cipher illustrates that the existence of chains of affine subspaces over the primitive can be used within the Malicious framework [PW20] to introduce a backdoor which greatly reduces the complexity of key recovery. Our aim with these algorithms is to show the importance of *substantially* restricting the space in which implementers can choose primitive variants.

Organization of the paper. The following section defines the general structure of monomial-based SPNs, and provides some examples. Section 3 analyzes the propagation of affine subspaces through such permutations, with a particular focus on subspaces of dimension 1. This property implies that, when two consecutive rounds of the primitive use Sboxes which are the inverse of each other, there always exist some round-constants for which the differential uniformity is very high: Section 4 exhibits examples of such instances of a variant of the AES whose differential uniformity behaves in an unexpected way. Finally, Section 5 shows how the existence of such a chain of affine subspaces can be used to insert a backdoor in a symmetric primitive, and describes a permutation (Stir) that can be used to build a backdoored hash function, and a backdoored tweakable block cipher (Snare).

2 Monomial-Based SPNs

In this paper, we focus on families of permutations which follow the SPN construction.[1] These permutations can be used in different settings: for instance, together with a key schedule, as block ciphers, or within the sponge construction as hash functions. In this second case, the data injected during each round corresponds to public round-constants, instead of secret round-keys. These permutations operate on $\mathbb{F}_q^m$ where q is a prime power and m is the number of field elements (*i.e.*, the number of blocks) in the internal state. In the particular context of arithmetization-oriented primitives, typical values for q which are implementation-friendly are $q = 2^k$ or $q = p$ where $p \approx 2^k$, and $k \geq 32$ (and can be as high as $k = 256$ in some cases), see e.g. [AAB+20]. Our work then focuses on SPNs, called monomial-based SPNs, whose round functions are defined as follows (see Figure 1).

Definition 1. (Monomial-Based SPN) Let q be a prime power and m a positive integer. Let α be an integer with $\gcd(q-1, \alpha) = 1$ and M be an $m \times m$ nonsingular matrix with coefficients in $\mathbb{F}_q$. For any round constant $\boldsymbol{r} \in \mathbb{F}_q^m$, $\mathcal{R}(q, m, \alpha, M, \boldsymbol{r})$ denotes the round function with round constant $\boldsymbol{r}$ of a *monomial-based substitution-permutation network (monomial-based SPN)* defined over $\mathbb{F}_q^m$ whose substitution function consists of the concatenation of m copies of $S : x \mapsto x^\alpha$ over $\mathbb{F}_q$ and whose linear layer corresponds to M (see Figure 1).

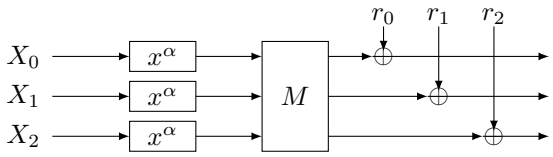


Figure 1: The transformation $\mathcal{R}(q, 3, \alpha, M, \boldsymbol{r})$.

We impose that $\gcd(q-1, \alpha) = 1$, to ensure that S is bijective. In arithmetization-oriented primitives, we usually have $\alpha = 3$ or $\alpha = 5$. In the whole paper, the non-linear permutation of $\mathbb{F}_q^m$ obtained by applying S on each coordinate in parallel is denoted $\mathcal{S}$.

The matrix that defines the linear layer is often chosen as an MDS matrix, i.e., (Id_m, M) generates a $[2m, m]$-MDS code over $\mathbb{F}_q$, in order to maximize the diffusion within the cipher. For the sake of simplicity, M will also refer to the linear function corresponding to the multiplication by this matrix.

Rescue. Rescue is a family of permutations proposed by Aly *et al.* [AAB+20]. Its Sbox is defined as a power mapping over a prime field, i.e., $q = p$ in the previous notation. One specificity of Rescue is that the Sbox layer $\mathcal{S}$ in every round with an even index[2] corresponds to the concatenation of m copies of S, while the Sbox layer in every odd round corresponds to its inverse $\mathcal{S}^{-1} = (S^{-1}, \ldots, S^{-1})$ where $S^{-1} : x \mapsto x^{1/\alpha}$. Here, a *round* is defined as a single application of the round function, *i.e.*, with a single Sbox layer ($\mathcal{S}$ or $\mathcal{S}^{-1}$). It is worth noting that, in some other works, a round corresponds to the successive application of two round functions, one with $\mathcal{S}$ and one with $\mathcal{S}^{-1}$.

More detailed specifications of a hash function named Rescue-Prime, based on this design, are presented in [SAD20]. Rescue-Prime follows the sponge construction and its core permutation, Rescue-XLIX, which corresponds to two successive rounds of Rescue, i.e. the composition of $\mathcal{R}(p, m, \alpha, M, \boldsymbol{r})$ and a similar transformation with an inverse[3]

[1] Here, *substitution-permutation* has to be understood without any restriction on the linear permutation.
[2] In the whole paper, the rounds are numbered starting from 0.
[3] This slightly differs from the specifications in [AAB+20] where of α and α^{-1} are swapped.

Sbox layer, $\mathcal{R}(p, m, \alpha^{-1}, M, \boldsymbol{r}')$, where p is a large prime and α is the smallest integer coprime with $(p-1)$. The MDS matrix M is derived from the row-echelon form of an $m \times 2m$ Vandermonde matrix. The sequence of round-constants is obtained by applying SHAKE-256 to a fixed ASCII string. More recently, a new variant of this hash function was described in [AKM+22], called "RESCUE-PRIME-OPTIMIZED". It differs from RESCUE-PRIME in subtle ways: the operations are reordered, and the MDS matrix uses a different structure to enable significant performance improvements in a specific context.

Modified AES, Vision and AES-Prime. A slightly modified round function of the AES can be written as $\mathcal{R}(2^8, 16, 254, M)$. The only difference between this and the AES round function is that, in the AES, the Sbox consists of the composition of the power mapping $x \mapsto x^{254}$ with an additional $\mathbb{F}_2$-affine function defined over the vector space $\mathbb{F}_2^8$.

The keyed permutation VISION has been proposed by Aly *et al.* [AAB+20] in the specific case of binary fields. Its round function has a similar structure as in the AES except that the linear layer M is MDS, while in the AES, it corresponds to an interleaving construction based on a smaller MDS matrix [ADK+14]. Exactly as in the AES, the inversion in $\mathbb{F}_{2^k}$ is followed by an additional $\mathbb{F}_2$-affine function.

Instead of using a power mapping over $\mathbb{F}_{2^k}$ as an Sbox, composing it with an $\mathbb{F}_2$-affine function seems to be a cheap safeguard to avoid a very simple representation over $\mathbb{F}_{2^k}$. However, this technique does not apply anymore when the Sbox operates on a prime field. The recently proposed variant of the AES over $\mathbb{F}_p$ with $p = 2^7 - 1$, named AES-PRIME [MMMS23], can also be seen as a monomial-based SPN. Indeed, it uses an Sbox of the form $x \mapsto x^\alpha + c$, and the constant addition can be seen as part of the key schedule. All these examples motivate the study of the round functions described in Definition 1.

3 Chains of Affine Subspaces

An affine space of dimension $d \leq m$ in $\mathbb{F}_q^m$ is defined by a basis, i.e. a family of d linearly independent vectors in $\mathbb{F}_q^m$, $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_d\}$, and by a single vector $\boldsymbol{a}$ in $\mathbb{F}_q^m$, called the *offset*. These $(d+1)$ vectors define the affine space V as follows:

$$V \;:=\; \boldsymbol{a} + \langle \boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_d \rangle \;=\; \{\boldsymbol{a} + x_1\boldsymbol{v}_1 + x_2\boldsymbol{v}_2 + ... + x_d\boldsymbol{v}_d : x_1, x_2, ..., x_d \in \mathbb{F}_q\}\ .$$

The goal of this section is to exhibit affine subspaces of a form such that their images by the round function are also affine spaces. Furthermore, in the case of a subspace of dimension 1, we show that, under some conditions on the round constants and the vectors defining the subspace, applying the round function again still yields an affine space. Step by step, it is possible to create chains of subspaces of arbitrary length which propagate through a cipher using such round constants.

"Chains" vs. "Trails". Our notion of subspace chains is related to the notion of invariant subspaces used in previous attacks, but differs on some key aspects. First, the subspaces we investigate are not invariant under the round function, but instead they are mapped to another subspace. In fact, invariant subspaces such as those in [GJN+16, LMR15] are particular cases of subspace chains. For this reason, the tools that have been previously developed for checking whether a given choice of the round constants guarantees that invariant subspace do not exist, like the algorithm presented in [BCLR17], cannot be applied in our context.

Another related notion is the notion of "subspace trail" defined in [GRR16] and analyzed in [LTW18]. But this is a stronger property than a subspace chain: it means that any coset of a given subspace is mapped to a coset of the output subspace, while we need this property for a single coset. We thus use the term "chain" rather than "trail" to avoid confusion.

Notation. In what follows, we always use the following notation. First, q is a prime power and m is an integer. Then, for any $\boldsymbol{x} \in \mathbb{F}_q^m$, its support $\mathrm{supp}(\boldsymbol{x})$ is the set of the indices of its nonzero coordinates, i.e., $\mathrm{supp}(\boldsymbol{x}) := \{i, 0 \leq i \leq m-1 : x_i \neq 0\}$. Its weight $\mathsf{wt}(\boldsymbol{x})$ is the size of $\mathrm{supp}(\boldsymbol{x})$.

3.1 Canonical Representation of Affine Subspaces

Let $V = \boldsymbol{a} + \langle \boldsymbol{v} \rangle$ be a subspace of dimension 1 in $\mathbb{F}_q^m$. Obviously, there are many pairs $(\boldsymbol{a}, \boldsymbol{v})$ which define the same subspace. In order to have a unique basis/offset representation of affine spaces of dimension 1, we define a *canonical representation* as follows.

- Since multiplying the basis vector by any nonzero scalar leaves the subspace invariant, the value of $\boldsymbol{v}$ on its first nonzero coordinate, i_S, is set to 1. Note that such an i_S always exists since $\boldsymbol{v} \neq \boldsymbol{0}$.
- The offset is then chosen such that it vanishes on this very same coordinate i_S. This is always possible because the offset can be defined as any $\boldsymbol{a}' = \boldsymbol{a} + \lambda \boldsymbol{v}$ with $\lambda \in \mathbb{F}_q$, and there is a unique vector $\boldsymbol{a}'$ whose coordinate in position i_S is 0, since $v_{i_S} \neq 0$.

For subspaces of higher dimension, the canonical representation is obtained by choosing the matrix formed by the basis vectors and the offset vector in reduced row-echelon form.

3.2 Separable Affine Subspaces

One of the key points conditioning the existence of chains of affine subspaces is the fact that an affine subspace is *separable* in the sense of the following definition.[4]

Definition 2 (Separable affine subspace)**.** An affine space V of dimension d in $\mathbb{F}_q^m$ with canonical representation $\boldsymbol{v}_0 + \langle \boldsymbol{v}_1, \ldots, \boldsymbol{v}_d \rangle$ is *separable* if

$$\mathrm{supp}(\boldsymbol{v}_i) \cap \mathrm{supp}(\boldsymbol{v}_j) = \emptyset, \ \forall i, j \in \{0, \ldots, d\}, i \neq j \, .$$

Note that the previous condition involves both the basis vector and the offset $\boldsymbol{v}_0$. The canonical representation is defined in such a way that, if V is separable for some representation, then it is separable for its canonical representation.

3.3 Multiplicative Sboxes

The following Sbox property plays a crucial role in subspace propagations.

Definition 3 (Multiplicative Sbox)**.** An Sbox $S : \mathbb{F}_q \to \mathbb{F}_q$ is said to be *multiplicative* if there exists a function $\pi : \mathbb{F}_q \to \mathbb{F}_q$ such that

$$\forall x, y \in \mathbb{F}_q, \ S(xy) = S(1)\pi(x)\pi(y) \, .$$

Considering the more general case where $S(xy) = S(1)\pi(x)\pi'(y)$ for two different functions π and π' would not be relevant. Indeed, π is defined up to a multiplicative constant since, for any $\lambda \in \mathbb{F}_q^*$, $(\lambda\pi, \lambda^{-1}\pi')$ satisfies the property too. Then, if we set $\pi(1) = 1$, we can easily deduce that $\pi = \pi'$ by using that $S(1 \times x) = S(x \times 1)$.

The following proposition shows that the multiplicative Sboxes over $\mathbb{F}_q$ correspond to power permutations (up to a multiplicative constant).

Proposition 1. *A bijective mapping $S : \mathbb{F}_q \to \mathbb{F}_q$ is multiplicative if and only if $S(x) = cx^k$ for some scalar $c \in \mathbb{F}_q$ and some integer k with* $\gcd(k, q-1) = 1$.

[4] Obviously, this notion has nothing to do with the notion of separable topological space.

Proof. Let $S(x) = cx^k$. For $\pi : x \mapsto x^k$, we have that $S(1) = c$, and that for all $x, y \in \mathbb{F}_q$, $S(xy) = cx^k y^k = S(1)\pi(x)\pi(y)$. Thus, these Sboxes are multiplicative.

We now prove that they are the only ones that are multiplicative. By definition of π, we have for all $x \in \mathbb{F}_q$ that $S(0) = S(0 \times x) = S(1)\pi(0)\pi(x)$ implying that $S(1)\pi(0) = S(0) = 0$ as π is bijective. Since S is bijective too, $S(1) \neq 0$, leading to $\pi(0) = 0$.

Let us now denote by α a primitive element in $\mathbb{F}_q$. Then, for any i, $0 \leq i \leq q-2$, we have

$$S(\alpha^{i+1}) = S(1)\pi(\alpha)\pi(\alpha^i) \text{ and } S(\alpha^i) = S(1)\pi(1)\pi(\alpha^i)$$

implying that

$$S(\alpha^{i+1}) = S(\alpha^i)\frac{\pi(\alpha)}{\pi(1)} .$$

Since $\pi(\alpha)$ cannot be zero, there exists some integer k such that

$$\frac{\pi(\alpha)}{\pi(1)} = \alpha^k .$$

We deduce that, for all i, $0 \leq i \leq q-2$,

$$S(\alpha^i) = S(1)(\alpha^k)^i ,$$

which equivalently means that, for all $x \in \mathbb{F}_q^*$, $S(x) = S(1)x^k$. Moreover, $\gcd(k, q-1) = 1$ since S is bijective. □

Therefore, in the rest of the paper, we will focus on the case where the Sbox is a power mapping (since the multiplicative constant does not have any cryptographic relevance).

3.4 Image of a Separable Affine Subspace by the Round Function

Theorem 1. *Let F be the round function $\mathcal{R}(q, m, \alpha, M, \boldsymbol{r})$ for any round constants. Let $\mathcal{S}$ denote its Sbox layer and $V = \boldsymbol{a} + \langle \boldsymbol{v}_1, ..., \boldsymbol{v}_d \rangle$ a separable subspace of $\mathbb{F}_q^m$, i.e., the supports of the $(d+1)$ vectors $\boldsymbol{a}$ and $\boldsymbol{v}_i$ are pairwise disjoint. Then, $F(V)$ is the affine subspace*

$$F(\boldsymbol{a}) + \langle M\mathcal{S}(\boldsymbol{v}_1), \ldots, M\mathcal{S}(\boldsymbol{v}_d) \rangle .$$

Moreover, for any $x_1, \ldots, x_d \in \mathbb{F}_q$, we have

$$F\left(\boldsymbol{a} + \sum_{i=1}^{d} x_i \boldsymbol{v}_i\right) = F(\boldsymbol{a}) + \sum_{i=1}^{d} S(x_i) \times M\mathcal{S}(\boldsymbol{v}_i) .$$

Proof. Let $\boldsymbol{x} = \boldsymbol{a} + \sum_{i=1}^{d} x_i \boldsymbol{v}_i$ be an element in V. Then,

$$\mathcal{S}(\boldsymbol{x}) = \left(S\left(a_0 + \sum_{i=1}^{d} x_i v_{i,0}\right), \ldots, S\left(a_{m-1} + \sum_{i=1}^{d} x_i v_{i,m-1}\right)\right) .$$

Since the supports of all vectors $\boldsymbol{a}$ and $\boldsymbol{v}_i$ are disjoint, only one term among the $(d+1)$ terms in the sum $a_j + \sum_{i=1}^{d} x_i v_{i,j}$ does not vanish. Therefore, for any coordinate j, we have

$$S\left(a_j + \sum_{i=1}^{d} x_i v_{i,j}\right) \;=\; S(a_j) + \sum_{i=1}^{d} S(x_i v_{i,j}) \;=\; S(a_j) + \sum_{i=1}^{d} S(x_i) S(v_{i,j}) ,$$

where the last equality comes from the fact that S is a power mapping (and thus a multiplicative function). It follows that

$$\mathcal{S}(\boldsymbol{x}) = \mathcal{S}(\boldsymbol{a}) + \sum_{i=1}^{d} S(x_i)\mathcal{S}(\boldsymbol{v}_i) .$$

Then, by linearity, we can directly deduce $M\mathcal{S}(\boldsymbol{x}) = M\mathcal{S}(\boldsymbol{a}) + \sum_{i=1}^{d} S(x_i) M\mathcal{S}(\boldsymbol{v}_i)$, to which we add $\boldsymbol{r}$ on both sides to obtain $F(\boldsymbol{x}) = F(\boldsymbol{a}) + \sum_{i=1}^{d} S(x_i) M\mathcal{S}(\boldsymbol{v}_i)$. □

3.5 Chaining Separable Affine Subspaces

Theorem 1 shows that a separable affine subspace is always mapped to an affine subspace. However, there is no guarantee that the resulting subspace is separable too, which would be needed in order to reiterate the process and get a chain of subspaces.

We argue in Appendix E that it is rather unlikely that the image of a subspace of dimension 2 or more is separable. In the rest of the paper, we focus on affine subspaces of dimension 1, which still contain many elements since q is large in the context of arithmetization-oriented primitives. The following corollary provides a necessary and sufficient condition under which the image of a separable subspace is a separable subspace too. It is an immediate consequence of Theorem 1 and of the definition of separable subspace.

Corollary 1. *Let F be the round function $\mathcal{R}(q, m, \alpha, M, \boldsymbol{r})$ with round constant $\boldsymbol{r}$ and Sbox layer $\mathcal{S}$. Let $V = \boldsymbol{a} + \langle \boldsymbol{v} \rangle$ be a separable subspace of dimension 1. Then, $F(V)$ is a separable subspace if and only if there exists $\lambda \in \mathbb{F}_q$ such that*

$$\forall i \in \mathrm{supp}(M\mathcal{S}(\boldsymbol{v})), \;\; r_i + \lambda[M\mathcal{S}(\boldsymbol{v})]_i + [M\mathcal{S}(\boldsymbol{a})]_i = 0, \tag{1}$$

where $[M\mathcal{S}(\boldsymbol{v})]_i$ denotes the i-th coordinate of the vector $M\mathcal{S}(\boldsymbol{v})$.

This condition is always satisfied when $M\mathcal{S}(\boldsymbol{v})$ has weight 1. Therefore, for any monomial-based SPN and any choice of the round constants, there exists some affine subspace V whose image by the round transformation is separable, which implies that the image of V after two rounds is still an affine subspace.

On the other hand, for any monomial-based SPN, and for any choice of $\boldsymbol{v}$ and $\boldsymbol{a}$ in $\mathbb{F}_q^m$, $\boldsymbol{v} \neq \boldsymbol{0}$, there exist several round constants such that the image of $V = \boldsymbol{a} + \langle \boldsymbol{v} \rangle$ after two rounds is still an affine subspace. We then deduce the following theorem, i.e. for any choice of Sbox and linear layer, and for any number of rounds, there always exist *weak round-constants*, for which an affine subspace of dimension 1 is mapped to an affine subspace. This holds even if the Sbox and the linear layer vary with the round, as in Rescue.

Theorem 2. *Let P be a monomial-based SPN defined by the composition of N round functions $\mathcal{R}(q, m, \alpha_t, M, \boldsymbol{r}_t)$ for $0 \leq t < N$. Then, for any separable affine subspace of dimension 1, $\boldsymbol{a} + \langle \boldsymbol{v} \rangle$, there exist some sequences of round-constants $(\boldsymbol{r}_0, \ldots, \boldsymbol{r}_{N-1})$ such that $P(\boldsymbol{a} + \langle \boldsymbol{v} \rangle)$ is an affine subspace. These weak round-constants include those satisfying, for some $\lambda_1, \ldots, \lambda_{N-1} \in \mathbb{F}_q$,*

$$\begin{cases} r_{t,i} \;=\; -M_i\left[\lambda_{t+1}\mathcal{S}_t(\boldsymbol{v}_t) + \mathcal{S}_t(\boldsymbol{a}_t)\right] \;, & \forall i \in \mathrm{supp}(\boldsymbol{v}_{t+1}), \forall t \in \{0, ..., N-2\} \;, \\ & \text{where } \boldsymbol{v}_{t+1} = M \circ \mathcal{S}_t \circ \ldots \circ M \circ \mathcal{S}_0(\boldsymbol{v}) \\ \boldsymbol{a}_{t+1} \;=\; M\mathcal{S}_t(\boldsymbol{a}_t) + \boldsymbol{r}_t + \lambda_{t+1}\boldsymbol{v}_{t+1} \;, & \end{cases}$$

where $\boldsymbol{a}_0 = \boldsymbol{a}$, M_i denotes the i-th row of matrix M, and S_t (resp. $\mathcal{S}_t$) denotes the Sbox (resp. Sbox layer) at Round t .

Moreover, for any $x \in \mathbb{F}_q$, the image of $(\boldsymbol{a} + x\boldsymbol{v})$ after N rounds equals $\boldsymbol{a}_N + \pi_N(x)\boldsymbol{v}_N$ where $\pi_{t+1}(x) = S_t(\pi_t(x)) - \lambda_{t+1}$ for all $t \geq 0$ and $\pi_0(x) = x$.

Proof. The proof by induction on the number of rounds is a direct consequence of Theorem 1 and Corollary 1. Indeed, the result holds for $N = 1$ for any value of the round-constant $\boldsymbol{r}_0$, and we have that, after one round,

$$P(\boldsymbol{a} + x\boldsymbol{v}) \;=\; P(\boldsymbol{a}) + S_0(x)\boldsymbol{v}_1 \;=\; \boldsymbol{a}_1 + (S_0(x) - \lambda_1)\boldsymbol{v}_1 = \boldsymbol{a}_1 + \pi_1(x)\boldsymbol{v}_1 \;.$$

Assume that the result holds after the first t rounds, i.e., the image of $(\boldsymbol{a} + x\boldsymbol{v})$ equals $\boldsymbol{a}_t + \pi_t(x)\boldsymbol{v}_t$. Let $\boldsymbol{v}_{t+1} = M\mathcal{S}_t(\boldsymbol{v}_t)$. The fact that the constant $\boldsymbol{r}_t$ at Round t satisfies

$$r_{t,i} + M_i\left[\lambda_{t+1}\mathcal{S}_t(\boldsymbol{v}_t) + \mathcal{S}_t(\boldsymbol{a}_t)\right] = 0$$

for all $i \in \mathrm{supp}(\boldsymbol{v}_{t+1})$ equivalently means that the restriction to $\mathrm{supp}(\boldsymbol{v}_{t+1})$ of

$$\boldsymbol{a}_{t+1} = M\mathcal{S}_t(\boldsymbol{a}_t) + \boldsymbol{r}_t + \lambda_{t+1} M \circ \mathcal{S}_t(\boldsymbol{v}_t)$$

vanishes. It follows that $\boldsymbol{a}_{t+1} + \langle \boldsymbol{v}_{t+1} \rangle$ is separable. Then, Theorem 1 applies, and we get that the image of $(\boldsymbol{a}_t + \pi_t(x)\boldsymbol{v}_t)$ after the $(t+1)$th round-function, F_t, is given by

$$\begin{aligned} F_t(\boldsymbol{a}_t + \pi_t(x)\boldsymbol{v}_t) &= M\mathcal{S}_t(\boldsymbol{a}_t) + \boldsymbol{r}_t + S_t(\pi_t(x))M\mathcal{S}_t(\boldsymbol{v}_t) \\ &= \boldsymbol{a}_{t+1} + [S_t(\pi_t(x)) - \lambda_{t+1}]\,\boldsymbol{v}_{t+1} \;=\; \boldsymbol{a}_{t+1} + \pi_{t+1}(x)\boldsymbol{v}_{t+1}\,. \end{aligned}$$

□

A similar result holds if different linear layers are used in the successive rounds, but we made the assumption of a single matrix for the sake of simplicity.

The weak round-constants described in Theorem 2 are exactly those ensuring a chain of separable affine subspaces of dimension 1 through the N rounds of the primitive. However, the image of an affine subspace after N rounds may be a subspace even if some of the intermediate sets are non-separable subspaces, or even not subspaces. This is why the condition in Theorem 2 is a sufficient condition only. Finding a more general condition which captures all situations seems much more difficult.

When the Sboxes satisfy that $S_{2k+1} = S_{2k}^{-1}$ for all k, like in Rescue, or if the same involutive Sbox is used at every round, we even obtain a stronger property since there are some round-constants for which the restriction of the permutation to $(\boldsymbol{a} + \langle \boldsymbol{v} \rangle)$ is affine.

Corollary 2. *Let N be an even integer and let P be a monomial-based SPN defined by the composition of N round functions $\mathcal{R}(q, m, \alpha_t, M, \boldsymbol{r}_t)$ for $0 \leq t < N$ with $\alpha_{2k}\alpha_{2k+1} \equiv 1 \bmod (q-1)$ for all $k < N/2$. Then, for any separable affine subspace of dimension 1, $\boldsymbol{a} + \langle \boldsymbol{v} \rangle$, there exist some sequences of round-constants $(\boldsymbol{r}_0, \ldots, \boldsymbol{r}_{N-1})$ such that P is affine on $\boldsymbol{a} + \langle \boldsymbol{v} \rangle$.*

These weak round-constants include those satisfying

$$\begin{cases} r_{t,i} \;=\; -M_i \circ \mathcal{S}_t(\boldsymbol{a}_t)\,, & \forall i \in \mathrm{supp}(\boldsymbol{v}_{t+1}), \forall t \in \{0, ..., N-2\}\,, \\ & \textit{where } \boldsymbol{v}_{t+1} = M \circ \mathcal{S}_t \circ \ldots \circ M \circ \mathcal{S}_0(\boldsymbol{v}) \\ \boldsymbol{a}_{t+1} \;=\; M\mathcal{S}_t(\boldsymbol{a}_t) + \boldsymbol{r}_t\,, & \end{cases}$$

and $\boldsymbol{a}_0 = \boldsymbol{a}$. In this case, for any $x, y \in \mathbb{F}_q$, $P(\boldsymbol{a} + x\boldsymbol{v}) - P(\boldsymbol{a} + y\boldsymbol{v}) = (x-y)\boldsymbol{v}_N$.

Proof. This follows directly from Theorem 2 applied to the case $\lambda_1 = \ldots = \lambda_{N-1} = 0$. We then have, for all $t \leq N$, that $\pi_t(x) = x^{\alpha_t}$ if t is odd, and $\pi_t(x) = x$ otherwise. □

3.6 Detecting Chains of Separable Subspaces

Theorem 2 provides a sufficient condition for a given separable subspace of dimension 1 to propagate through a monomial-based SPN. In Appendix A, we use this result to estimate the number of weak round-constants. However, such statistical arguments cannot be used to assess the security of a well-specified primitive. To this end, we devised Algorithm 1, which checks the existence of such a chain of separable subspaces, works as follows. First, compute all pairs of separable affine subspaces V and W such that W is the image of V under the round function F. This guarantees that the image of V after two rounds is an affine subspace. Checking whether the image of W after one additional round is still separable or not, we can construct the (potentially empty) list of all chains of three consecutive separable spaces. If this list is not empty, then we may have found an abnormally long chain.

Since only a few pairs of separable subspaces are images of each other by F, the most expensive part of the algorithm is the first one. Finding all separable $V = \boldsymbol{a} + \langle \boldsymbol{v} \rangle$ and

$W = F(\boldsymbol{a}) + \langle \boldsymbol{w} \rangle$ such that $F(V) = W$ can be done by first fixing the respective supports I_1 and I_2 of $\boldsymbol{v}$ and $\boldsymbol{w}$. We can then assume that $\boldsymbol{a} + \langle \boldsymbol{v} \rangle$ is the canonical representation of V, implying that $v_{i_S} = 1$ for $i_S = \min I_1$ and $\mathrm{supp}(\boldsymbol{a}) = \{0, \ldots, m-1\} \setminus I_1$. As in Theorem 2, the fact that W is separable means that there exists $\lambda \in \mathbb{F}_q$ such that the support of $\boldsymbol{b} = F(\boldsymbol{a}) - \lambda \boldsymbol{w}$ is included in $\{0, \ldots, m-1\} \setminus I_2$. Since the support of a vector is invariant under $\mathcal{S}$, we can define $\boldsymbol{x}$ and $\boldsymbol{c}$ by $\boldsymbol{x}_i = \boldsymbol{v}_i^\alpha$ and $\boldsymbol{c}_i = \boldsymbol{a}_i^\alpha$ for all $0 \leq i < m$. Then, the condition in Theorem 2 holds if and only if there exists $\lambda \in \mathbb{F}_q$ such that

$$\begin{cases} \boldsymbol{w}_i - M_i \boldsymbol{x} &= 0 \quad \text{for all } 0 \leq i < m \\ \boldsymbol{b}_i - \boldsymbol{r}_i - M_i \boldsymbol{c} + \lambda \boldsymbol{w}_i &= 0 \quad \text{for all } 0 \leq i < m\ . \end{cases} \tag{2}$$

This corresponds to a quadratic system of $2m$ equations with $2m$ unknowns: the $(m-1)$ unknown coordinates of $\boldsymbol{x}$ and $\boldsymbol{c}$, the m possibly nonzero coordinates of $\boldsymbol{w}$ and $\boldsymbol{b}$, and λ.

We have implemented this algorithm in `SAGE`.[5] It can successfully identify the subspace chain in the Stir permutation defined in Section 5.2. For these parameters, System (2) turns out to have exactly one solution for each valid pair of supports (I_1, I_2).

Algorithm 1 Algorithm for finding all chains of separable one-dimensional affine subspaces over three rounds.

Compute the list $\mathcal{L}$ of all pairs of possible supports (I_1, I_2) for a pair $(\boldsymbol{x}, M\boldsymbol{x})$.
for all (I_1, I_2) in $\mathcal{L}$ **do**
  Determine all 5-tuples $(\boldsymbol{c}, \boldsymbol{x}, \boldsymbol{b}, \boldsymbol{w}, \lambda)$ with $\mathrm{supp}(\boldsymbol{x}) = I_1$, $\mathrm{supp}(\boldsymbol{c}) = \{0, \ldots, m-1\} \setminus I_1$, $\mathrm{supp}(\boldsymbol{w}) = I_2$, $\mathrm{supp}(\boldsymbol{b}) = \{0, \ldots, m-1\} \setminus I_2$, $\lambda \in \mathbb{F}_q$ that satisfy (2) using a Gröbner basis algorithm.
  for all $(\boldsymbol{c}, \boldsymbol{x}, \boldsymbol{b}, \boldsymbol{w}, \lambda)$ solutions of the previous system **do**
    if $\boldsymbol{b} + \langle \boldsymbol{w} \rangle$ maps to yet another separable space through the next round **then**
      $\boldsymbol{v}_i \leftarrow (\boldsymbol{x}_i)^{1/\alpha}$ and $\boldsymbol{a}_i \leftarrow (\boldsymbol{c}_i)^{1/\alpha}$ for all $0 \leq i < m$.
      $\boldsymbol{a} + \langle \boldsymbol{v} \rangle$ and its successive images form a chain of separable subspaces.
    end if
  end for
end for

4 High Differential Uniformities of monomial-based SPNs

The previous analysis was originally motivated by some anomalies in the *fixed-key maximal differential probability* of some small instances of the block cipher Rescue that were pointed out in [BCL+20]. As we will see, these anomalies are side effects of particular subspace chains. This observation can be pushed further, and we will show that it is in fact possible to trigger much more sophisticated patterns. Before going further, let us recall some definitions.

Definition 4 (DDT and differential uniformity [Nyb94])**.** Let $F : \mathbb{F}_q^n \to \mathbb{F}_q^n$. The DDT (difference distribution table) of F is an array of size $q^n \times q^n$ whose coefficients are:

$$\mathsf{DDT}_F[\alpha, \beta] := |\{x \in \mathbb{F}_q : F(x+\alpha) - F(x) = \beta\}|\ .$$

The *differential uniformity* of F is $\delta(F) = \max_{\alpha \neq 0, \beta} \mathsf{DDT}_F[\alpha, \beta]$.

It is worth noting that, for a keyed function such as a block cipher, the fixed-key maximal differential probability coincides with the differential uniformity of the function defined by a specific key.

[5] Our implementation is provided in a separate file along with this submission.

4.1 Fixed-Key Maximal Differential Probability of some monomial-based SPNs

When the Sbox alphabet q is large, computing the differential uniformity becomes unfeasible. Instead, in the case of an iterated permutation, an alternative approach could be to derive some upper bounds on the differential uniformity of the function from some properties of its building-blocks. However, all known results in this direction rely on some strong assumptions, typically on the fact that the sequence of differences through the permutation is a Markov chain [LMM91]. This kind of results (e.g. [DR02, HLL+01, PSLL03, CR15]) may be relevant for keyed primitives since they usually provide some information on the maximum expected differential probability (MEDP) of the whole family of permutations. However, the MEDP has no practical significance in several contexts, for example, when a specific instance of the family is used as an inner permutation in a sponge, like in Rescue-Prime. In that case, the expected maximum differential probability (EMDP, instead of MEDP) would be much more relevant, and these two quantities may significantly differ as pointed out in [CDL16, Page 378].

Even for a keyed primitive like a block cipher, the average probability of a given differential often provides little information on the behavior of the primitive with respect to differential cryptanalysis. Indeed, as extensively analyzed in [BR22], many ciphers are vulnerable to differential cryptanalysis only for some keys. It is therefore important to estimate the differential uniformity of the fixed-key primitive, even for a block cipher. The notion of quasi-differential trails recently introduced in [BR22] provides a very elegant framework for computing fixed-key probabilities of differential trails for iterated functions. But, its complexity is expensive, especially when focusing on differentials instead of trails.

Still, experiments are possible for toy ciphers with a very small block size, as was shown in [BCL+20]. Their experimental results on variants of Rescue with small p are summarized in Figure 2, which was obtained using a reference implementation of this cipher—and thus using a proper key derivation algorithm. As we can see, for $p = 13$, one instance has its differential uniformity decrease as expected for the first 3 rounds but then spike at the 4th to reach a value just under $1.5p$. Unfortunately (from the designer's point if view), abnormally high differential uniformity are compatible with a strong round-constant generation.

As illustrated by the following proposition, such a high differential uniformity can be explained by the presence of chains of affine subspaces.

Proposition 2. *Let N be an even integer and let P be the permutation defined by the composition of N round functions $\mathcal{R}(q, m, \alpha_t, M_t, \boldsymbol{r}_t)$ for $1 \leq t \leq N$ with $\alpha_{2k}\alpha_{2k+1} \equiv 1 \bmod (q-1)$ for all $k < N/2$. Then, there exist some round constants for which P has differential uniformity at least q, such as those defined in Corollary 2 for some pair $(\boldsymbol{a}, \boldsymbol{v})$.*

Proof. For any sequence of round-constants satisfying the relation given in Corollary 2 for $(\boldsymbol{a}, \boldsymbol{v})$, there exists $\boldsymbol{v}_N$ such that $P(\boldsymbol{a}+x\boldsymbol{v}) - P(\boldsymbol{a}+y\boldsymbol{v}) = (x-y)\boldsymbol{v}_N$ for any $x, y \in \mathbb{F}_q$. So, for all $x \in \mathbb{F}_q$, $P(\boldsymbol{a}+(x+1)\boldsymbol{v}) - P(\boldsymbol{a}+x\boldsymbol{v}) = \boldsymbol{v}_N$, implying that $\mathsf{DDT}_F[\boldsymbol{v}, \boldsymbol{v}_N] \geq q$. □

4.2 Application to a Variant of AES

This behaviour can be observed in other monomial-based SPNs. As an illustration, we consider a variant of AES [AES01] where the Sbox corresponds to the monomial x^{-1} (with the convention that $0^{-1} = 0$). Thus, the only difference with AES is that the $\mathbb{F}_2$-affine transformation applied at the end of the AES Sbox has been removed.

Let $\boldsymbol{k}_t = (k_t^0, k_t^1, \ldots, k_t^{m-1})$ denote the subkey inserted at the end of Round t, $1 \leq t \leq N$. This AES variant[6] follows the model analyzed in Theorem 2, where $\boldsymbol{k}_0$ is added at the

[6] Removing the last MixColumns as in the AES would only change last subspace without impacting the others, so we keep it for the sake of simplicity.

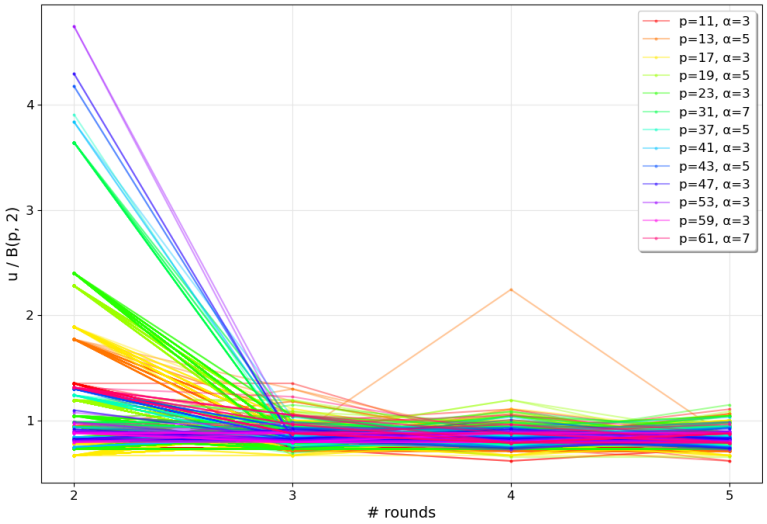


Figure 2: The evolution of the differential uniformity of several RESCUE instances with $m = 2$ through their rounds, divided by $B(p, m)$ (see Appendix C). Each color corresponds to a different prime characteristic p, from 13 to 61.

beginning. Applying Corollary 2 then yields the following.

Proposition 3. *Let P be the function corresponding to N rounds of the monomial-based variant of AES, with N even. Let $\boldsymbol{v}_0$ be a nonzero element in $\mathbb{F}_{2^k}^m$ and $\boldsymbol{a}_0 \in \mathbb{F}_{2^k}^m$. Then, for any round-key sequence such that the vectors $\boldsymbol{v}_t$ and $\boldsymbol{a}_t$ defined by*

$$\begin{aligned} \boldsymbol{v}_{t+1} &= M \circ \mathsf{SB}(\boldsymbol{v}_t), & 0 \leq t < N \\ \boldsymbol{a}_{t+1} &= M \circ \mathsf{SB}(\boldsymbol{a}_t) + \boldsymbol{k}_{t+1}, & 0 \leq t < N \end{aligned}$$

satisfy $a_{t,j} = 0$ for all $j \in \mathrm{supp}(\boldsymbol{v}_t)$ and all $0 \leq t < N - 1$, we have

$$P(\boldsymbol{a}_0 + \boldsymbol{k}_0 + x\boldsymbol{v}_0) = \boldsymbol{a}_N + x\boldsymbol{v}_N, \text{ for all } x \in \mathbb{F}_{2^k}\,.$$

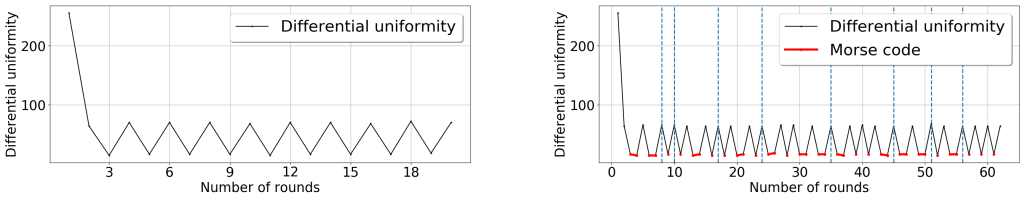
In particular, in this case, $\mathsf{DDT}_P[\alpha\boldsymbol{v}_0, \alpha\boldsymbol{v}_N] \geq 2^k$ for all $\alpha \in \mathbb{F}_{2^k}$.

Figure 3(a) shows how the differential uniformity of the monomial-based variant of AES varies with the number of rounds, for a round-key sequence chosen according to the conditions defined in Proposition 3. These results have been obtained for a small-scaled variant of the cipher, where the Sbox operates on a field of size $q = 2^6$, and the inner state contains $m = 2$ elements. For any even number of rounds, the permutation has differential uniformity slightly higher than the size of the Sbox alphabet, while for any odd number of rounds, it is close to the expected value of the differential uniformity for a random permutation, namely 11.5 (see Corollary 3).

In fact, an even stranger behaviour can be observed for some round-keys, where spikes in the differential uniformity can occur every 2 or 3 rounds. This behaviour relies on the following property of the inverse mapping, which holds in characteristic 2 only.

Lemma 1. *Let $x, b \in \mathbb{F}_{2^k}$ for some $k \geq 1$, with $x, b \neq 0$ and $x \neq b^{-1}$. Then,*

$$(x^{-1} + b)^{-1} = b^{-1}(xb + 1)^{-1} + b^{-1}\,.$$



(a) (b)

Figure 3: (a) Differential uniformity of the monomial-based variant of AES where the key is chosen as detailed in Proposition 3 to get a spike at every even round. (b) Differential uniformity which encodes "MERRYXMAS" in Morse code, without spaces. Blue lines separate letters.

Proof. The right-hand side of the equation can be rewritten as

$$\frac{1}{b(xb+1)} + \frac{1}{b} \;=\; (1+xb+1)\,\frac{1}{b(xb+1)} \;=\; \frac{x}{xb+1} = \frac{1}{x^{-1}+b}\,. \qquad \square$$

Based on this lemma, we can exhibit some round-keys for which the monomial-based variant of AES after three rounds is affine on a given affine subspace.

Proposition 4. *Let P be the function corresponding to three rounds of the monomial-based variant of AES. Let $\boldsymbol{v}_0$ be a nonzero element in $\mathbb{F}_{2^k}^m$. Assume that the round-keys are defined by $\boldsymbol{k}_1 = b\boldsymbol{v}_1$, $\boldsymbol{k}_2 = b^{-1}\boldsymbol{v}_2$, and $\boldsymbol{k}_3 = c\boldsymbol{v}_3$ for some $b \neq 0$ and c in $\mathbb{F}_{2^k}$, where $\boldsymbol{v}_{t+1} = M \circ \mathsf{SB}(\boldsymbol{v}_t)$. Then, for all x in $\mathbb{F}_{2^k}$ such that $x \neq 0$ and $x \neq b^{-1}$, we have that*

$$P(x\boldsymbol{v}_0 + \boldsymbol{k}_0) = (b^2x + b + c)\boldsymbol{v}_3\,.$$

In particular, for all x and α such that $x \notin \{0, \alpha, b^{-1}, \alpha + b^{-1}\}$, we have

$$P(x\boldsymbol{v}_0 + \boldsymbol{k}_0) - P(x\boldsymbol{v}_0 + \alpha\boldsymbol{v}_0 + \boldsymbol{k}_0) = b^2\alpha\boldsymbol{v}_3\,,$$

and, for all $x \in \mathbb{F}_{2^k}$, we have

$$P(x\boldsymbol{v}_0 + \boldsymbol{k}_0) - P(x\boldsymbol{v}_0 + b^{-1}\boldsymbol{v}_0 + \boldsymbol{k}_0) = b\boldsymbol{v}_3\,.$$

Proof. Let us first compute the values $\boldsymbol{a}_t$ as defined in Theorem 2 where $\boldsymbol{a}_0 = 0$, $\lambda_1 = -b$, $\lambda_2 = -b^{-1}$ and $\lambda_3 = -c$. Then we get that, for all $1 \leq t \leq 3$,

$$\boldsymbol{a}_t = \boldsymbol{k}_t + \lambda_t\boldsymbol{v}_t = 0\,,$$

because of the choice of the round-keys. Therefore, these round-keys satisfy the conditions given in Theorem 2. It follows that the image of $\boldsymbol{k}_0 + \langle\boldsymbol{v}_0\rangle$ after N rounds of P equals $\langle\boldsymbol{v}_N\rangle$, for $1 \leq N \leq 3$. Moreover, the image after N rounds of each individual element $(x\boldsymbol{v}_0 + \boldsymbol{k}_0)$ in $\boldsymbol{k}_0 + \langle\boldsymbol{v}_0\rangle$ is equal to $\pi_N(x)\boldsymbol{v}_N$ with $\pi_1(x) = x^{-1} + b$, $\pi_2(x) = (x^{-1} + b)^{-1} + b^{-1}$, and $\pi_3(x) = \pi_2(x)^{-1} + c$. It follows from the previous lemma that, for all $x \notin \{0, b^{-1}\}$,

$$\pi_2(x) = (x^{-1} + b)^{-1} + b^{-1} = b^{-1}(xb+1)^{-1}\,,$$

leading to $\pi_3(x) = b(xb+1) + c$. We deduce that, for any $\alpha \in \mathbb{F}_{2^k}$, the equation

$$P(x\boldsymbol{v}_0 + \boldsymbol{k}_0) - P(x\boldsymbol{v}_0 + \alpha\boldsymbol{v}_0 + \boldsymbol{k}_0) = b^2\alpha\boldsymbol{v}_3$$

holds for all $x \notin \{0, \alpha, b^{-1}, \alpha + b^{-1}\}$. However, when $\alpha = b^{-1}$, we can check that it also holds for $x \in \{0, b^{-1}\}$ since $P(\boldsymbol{k}_0) = c\boldsymbol{v}_3$ and $P(b^{-1}\boldsymbol{v}_0 + \boldsymbol{k}_0) = (b + c)\boldsymbol{v}_3$. $\square$

In order to show how much control a malevolent designer could have over the differential uniformity of such an SPN, we combine the previous proposition and Proposition 3 to exhibit a round-key sequence for which any given binary message is encoded in the curve representing the evolution of the differential uniformity of the iterated permutation, as in Figure 3(b). The round-keys are chosen to guarantee that the image after N rounds of $\boldsymbol{k}_0 + \langle \boldsymbol{v}_0 \rangle$ equals $\langle \boldsymbol{v}_N \rangle$. Moreover, we make some specific choice to ensure that the permutation is an affine function on this subspace after some rounds. Indeed, if we assume that the function after t rounds is affine on $\langle \boldsymbol{v}_t \rangle$,

- choosing $\boldsymbol{k}_{t+1} = 0$ and $\boldsymbol{k}_{t+2} = c\boldsymbol{v}_{t+2}$ for some c implies that the function after $(t+2)$ rounds is affine on $\langle \boldsymbol{v}_{t+2} \rangle$. The variation of the differential uniformity between Rounds t and $(t+2)$ then looks like "high, low, high", a dit $(\cdot)$ in Morse code.

- choosing $\boldsymbol{k}_{t+1} = b\boldsymbol{v}_{t+1}$, $\boldsymbol{k}_{t+2} = b^{-1}\boldsymbol{v}_{t+2}$ and $\boldsymbol{k}_{t+3} = c\boldsymbol{v}_{t+3}$ for some $b \neq 0$ and c, implies that the function after $(t+3)$ rounds is affine on $\langle \boldsymbol{v}_{t+3} \rangle$. The variation of the differential uniformity between Rounds t and $(t+3)$ then looks like "high, low, low, high", a dah $(-)$ in Morse code.

This algorithm, applied to a small-scaled version of the monomial-based variant of the AES, with $m = 2$ and $q = 2^6$, then enables us to exhibit a round-key sequence for which the differential uniformity varies through the rounds as depicted on Figure 3(b).

5 Purposeful Weaknesses Against Chains of Subspaces

In this section, we exhibit some other weaknesses that may arise from the existence of chains of subspaces. While our attacks can a priori *not* be applied to functions satisfying the full description of e.g. Rescue with its original key-schedule, or Rescue-Prime with randomly chosen round-constants, we show how to build primitives that could allow such attacks. Our aim is to illustrate the fact that the arguments currently used when discussing the security of arithmetization-oriented primitives are not sufficient for assessing their real security level, since the very same arguments apply to the following weak primitives. In summary, security arguments in this area usually boil down to a simple wide trail argument against differential and linear attacks coupled with a complexity analysis of algebraic attacks. There should a priori not be anything wrong with this approach, but we argue here that other attack vectors are not properly captured by such analyses.

First, we present in Section 5.1 how to ensure the presence of a subspace chain in a permutation or block cipher using a slight variant of the Rescue round function.[7] Using this tool, we construct a permutation intended to build a sponge-based hash function, Stir, (Section 5.2). Using the extra freedom given by the choice of arbitrary round constants, we ensure the presence of a chain of affine subspaces that, thanks to their structure, enables trivial solutions for the CICO problem (whose definition we will recall).

Then, we apply the Malicious framework [PW20] to construct Snare, a backdoored tweakable block cipher over a large prime field (Section 5.3), which reuses the overall structure of Rescue but uses a different key schedule, and a specific linear layer. Based on similar ideas to those in Malicious-AES and Boomslang [BBFL22], Snare weaponizes subspace chains to enforce the existence of a specific invariant subspace, whose presence is used to significantly speed-up a key-recovery attack.

For the sake of simplicity, these two primitives are defined over a field with odd characteristic, i.e., q is not a power of 2.

[7] The analysis presented in [BCLR17] on the existence of weak round-constants does not apply here since it only considers subspaces that are invariant under each layer of the cipher. Here, the subspaces are not necessarily under the linear layer or the nonlinear layer.

5.1 Forcing a Subspace Chain

Consider a potential chain $V_0 \to V_1 \to V_2 ... \to V_N$ for a monomial-based SPN with round function $\mathcal{R}(q, m, \alpha, M, \boldsymbol{r}_t)$. Using the notation of Section 3, we may write $V_t = \{\boldsymbol{a}_t + x\boldsymbol{v}_t, x \in \mathbb{F}_q\}$ for all $0 \leq t \leq N$. Where appropriate, the first index denotes the round number and the second one denotes the index of the coordinate. Recall that Rescue uses both $\mathcal{S}$ and $\mathcal{S}^{-1}$. We write $\mathcal{S}_t = \mathcal{S}$ whenever t is even and $\mathcal{S}^{-1}$ whenever t is odd.

By adding together the relationships between consecutive separable affine spaces and the conditions described in Section 3.5 guaranteeing that they do in fact chain, we obtain a system of equations over this chain, where the pairs $(\boldsymbol{r}_t, \lambda_{t+1})$ are chosen in order to satisfy the conditions in Theorem 2, namely

$$\forall 0 \leq t < N-1, \begin{cases} \boldsymbol{v}_{t+1} = M \circ \mathcal{S}_t(\boldsymbol{v}_t) \\ \boldsymbol{a}_{t+1} = M \circ \mathcal{S}_t(\boldsymbol{a}_t) + \boldsymbol{r}_t + \lambda_{t+1}\boldsymbol{v}_{t+1} \\ \forall i \in \text{supp}(\boldsymbol{v}_{t+1}), \boldsymbol{a}_{t+1,i} = 0\ . \end{cases}$$

To ensure that the subspace chain happens, we choose M such that all $\boldsymbol{v}_t$, $0 \leq t < N$, are equal to a unique vector $\boldsymbol{u}$ with $\boldsymbol{u}_{m-1} = 0$ (the importance of this last requirement will become obvious in Section 5.2). It must also be such that there exists a scalar μ verifying:

$$M \circ \mathcal{S}(\boldsymbol{u}) = M \circ \mathcal{S}^{-1}(\boldsymbol{u}) = \mu\boldsymbol{u}\ . \tag{3}$$

A vector $\boldsymbol{u}$ satisfying Equation (3) is a priori not an eigenvector of M, which makes it more difficult to detect.

The fact that Rescue uses both $\mathcal{S}$ and $\mathcal{S}^{-1}$ limits our possible choices for $\boldsymbol{u}$. In order to simplify this task, we will try to get $\boldsymbol{u}$ such that $\mathcal{S}(\boldsymbol{u}) = \mathcal{S}^{-1}(\boldsymbol{u})$. Since $S(x) = x^\alpha$, we need solutions[8] of $x^{\alpha^2} = x$, so that the coordinates of $\boldsymbol{u}$ will be (α^2-1)-th roots of unity. Since α and q are both odd when $p \neq 2$, it holds that $\gcd(\alpha^2-1, q-1) \geq 2$, and we can always find $\gcd(\alpha^2-1, q-1)$-th primitive roots of unity. As said above, it is better if $\boldsymbol{u}$ is not an eigenvector of M, which implies that its nonzero coefficients should take different values.[9] Not only that, if we define $\boldsymbol{u}$ as a succession of 1 and -1 (as 2nd roots of unity) with 0 as its last coefficient, $\boldsymbol{u}$ would still be an eigenvector of M, due to the fact that $1^\alpha = 1$ and $(-1)^\alpha = -1$. Instead, we can be a little more subtle by using $\gcd(\alpha^2-1, q-1)$-th roots of unity that would not make $\boldsymbol{u}$ an eigenvector.[10]

Example 1 (Computation of $\boldsymbol{u}$)**.** Let us use the parameters of the permutations in the ZK Hash Bounties [Eth21], i.e. $p = 18446744073709551615$, $\alpha = 3$ and $m = 3$. This gives $\gcd(\alpha^2-1, p-1) = 4$, and we are able to define $\boldsymbol{u}$ using ρ, a 4-th primitive root of unity, among 2 possible choices in $\mathbb{F}_p$, both verifying $\rho^2 = -1$. For instance, we choose $\boldsymbol{u} = (1, \rho, 0)^T$. We then have $\mathcal{S}(\boldsymbol{u}) = \mathcal{S}^{-1}(\boldsymbol{u}) = (1, -\rho, 0)^T$.

Now that the space in which $\boldsymbol{u}$ must live is specified, we need to find M such that Equation (3) holds for such a $\boldsymbol{u}$, i.e. such that $M \circ \mathcal{S}(\boldsymbol{u}) = \mu\boldsymbol{u}$ for some $\mu \in \mathbb{F}_q$. A usual method to construct MDS matrices is based on so-called Cauchy matrices.

Definition 5 (Cauchy Matrices)**.** A matrix M of size $(m \times m)$ is a Cauchy matrix if and only if there exist scalars $\sigma_1, \sigma_2, ..., \sigma_m, \tau_1, \tau_2, ..., \tau_m$ such that, for all $1 \leq i, j \leq m$, $\sigma_i + \tau_j$

[8] Note that we assume $p > 2$. In characteristic 2, the conditions would be different.

[9] Otherwise, the multiplicative property of the monomial would imply that $\boldsymbol{u}$ is an eigenvector of M.

[10] Of course, for values of q such that $\gcd(\alpha^2-1, q-1) = 2$, we will have no choice but to use successions of 1 and -1, i.e. impose a special eigenspace on M. When q is a prime and α is the smallest integer such that $\gcd(\alpha, q-1) = 1$, this situation occurs if and only if $q \equiv 11 \bmod 12$.

is nonzero and

$$M = \begin{pmatrix} (\sigma_1 - \tau_1)^{-1} & (\sigma_1 - \tau_2)^{-1} & \dots & (\sigma_1 - \tau_m)^{-1} \\ (\sigma_2 - \tau_1)^{-1} & (\sigma_2 - \tau_2)^{-1} & \dots & (\sigma_2 - \tau_m)^{-1} \\ \vdots & \vdots & & \vdots \\ (\sigma_m - \tau_1)^{-1} & (\sigma_m - \tau_2)^{-1} & \dots & (\sigma_m - \tau_m)^{-1} \end{pmatrix} .$$

Choosing a good M for our purpose then consists in finding distinct $\sigma_1, \sigma_2, ..., \sigma_m$, $\tau_1, \tau_2, ..., \tau_m$ which verify the constraints in Equation (3). These constraints correspond to m non-linear equations with $2m + 1$ unknowns (including μ). By multiplying each line $1 \leq i \leq m$ by $(\sigma_i - \tau_1)(\sigma_i - \tau_2)...(\sigma_i - \tau_m)$, and setting $m = 3$ for readability, we obtain:

$$\begin{cases} (\sigma_1 - \tau_2)\boldsymbol{u}_1^\alpha + (\sigma_1 - \tau_1)\boldsymbol{u}_2^\alpha - \mu(\sigma_1 - \tau_1)(\sigma_1 - \tau_2)\boldsymbol{u}_1 = 0 \\ (\sigma_2 - \tau_2)\boldsymbol{u}_1^\alpha + (\sigma_2 - \tau_1)\boldsymbol{u}_2^\alpha - \mu(\sigma_2 - \tau_1)(\sigma_2 - \tau_2)\boldsymbol{u}_2 = 0 \\ (\sigma_3 - \tau_2)\boldsymbol{u}_1^\alpha + (\sigma_3 - \tau_1)\boldsymbol{u}_2^\alpha = 0 \, . \end{cases}$$

This gives m equations, $m - 1$ of which have degree $m - 1$ with the last one having degree $m - 2$. We can see that solving this system is not hard when m is small, as is typically the case. Moreover, by fixing μ and $\tau_1, ..., \tau_m$, we get m independent univariate equations of degree $m - 1$ (except for the last one which is of degree $m - 2$).

In practice, we can sample μ and distinct $\tau_1, ...\tau_m$ randomly until the univariate solver natively present in `SAGE` [The20] finds a distinct solution for each equation such that for all $1 \leq i \leq m, 1 \leq j \leq m$, $\sigma_i \neq \tau_j$. We have found that sampling one tuple is usually enough. The code in `SAGE` used to generate such matrices is given in Appendix B.1.

5.2 The Weak Hash Function Stir

The CICO problem. Before introducing the specification of Stir, we recall the CICO problem and its role in the security analysis of permutations [BDPA11]. As we will see later, the existence of a chain of subspaces makes the resolution of this problem easier.

The state-of-the-art problem used to analyze sponge functions is the *CICO* (*Constrained-input constrained-output*) problem. It has to be difficult to solve, otherwise the security of the hash function may be compromised. It is often used as a proxy to estimate the security provided by a public permutation as, for instance, the complexity of a key recovery does not apply in this context. It is actually the problem that had to be solved for the *ZK Hash Function Cryptanalysis Bounties 2021* organized by *Ethereum* [Eth21], which was tackled in [BBLP22]. The CICO problem is defined as follows.

Definition 6 (CICO Problem)**.** Let $f : \mathbb{F}_q^m \to \mathbb{F}_q^m$ be a mapping, and $c < m$ an integer. We define the *CICO problem* as finding $\boldsymbol{x} \in \mathbb{F}_q^m$ such that the last c coordinates of $\boldsymbol{x}$ *and* of $f(\boldsymbol{x})$ are equal to zero.

Crucially, the set of elements of $\mathbb{F}_q^m$ whose last c coordinates are 0 is a separable affine space of dimension $(m - c)$. In a more limited scope, we may consider the separable affine spaces of dimension 1 whose last c coordinates of the basis vector *and* of the offset are 0.

Specification. Stir is a hash function following the sponge construction. Its inner permutation operates on tuples of m elements of $\mathbb{F}_p$, where p is a prime. The specification of this permutation is very similar to that of Rescue: its even rounds consist of a layer of S-boxes $x \mapsto x^\alpha$, followed by a multiplication of the state by an MDS matrix M, and then a round-constant addition. Its odd rounds are very similar, except that the S-box layer is replaced by its inverse defined as explained below. As we can choose the round constant freely, we can enforce the existence of subspace chains.

Here, we consider a potential chain of affine subspaces of dimension 1, $V_t = \boldsymbol{a}_t + \langle \boldsymbol{u} \rangle$, for $0 \leq l \leq N$, where $\boldsymbol{u}$ is defined as in Section 5.1, i.e., it satisfies Equation (3). If the last coordinates of $\boldsymbol{a}_0$ and $\boldsymbol{a}_N$ are equal to 0, then each input in $V_0 = \boldsymbol{a}_0 + \langle \boldsymbol{u} \rangle$ is a solution of the CICO problem. Ensuring that the subspaces V_t, $0 \leq t \leq N$ form a chain of separable affine subspaces for Stir then boils down to finding appropriate round constants.

Since $\boldsymbol{u}$ was chosen so that all of its coordinates are nonzero (as roots of unity), except the last one, and since we are interested in inputs with their last coordinates always equal to 0, we can take $\boldsymbol{a}_0 = 0$. The round constants $\boldsymbol{r}_0, \boldsymbol{r}_1, ..., \boldsymbol{r}_{N-2}$ are then determined by the conditions of Theorem 2. Since all $\boldsymbol{v}_t$ are equal to $\boldsymbol{u}$ and $\mathrm{supp}(\boldsymbol{u}) \subseteq \{0, \ldots, m-2\}$, the conditions can be rewritten as: there exist $\lambda_1, \ldots, \lambda_{N-1}$ such that

$$\forall 0 \leq t < N-1,\ \forall 0 \leq i \leq m-2,\ \boldsymbol{r}_{t,i} = -\lambda_{t+1} u_i - M_i \mathcal{S}_t(\boldsymbol{a}_t),$$

where M_i denotes the ith row of M.

In order to compute $\boldsymbol{r}_t$, we can randomly sample $\boldsymbol{r}_{t,0}$ as well as $\boldsymbol{r}_{t,m-1}$ (the latter does not have any constraint) in $\mathbb{F}_p$, and then insert the expression of λ_{t+1} in the other equations, which yields explicit formulas to compute $\boldsymbol{r}_t$, for all $0 \leq t < N-1$:

$$\begin{cases} \boldsymbol{r}_{t,0}, \boldsymbol{r}_{t,m-1} \in \mathbb{F}_p \\ \forall 1 \leq i \leq m-2, \boldsymbol{r}_{t,i} = \left(\boldsymbol{r}_{t,0} + M_0 \mathcal{S}(\boldsymbol{a}_t)\right) u_0^{-1} u_i - M_i \mathcal{S}(\boldsymbol{a}_t)\ . \end{cases} \tag{4}$$

There is a small exception for the last round, since $\boldsymbol{r}_{N-1,m-1}$ must be chosen in order to have $\boldsymbol{a}_{N,m-1} = 0$, and thus to ensure the existence of many simple solutions for the CICO problem. This gives, similarly to the other coefficients:

$$\boldsymbol{r}_{N-1,m-1} = \left(\boldsymbol{r}_{t,0} + M_0 \mathcal{S}(\boldsymbol{a}_t)\right) u_0^{-1} u_{m-1} - M_{m-1} \mathcal{S}(\boldsymbol{a}_{N-1}) = -M_{m-1} \mathcal{S}(\boldsymbol{a}_{N-1})\,, \tag{5}$$

since $u_{m-1} = 0$.

Finally, by construction, these choices of M and $\boldsymbol{r}_t$ guarantee that the linear subspace V_0 is mapped to the output affine subspace V_N, both having their last coordinate always equal to 0. Thus, the designer has access to p solutions to the CICO problem that are all included in the same subspace.

Stir has been designed such that there exists a chain of subspaces propagating through the function, starting from a linear subspace $V_0 = \langle \boldsymbol{u} \rangle$, and with affine subspaces $V_t = \boldsymbol{a}_t + \langle \boldsymbol{u} \rangle$ in all intermediate states. Instead, if we want V_0 and V_N to be affine subspaces too, i.e., with $\boldsymbol{a}_0$ and $\boldsymbol{a}_N$ nonzero, we have to find a nonzero vector $\boldsymbol{a}_0$ whose last coordinate vanishes and which satisfies $\mathrm{supp}(\boldsymbol{a}_0) \cap \mathrm{supp}(\boldsymbol{u}) = \emptyset$. This implies that $|\mathrm{supp}(\boldsymbol{u})| \leq m-2$, and requires $m \geq 5$ when M is MDS, as shown by the following lemma.

Lemma 2. *Let M be an $m \times m$-MDS matrix over $\mathbb{F}_q$ and $\mathcal{S}$ be a nonlinear layer over $\mathbb{F}_q^m$. If there exists $\boldsymbol{u} \in \mathbb{F}_q^m$ such that $M \circ \mathcal{S}(\boldsymbol{u}) = \mu \boldsymbol{u}$ for some $\mu \neq 0$, then*

$$|\mathrm{supp}(\boldsymbol{u})| \geq \frac{m+1}{2}\ .$$

Proof. By hypothesis, $\mathrm{supp}(M \circ \mathcal{S}(\boldsymbol{u})) = \mathrm{supp}(\mu \boldsymbol{u}) = \mathrm{supp}(\boldsymbol{u})$. Since M is MDS, we deduce that

$$|\mathrm{supp}(M \circ \mathcal{S}(\boldsymbol{u}))| + |\mathrm{supp}(\mathcal{S}(\boldsymbol{u}))| = 2|\mathrm{supp}(\boldsymbol{u})| \geq m+1\ . \qquad \square$$

Thus, if it is also required that $|\mathrm{supp}(\boldsymbol{u})| \leq m-2$, then we need that $m \geq 5$.

By comparison with a uniformly random sampling, the round constants in Stir satisfy $m-2$ constraints at each round (resp. $m-1$ constraints at the last round). This is similar to sampling, for all t, the restriction of $\boldsymbol{r}_t$ to $\mathrm{supp}(\boldsymbol{u})$, $(\boldsymbol{r}_t)_{|\mathrm{supp}(\boldsymbol{u})}$, randomly in an affine space of basis $(\boldsymbol{u})_{|\mathrm{supp}(\boldsymbol{u})}$ and of offset $-M \circ S(\boldsymbol{a}_t)_{|\mathrm{supp}(\boldsymbol{u})}$. This pattern seems a priori hardly recognizable, especially when $m = 3$, since each vector $\boldsymbol{r}_t$ only has one linear constraint among its coordinates. However, Algorithm 1 successfully detects the existence of a chain of subspaces through the permutation, as can be checked with our implementation of Stir in `SAGE` .

5.3 The Backdoored Tweakable Block Cipher Snare

We now use the variant of the Malicious framework presented in [BBFL22] to design a backdoored tweakable block cipher, Snare, based on the existence of chain of subspaces over the cipher. The Malicious framework provides a generic construction of a secure tweakable block cipher for which a specific tweak $T^\star$ acts as a backdoor for recovering the secret key. An interesting feature of this framework is that discovering this backdoor is computationally difficult, even if its general form is known.

Specification. Snare encrypts a tuple of m elements of $\mathbb{F}_p$ using a key K and a tweak T in $\mathbb{F}_p$, where p is a prime with a bitlength larger than the intended security level. In what follows, we let[11] $m = 3$. The round function reuses the inner permutation of Stir, where the round-constant addition is replaced by the addition of a round key and of a round-tweak.

Each round-key is an element of $\mathbb{F}_p^2$ which is added to the first two branches of the internal state. The coordinates of the round-key used during Round t are obtained from a key-state $K_t \in \mathbb{F}_p$ by setting $\mathrm{rk}_{0,t} = K_t$ and $\mathrm{rk}_{1,t} = \rho\, K_t$, where ρ is defined as in Section 5.1. In order to prevent slide attacks, and more generally to ensure that the rounds are different from one another, the key state is updated by a simple affine function at each round, namely $K_{t+1} = A_t\, K_t + t$, where t is the round counter interpreted as an element of $\mathbb{F}_p$, and each A_t is a pseudo-randomly generated element of $\mathbb{F}_p$ that is part of the specification, e.g. it can be generated using the output of Shake [SHA15]. We simply set K_0 to be the master key.

The round tweak is an element of $\mathbb{F}_p$ which is added to the last branch of the internal state. It corresponds to the addition of a round constant r_t and a value derived from the master tweak T using an extendable output function H, like Shake:

$$(\mathrm{rt}_0, \ldots, \mathrm{rt}_N) = H(T)\ .$$

The way we derive elements of $\mathbb{F}_p$ from the output of Shake is much like the way the original Rescue designers derive their round constants from a short seed, simply by interpreting byte chunks as integers the same order of magnitude as p and reducing them modulo p. The backdoor is inserted in the cipher by choosing a secret master tweak $T^\star$ and computing the values $(\mathrm{rt}_0^\star, \ldots, \mathrm{rt}_N^\star) = H(T^\star)$. The round-constants are then chosen as $r_t = -\mathrm{rt}_t^\star$ for all $0 \leq t < N$.

The number of rounds is chosen following the same rules as Rescue. The first two rounds of Snare is depicted on Figure 4.

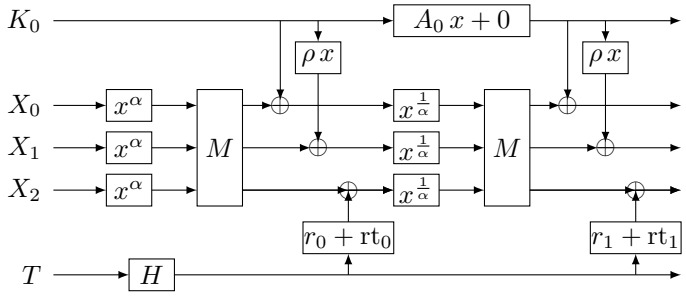


Figure 4: The first two rounds of Snare.

[11] Snare could be generalized to a larger number of blocks by adding more key additions on the additional branches, keeping the tweak addition on the last branch.

Example 2 (SNARE for $p = 18446744073709551557$)**.** This value of p gives $\alpha = 3$, and the security analysis of RESCUE implies that SNARE needs 16 rounds.

The following matrix should be used for the linear layer:

$$M = \begin{pmatrix} 6017427262211708339 & 4233710868239061116 & 255763764601897161 \\ 4380958956192272000 & 3879200506172427250 & 7786436214045931935 \\ 17417238990532664663 & 911749722830413412 & 15914259197475261016 \end{pmatrix}.$$

It was generated as explained in Section 5.1, using the code in Appendix B.1. We generated the A_t pseudo-randomly using a built-in function of `SAGE`, namely `F.random_element` with `F = GF(p)`. As the details of the A_t do not matter for our purposes, they could be generated in any other way.

Opening its Backdoor. Let $\boldsymbol{X}^t$ be the internal state of SNARE at the beginning of Round t, so that

$$\boldsymbol{X}^{t+1} = M \circ \mathcal{S}_t(\boldsymbol{X}^t) + K_t\boldsymbol{u} + (r_t + \mathrm{rt}_t)\boldsymbol{e}_3 \ ,$$

with $\boldsymbol{e}_3 = (0, 0, 1)^T$, as indeed the key injection was chosen so as to be aligned with $\boldsymbol{u}$. For the malicious tweak $T^\star$, the last term vanishes. Then, as M was chosen so as to satisfy Equation (3), we have that, for any $\boldsymbol{X}^t = x_t\boldsymbol{u}$, with $x_t \in \mathbb{F}_p$,

$$\boldsymbol{X}^{t+1} \ = \ M \circ \mathcal{S}_t(x_t\boldsymbol{u}) + K_t\boldsymbol{u} \ = \ (\mu x_t^d + K_t)\boldsymbol{u} \ , \tag{6}$$

where $d = \alpha$ if i is even, $d = 1/\alpha$ otherwise. As we can see, $\boldsymbol{X}^{t+1}$ is then also a multiple of $\boldsymbol{u}$. Indeed, a simple induction shows that if the plaintext is in $\langle\boldsymbol{u}\rangle$, then all the successive internal states of the block cipher are in the same space with probability 1, including the ciphertext. In other words, $\langle\boldsymbol{u}\rangle$ is an invariant subspace of the round function. While the existence of this invariant subspace when all $(r_t + \mathrm{rt}_t)$ vanish can be easily detected, recovering the malicious tweak $T^\star$ is difficult since it requires finding a preimage for H.

When the master tweak differs from $T^\star$, the probability that $(r_t + \mathrm{rt}_t)$ vanishes for a given round is $1/p$, assuming that rt_t is sampled from a uniform distribution over $\mathbb{F}_p$. Then, $\langle\boldsymbol{u}\rangle$ very unlikely to be an invariant subspace over N rounds when $T \neq T^\star$.

Using `SAGE`, the `get_alphas`, `rescue_XLIX_permutation` and other functions from a reference implementation[12], as well as the aforementioned matrix M, we were able to experimentally test SNARE and show that it does work as intended. In particular, we verified that the images of several multiples of $\boldsymbol{u}$ by this instance of SNARE are of the form $\mu\boldsymbol{u}$ for the malicious tweak $T^\star$, and that it is not the case when $T \neq T^\star$.

Beyond the fact that this property is a trivial distinguisher for the block cipher, it also significantly impacts key recovery. One of the main attacks targeting arithmetization-oriented primitives is based on modeling the relation between the successive internal states using non-linear equations, and then solving those using a dedicated tool (e.g. a Gröbner bases solver) in order to recover a secret key, or a preimage, etc. In practice, this attack is often the one deciding the total number of rounds. For SNARE, it is possible to greatly simplify this attack using the property highlighted in Equation (6).

Let us recall the principle of the general attack that could be applied to any variant of RESCUE. The idea is to write a set of nonlinear equations for each pair of rounds (assuming the total number of rounds is even), namely those modeling that, for even t,

$$\mathcal{S} \circ M^{-1}(\boldsymbol{X}^{t+2} - K_{t+1}\boldsymbol{u}) = M \circ \mathcal{S}(\boldsymbol{X}^t) + K_t\boldsymbol{u} \ .$$

We also need to add the equation $K_{t+1} = A_tK_t + t$ in each round in order to track the evolution of the key state. Then, we can fix $\boldsymbol{X}^0$ and $\boldsymbol{X}^N$ using a known plaintext/ciphertext pair, and solve the system in order to recover the intermediate values $\boldsymbol{X}^t$ as well as the master K_0. This basic attack should not work: the number of rounds of SNARE is chosen

[12] https://github.com/KULeuven-COSIC/Marvellous/blob/master/rescue_prime.sage

using the same rules as for Rescue, and the security analysis done by the authors of this algorithm shows that such an attack is not faster than a brute-force search for the key.

However, suppose now that an attacker aware of the backdoor uses specifically a plaintext of the form $x_0\boldsymbol{u}$. Then, they can deduce from their knowledge of the backdoor that $\boldsymbol{X}^t$ is of the form $x_t\boldsymbol{u}$ at each round, which allows a significant speed up of the attack. Indeed, instead of introducing a system with m equations and m variables modeling the internal state at each even round, we only need to introduce one for each:

$$(\mu^{-1}(x_{t+2} - K_{t+1}))^\alpha = \mu x_t^\alpha + K_t \ ,$$

and the same one as before for the key schedule. This greatly simplifies the system, and decreases its resolution time. It is actually equivalent to the previous model over a single branch, and where the linear layer M corresponds to the multiplication by a scalar μ. For an attacker unaware of the hidden structure, the system of equations will remain hidden: only people familiar with the backdoor can effectively recover the key.

For N rounds (with N even), assuming that we have access to a plaintext/ciphertext pair with a plaintext of our choice, we can solve the system by computing one of its Gröbner bases in lexicographical order, with the master key as the variable of maximum order. Then, we can extract a univariate polynomial equation verified by the master key and solve it, yielding a small number of candidates which includes the key. Replacing all of the K_t by their linear expression in K_0 yields $N/2$ equations of degree α over $N/2$ variables (the $(N/2-1)$ intermediate x_t and the master key K_0).

We were able to solve this system for small numbers of rounds, as a verification of its correctness, using a toy implementation in `SAGE`. Experimentally, it seems that the degree of regularity d_{reg} achieves Macaulay's bound (see [BFS15]). This is different from the polynomial systems studied in the original Rescue-Prime paper, most likely due to the fact that we introduce the master key as an additional variable. Using the formula in [BFS15] and Macaulay's bound for the degree of regularity, the complexity of F_5 (with Strassen's algorithm in $O(n^\omega)$ with $\omega \approx 2.807$) can be bounded by

$$O\left(\frac{N}{2}\left(1 + (\alpha-1)\frac{N}{2}\right)\binom{1+\alpha\frac{N}{2}}{\frac{N}{2}}^\omega\right) \ ,$$

and, using the probabilistic methods in [FGHR14], the complexity of FGLM with the same parameters is bounded by $O\left((N/2)\alpha^{\omega N/2}\right)$.

The complexity bounds for the more general system use the same expressions, with mN instead of N. By dividing the number of variables and equations by m, we get a time complexity that is essentially the m-th root of the one in the general attack, effectively breaking the security of a potential implementation of Snare by a wide margin, even though the expected security claims based on e.g. the wide trail strategy and the usual complexity bounds would apply.

6 Conclusions

Our analysis shows that monomial-based Sboxes over large finite fields may introduce weaknesses due to the existence of chains of affine subspaces of dimension 1 through the primitive, for some round-constants (or round-keys). Such chains lead to some abnormal behaviour, including a very high differential uniformity for some fixed keys. These weak keys (or weak round-constants) may also be chosen intentionally, opening the path towards backdoored primitives. This points out that the arguments currently used for assessing the security of the recently proposed arithmetization-oriented primitives are not enough, and that a more in-depth analysis is required.

These issues are especially worrying in the case of arithmetization-oriented primitives operating on prime fields. Over binary fields, a classical method to avoid such threats consists in composing a monomial Sbox with an $\mathbb{F}_2$-affine transformation, like in the AES. But this simple technique cannot be used anymore when the Sbox operates on a prime field. Monomial transformations are then the only known general families of Sboxes over $\mathbb{F}_p$ having low differential uniformity and linearity, and good performance. Therefore, the search for good Sboxes over $\mathbb{F}_p$ with a more complex univariate representation is an interesting direction which would offer some better choices to the designers of arithmetization-oriented primitives and avoid the risk related to weak round-constants.

Another conclusion we draw from our results is the importance of limiting the freedom of the implementers of such primitives. Indeed, as the latter might be tempted to tweak the designs to suit their specific implementation constraints, authors must clearly specify which changes are safe and which are not.

Acknowledgments

This work is partially supported by the European Research Council (ERC, grant agreement no. 101041545 "ReSCALE") and by ANR (French National Research Agency) grant ANR-21-CE39-0012 (SWAP).

References

[AAB+20] Abdelrahaman Aly, Tomer Ashur, Eli Ben-Sasson, Siemen Dhooghe, and Alan Szepieniec. Design of symmetric-key primitives for advanced cryptographic protocols. *IACR Trans. Symm. Cryptol.*, 2020(3):1–45, 2020.

[ACG+19] Martin R. Albrecht, Carlos Cid, Lorenzo Grassi, Dmitry Khovratovich, Reinhard Lüftenegger, Christian Rechberger, and Markus Schofnegger. Algebraic cryptanalysis of STARK-friendly designs: Application to MARVELlous and MiMC. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part III*, volume 11923 of *LNCS*, pages 371–397. Springer, Heidelberg, December 2019.

[ADK+14] Martin R. Albrecht, Benedikt Driessen, Elif Bilge Kavun, Gregor Leander, Christof Paar, and Tolga Yalçin. Block ciphers - focus on the linear layer (feat. PRIDE). In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 57–76. Springer, Heidelberg, August 2014.

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

[AGP+19] Martin R. Albrecht, Lorenzo Grassi, Léo Perrin, Sebastian Ramacher, Christian Rechberger, Dragos Rotaru, Arnab Roy, and Markus Schofnegger. Feistel structures for MPC, and more. In Kazue Sako, Steve Schneider, and Peter Y. A. Ryan, editors, *ESORICS 2019, Part II*, volume 11736 of *LNCS*, pages 151–171. Springer, Heidelberg, September 2019.

[AGR+16] Martin R. Albrecht, Lorenzo Grassi, Christian Rechberger, Arnab Roy, and Tyge Tiessen. MiMC: Efficient encryption and cryptographic hashing with minimal multiplicative complexity. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part I*, volume 10031 of *LNCS*, pages 191–219. Springer, Heidelberg, December 2016.

[AKM+22] Tomer Ashur, Al Kindi, Willi Meier, Alan Szepieniec, and Bobbin Threadbare. Rescue-Prime Optimized. Cryptology ePrint Archive, Report 2022/1577, 2022. https://eprint.iacr.org/2022/1577.

[BBC+23] Clémence Bouvier, Pierre Briaud, Pyrros Chaidos, Léo Perrin, Robin Salen, Vesselin Velichkov, and Danny Willems. New design techniques for efficient arithmetization-oriented hash functions: Anemoi permutations and Jive compression mode. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part III*, volume 14083 of *LNCS*, pages 507–539. Springer, Heidelberg, August 2023.

[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, *CRYPTO 2022, Part III*, volume 13509 of *LNCS*, pages 748–778. Springer, Heidelberg, August 2022.

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Scalable, transparent, and post-quantum secure computational integrity. Cryptology ePrint Archive, Report 2018/046, 2018. https://eprint.iacr.org/2018/046.

[BBLP22] Augustin Bariant, Clémence Bouvier, Gaëtan Leurent, and Léo Perrin. Algebraic attacks against some arithmetization-oriented primitives. *IACR Trans. Symm. Cryptol.*, 2022(3):73–101, 2022.

[BCD+20] Tim Beyne, Anne Canteaut, Itai Dinur, Maria Eichlseder, Gregor Leander, Gaëtan Leurent, María Naya-Plasencia, Léo Perrin, Yu Sasaki, Yosuke Todo, and Friedrich Wiemer. Out of oddity - new cryptanalytic techniques against symmetric primitives optimized for integrity proof systems. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part III*, volume 12172 of *LNCS*, pages 299–328. Springer, Heidelberg, August 2020.

[BCL+20] Tim Beyne, Anne Canteaut, Gregor Leander, María Naya-Plasencia, Léo Perrin, and Friedrich Wiemer. On the security of the rescue hash function. Cryptology ePrint Archive, Report 2020/820, 2020. https://eprint.iacr.org/2020/820.

[BCLR17] Christof Beierle, Anne Canteaut, Gregor Leander, and Yann Rotella. Proving resistance against invariant attacks: How to choose the round constants. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 647–678. Springer, Heidelberg, August 2017.

[BCP23] Clémence Bouvier, Anne Canteaut, and Léo Perrin. On the algebraic degree of iterated power functions. *Des. Codes Cryptogr.*, 91(3):997–1033, 2023.

[BDPA11] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Cryptographic sponge functions. https://keccak.team/files/CSF-0.1.pdf, 2011.

[BFS15] Magali Bardet, Jean-Charles Faugère, and Bruno Salvy. On the complexity of the F5 Gröbner basis algorithm. *Journal of Symbolic Computation*, 70:49–70, 2015.

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

[CDL16] Anne Canteaut, Sébastien Duval, and Gaëtan Leurent. Construction of lightweight S-boxes using Feistel and MISTY structures. In Orr Dunkelman and Liam Keliher, editors, *SAC 2015*, volume 9566 of *LNCS*, pages 373–393. Springer, Heidelberg, August 2016.

[CR15] Anne Canteaut and Joëlle Roué. On the behaviors of affine equivalent sboxes regarding differential and linear attacks. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 45–74. Springer, Heidelberg, April 2015.

[DGGK21] Christoph Dobraunig, Lorenzo Grassi, Anna Guinet, and Daniël Kuijsters. Ciminion: Symmetric encryption based on Toffoli-gates over large finite fields. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part II*, volume 12697 of *LNCS*, pages 3–34. Springer, Heidelberg, October 2021.

[DR02] Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Springer, 2002.

[DR05] Joan Daemen and Vincent Rijmen. Probability distributions of correlation and differentials in block ciphers. Cryptology ePrint Archive, Report 2005/212, 2005. https://eprint.iacr.org/2005/212.

[DR07] Joan Daemen and Vincent Rijmen. Probability distributions of correlation and differentials in block ciphers. *J. Math. Cryptol.*, 1(3):221–242, 2007.

[EGL+20] Maria Eichlseder, Lorenzo Grassi, Reinhard Lüftenegger, Morten Øygarden, Christian Rechberger, Markus Schofnegger, and Qingju Wang. An algebraic attack on ciphers with low-degree round functions: Application to full MiMC. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part I*, volume 12491 of *LNCS*, pages 477–506. Springer, Heidelberg, December 2020.

[Eth21] Ethereum Foundation. ZK hash function cryptanalysis bounties 2021. Available online: https://www.zkhashbounties.info/, 2021.

[FGHR14] Jean-Charles Faugère, Pierrick Gaudry, Louise Huot, and Guénaël Renault. Sub-cubic change of ordering for Gröbner basis: a probabilistic approach. In *International Symposium on Symbolic and Algebraic Computation - ISSAC 2014*, pages 170–177, 2014.

[GHR+23] Lorenzo Grassi, Yonglin Hao, Christian Rechberger, Markus Schofnegger, Roman Walch, and Qingju Wang. Horst meets Fluid-SPN: Griffin for zero-knowledge applications. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part III*, volume 14083 of *LNCS*, pages 573–606. Springer, Heidelberg, August 2023.

[GJN+16] Jian Guo, Jérémy Jean, Ivica Nikolic, Kexin Qiao, Yu Sasaki, and Siang Meng Sim. Invariant subspace attack against Midori64 and the resistance criteria for S-box designs. *IACR Trans. Symm. Cryptol.*, 2016(1):33–56, 2016. https://tosc.iacr.org/index.php/ToSC/article/view/534.

[GKL+22] Lorenzo Grassi, Dmitry Khovratovich, Reinhard Lüftenegger, Christian Rechberger, Markus Schofnegger, and Roman Walch. Reinforced concrete: A fast hash function for verifiable computation. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *ACM CCS 2022*, pages 1323–1335. ACM Press, November 2022.

[GKR+21] Lorenzo Grassi, Dmitry Khovratovich, Christian Rechberger, Arnab Roy, and Markus Schofnegger. Poseidon: A new hash function for zero-knowledge proof systems. In Michael Bailey and Rachel Greenstadt, editors, *USENIX Security 2021*, pages 519–535. USENIX Association, August 2021.

[GRR16] Lorenzo Grassi, Christian Rechberger, and Sondre Rønjom. Subspace trail cryptanalysis and its applications to AES. *IACR Trans. Symm. Cryptol.*, 2016(2):192–225, 2016. https://tosc.iacr.org/index.php/ToSC/article/view/571.

[GW20] Ariel Gabizon and Zachary J. Williamson. plookup: A simplified polynomial protocol for lookup tables. Cryptology ePrint Archive, Report 2020/315, 2020. https://eprint.iacr.org/2020/315.

[HLL+01] Seokhie Hong, Sangjin Lee, Jongin Lim, Jaechul Sung, Dong Hyeon Cheon, and Inho Cho. Provable security against differential and linear cryptanalysis for the SPN structure. In Bruce Schneier, editor, *FSE 2000*, volume 1978 of *LNCS*, pages 273–283. Springer, Heidelberg, April 2001.

[HO99] Philip Hawkes and Luke O'Connor. XOR and non-XOR differential probabilities. In Jacques Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 272–285. Springer, Heidelberg, May 1999.

[KR21] Nathan Keller and Asaf Rosemarin. Mind the middle layer: The HADES design strategy revisited. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part II*, volume 12697 of *LNCS*, pages 35–63. Springer, Heidelberg, October 2021.

[LAW+23] Fukang Liu, Ravi Anand, Libo Wang, Willi Meier, and Takanori Isobe. Coefficient grouping: Breaking chaghri and more. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part IV*, volume 14007 of *LNCS*, pages 287–317. Springer, Heidelberg, April 2023.

[LMM91] Xuejia Lai, James L. Massey, and Sean Murphy. Markov ciphers and differential cryptanalysis. In Donald W. Davies, editor, *EUROCRYPT'91*, volume 547 of *LNCS*, pages 17–38. Springer, Heidelberg, April 1991.

[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, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 254–283. Springer, Heidelberg, April 2015.

[LTW18] Gregor Leander, Cihangir Tezcan, and Friedrich Wiemer. Searching for subspace trails and truncated differentials. *IACR Trans. Symm. Cryptol.*, 2018(1):74–100, 2018.

[MMMS23] Loïc Masure, Pierrick Méaux, Thorben Moos, and François-Xavier Standaert. Effective and efficient masking with low noise using small-Mersenne-prime ciphers. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part IV*, volume 14007 of *LNCS*, pages 596–627. Springer, Heidelberg, April 2023.

[Nyb94] Kaisa Nyberg. Differentially uniform mappings for cryptography. In Tor Helleseth, editor, *EUROCRYPT'93*, volume 765 of *LNCS*, pages 55–64. Springer, Heidelberg, May 1994.

[O'C94] Luke O'Connor. On the distribution of characteristics in bijective mappings. In Tor Helleseth, editor, *EUROCRYPT'93*, volume 765 of *LNCS*, pages 360–370. Springer, Heidelberg, May 1994.

[PSLL03] Sangwoo Park, Soo Hak Sung, Sangjin Lee, and Jongin Lim. Improving the upper bound on the maximum differential and the maximum linear Hull probability for SPN structures and AES. In Thomas Johansson, editor, *FSE 2003*, volume 2887 of *LNCS*, pages 247–260. Springer, Heidelberg, February 2003.

[PW20] Thomas Peyrin and Haoyang Wang. The MALICIOUS framework: Embedding backdoors into tweakable block ciphers. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part III*, volume 12172 of *LNCS*, pages 249–278. Springer, Heidelberg, August 2020.

[SAD20] Alan Szepieniec, Tomer Ashur, and Siemen Dhooghe. Rescue-prime: a standard specification (SoK). Cryptology ePrint Archive, Report 2020/1143, 2020. https://eprint.iacr.org/2020/1143.

[SHA15] SHA-3 Standard: Permutation-based hash and extendable-output functions. National Institute of Standards and Technology, NIST FIPS PUB 202, U.S. Department of Commerce, August 2015.

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

A Proportion of weak round-constants

The number of sequences of round-constants $(\boldsymbol{r}_0, \ldots, \boldsymbol{r}_{N-1})$ for which a given subspace $(\boldsymbol{a} + \langle \boldsymbol{v} \rangle)$ propagates through N rounds as described in Theorem 2 is

$$q^m \prod_{t=1}^{N-1} q^{m-w_t+1} \text{ where } w_t = wt(v_t) \, .$$

It follows that the proportion of such sequences is $q^{-\sum_{t=1}^{N-1} w_t + N - 1}$. When the linear layers are MDS (i.e. $w_t + w_{t+1} \geq m + 1$), this proportion is at most

$$\begin{cases} q^{-(m-1)\frac{N-1}{2}} & \text{for } N \text{ odd} \\ q^{-(m-1)\frac{N-2}{2}} & \text{for } N \text{ even.} \end{cases}$$

The number of distinct separable affine subspaces of dimension 1 is given by the following lemma.

Lemma 3. *For any $m \geq 2$ and prime power q, the number of distinct separable affine spaces of dimension 1 in $\mathbb{F}_q^m$ is:*

$$A_{q,m} = \frac{1}{q-1}\left((2q-1)^m - q^m\right) \underset{q \to \infty}{\sim} (2^m - 1)q^{m-1} \, .$$

Proof. We can enumerate them according to a parameter w varying from 1 to m that represents the size of $\text{supp}(\boldsymbol{v})$. For a fixed w, there are $\binom{m}{w}$ possible supports. Using the canonical representation and the fact that we focus on separable affine subspaces, we know that the first nonzero coordinate of $\boldsymbol{v}$ equals 1 and all coordinates of $\boldsymbol{a}$ in $\text{supp}(\boldsymbol{v})$ to 0.

Therefore, $(q-1)$ choices remain for each of the $(w-1)$ other coordinates of $\boldsymbol{v}$ and q choices remain for each of the $(m-w)$ other coordinates of $\boldsymbol{a}$. Hence:

$$\begin{aligned} A_{q,m} &= \sum_{w=1}^{m} \binom{m}{w} (q-1)^{w-1} q^{m-w} \\ &= \frac{1}{q-1} \left(\sum_{w=0}^{m} \binom{m}{w} (q-1)^{w} q^{m-w} \right) - \frac{q^m}{q-1} \\ &= \frac{1}{q-1} \left((2q-1)^m - q^m \right) \underset{q \to \infty}{\sim} (2^m - 1) q^{m-1}. \end{aligned}$$

□

Then, the probability that a sequence of round-constants is such that there exists a separable affine subspace that propagates through N rounds as described in Theorem 2, can be approximated by

$$(2^m - 1) q^{-(m-1)\lfloor \frac{N-3}{2} \rfloor},$$

which is marginal for practical values of q and N. As a consequence, round constants picked randomly can be safely expected to effectively thwart such patterns.

B Code Snippets

B.1 Matrix Generation.

The following `SAGE` code to generates matrices that can be used as linear layers for Snare. For brevity, and due to the extremely low probability of collision, we do not test whether all coefficients are distinct, but a more thorough program should.

```
p = 18446744073709551557    # big prime
F = GF(p)
m = 3
alpha = 3

# Choose some rho that is a gcd(alpha**2 - 1, p - 1)-th root of unity.
# rho can be equal to -1, but not to 1.
rho = F.zeta(gcd(alpha ** 2 - 1, p - 1))

Pol_ring.<x_pol> = F[]

"""
GENERATION OF M
"""

# M is a Cauchy matrix. We must generate the vectors x and y of size m.
# y is arbitrarily generated to reduce the number of degrees of freedom
y = []
for i in range(m):
    y.append(F.random_element())

# Define equations for the coefficients of M
# M must verify, for some random mu:
# M * (1, rho ** alpha, 0) = mu * (1, rho, 0)

mu = F.random_element()
R_0 = Pol_ring((x_pol - y[1]) + (x_pol - y[0]) * (rho ** alpha) - mu * (x_pol - y[1]) * (x_pol - y[0]))
R_1 = Pol_ring((x_pol - y[1]) + (x_pol - y[0]) * (rho ** alpha) - mu * rho * (x_pol - y[1]) * (x_pol - y[0]))
R_2 = Pol_ring((x_pol - y[1]) + (x_pol - y[0]) * (rho ** alpha))
while ((not R_0.roots()) or (not R_1.roots()) or (not R_2.roots())):
    mu = F.random_element()
    R_0 = Pol_ring((x_pol - y[1]) + (x_pol - y[0]) * (rho ** alpha) - mu * (x_pol - y[1]) * (x_pol - y[0]))
    R_1 = Pol_ring((x_pol - y[1]) + (x_pol - y[0]) * (rho ** alpha) - mu * rho * (x_pol - y[1]) * (x_pol - y[0]))
    R_2 = Pol_ring((x_pol - y[1]) + (x_pol - y[0]) * (rho ** alpha))

# Solve equations
x_0 = R_0.roots()[0][0]
x_1 = R_1.roots()[0][0]
x_2 = R_2.roots()[0][0]

x = [x_0,x_1,x_2]

M = matrix(F, m, m)

for i in range(m):
    for j in range(m):
        M[i,j] = 1/(x[i] - y[j])
```

C Differential Properties of Random Functions

In order to investigate the differential properties of functions of $\mathbb{F}_q^n$, we must first establish a baseline: what is the expected differential behaviour of a random function of this set?

The distribution of the coefficients in the DDT of a permutation on $\mathbb{F}_2^n$ is well-known, it has for instance been studied in [O'C94, DR07, DR05]. Each entry in the DDT can be approximated by a random variable, all of them being independent and identically distributed. They follow a Poisson distribution with parameter 2^{-1} and the expected value of the maximal coefficient is $2n$.

However, due to their much rarer relevance, differences defined in arbitrary groups, different from $(\mathbb{F}_2^n, +)$, have received less attention. To the best of our knowledge, the only paper that deals with such quantities is [HO99]. While its authors consider differentials for the modular addition, they unfortunately focus on $\mathbb{Z}/2^n\mathbb{Z}$, which is not the type of ring we consider. Nevertheless, their main result is that the DDT coefficients of a permutation in such a ring behave like independent and identically distributed random variables that follow a Poisson distribution with parameter 1.

The following conjecture argues that the situation is the same in the case of $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}$ and any finite field, as soon as the characteristic is not 2.

Conjecture 1. *Let $\mathbb{F}_q$ be a finite field of characteristic $p > 2$ and F be a permutation of $\mathbb{F}_q$ picked uniformly at random. Its DDT coefficients that correspond to non-zero input differences can be accurately modeled as independent and identically distributed variables following a Poisson distribution with parameter 1, with the caveat that $\mathsf{DDT}_F(\alpha, \beta) = \mathsf{DDT}_F(-\alpha, -\beta)$.*

This conjecture follows simply from modeling $\Delta_a F : x \mapsto F(x+a) - F(x)$ as a random function of $\mathbb{F}_q$ and from remarking that $\Delta_a F(x) = -\Delta_{-a} F(x+a)$ As the independence of the variables cannot be proved, we cannot call this result a theorem and stick with "conjecture". Nevertheless, it is backed by our experiments (see Appendix D.1).

A consequence of Conjecture 1 is that the maximum coefficient of the DDT is the maximum out of about $(q-1)^2/2$ independent Poisson variables. Then, the bound established in [HO99] is easily adapted in $\mathbb{F}_q$ as follows.

Corollary 3. *Let $\mathbb{F}_q$ be a finite field of characteristic $p > 2$. The probability that the differential uniformity of a permutation of $\mathbb{F}_q^m$ is upper bounded by*

$$B(q,m) = \frac{2\ln\left(\frac{(q^m-1)^2}{2}\right)}{\ln\left(\ln\left(\frac{(q^m-1)^2}{2}\right)\right)}$$

converges to 1 as q^m increases, where ln *is the Neperian logarithm.*

This quantity only depends on q^m, the size of the set on which the permutation operates. Most notably, it takes the same value for permutations of $\mathbb{F}_q^m$ and permutations of $\mathbb{F}_{q^m}$.

D Experimental Verifications

D.1 Conjecture 1.

Conjecture 1 states that the entries of the DDT of a random permutation follow a Poisson distribution. In order to test it, we have computed the differential spectra of multiple permutations picked uniformly at random for $q = p^2$ and for various values of p. The result is given in Figure 5. As we can see, even for small values of p, the entries do behave like independent and identically distributed random variables following a Poisson distribution with parameter 1.

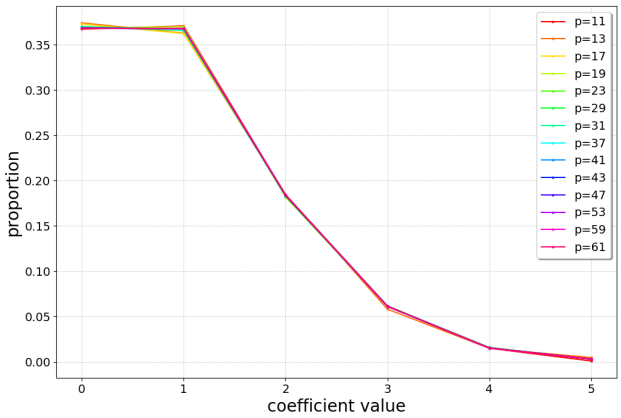


Figure 5: Distribution of the DDT coefficients of some permutations picked uniformly at random.

D.2 Corollary 3.

Corollary 3, based on Conjecture 1 is all the more plausible in light of our experimental results detailed in Table 1: this table compares the bound from Corollary 3 and the exact differential uniformities for 100 randomly generated permutations of $\mathbb{F}_p^2$ where p takes each prime value between 2^3 and 2^5.

Table 1: Differential uniformities of pseudo-randomly generated permutations of $\mathbb{F}_p^2$.

	# permutations with a given δ								
p	5	6	7	8	9	10	11	12	$B(p,2)$
11	1	50	42	6	1	–	–	–	8.13
13	–	36	50	10	4	–	–	–	8.47
17	–	3	56	34	6	1	–	–	9.00
19	–	–	55	40	3	2	–	–	9.21
23	–	–	27	59	13	1	–	–	9.58
29	–	–	4	57	38	1	–	1	10.03
31	–	–	–	61	35	4	–	–	10.16

E Chaining subspaces of higher dimension

As previously mentioned, it is rather unlikely that the image of a subspace of dimension 2 or more is separable. The following proposition shows, as a special case, that this cannot occur when the linear layer is MDS if

$$F(\boldsymbol{a}) + \langle M\mathcal{S}(\boldsymbol{v}_1), \ldots, M\mathcal{S}(\boldsymbol{v}_d)\rangle$$

is the canonical representation of $F(V)$.

Proposition 5. *Let $\mathcal{S}$ be a substitution layer corresponding to m copies of an Sbox S over $\mathbb{F}_q$, and M be an MDS matrix over $\mathbb{F}_q$. For any $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ in $\mathbb{F}_q^m$ such that* $\mathrm{supp}(\boldsymbol{v}_1) \cap \mathrm{supp}(\boldsymbol{v}_2) = \emptyset$*, we have*

$$\mathrm{supp}\left(M\mathcal{S}(\boldsymbol{v}_1)\right) \cap \mathrm{supp}\left(M\mathcal{S}(\boldsymbol{v}_2)\right) \neq \emptyset\,.$$

Proof. Suppose that $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ have disjoint supports. It obviously follows that

$$\mathsf{wt}(\boldsymbol{v}_1) + \mathsf{wt}(\boldsymbol{v}_2) \leq m \ .$$

If we also had $\mathrm{supp}(M\mathcal{S}(\boldsymbol{v}_1)) \cap \mathrm{supp}(M\mathcal{S}(\boldsymbol{v}_2)) = \emptyset$, then likewise it would hold that $\mathsf{wt}(M\mathcal{S}(\boldsymbol{v}_1)) + \mathsf{wt}(M\mathcal{S}(\boldsymbol{v}_2)) \leq m$, hence by adding together these inequalities:

$$\mathsf{wt}(\boldsymbol{v}_1) + \mathsf{wt}(M\mathcal{S}(\boldsymbol{v}_1)) + \mathsf{wt}(\boldsymbol{v}_2) + \mathsf{wt}(M\mathcal{S}(\boldsymbol{v}_2)) \leq 2m \ . \tag{7}$$

However, since M is MDS, by definition, for any $\boldsymbol{x} \in \mathbb{F}_q^m$, we have that

$$\mathsf{wt}(\boldsymbol{x}) + \mathsf{wt}(M\boldsymbol{x}) \geq m + 1 \ .$$

By applying this to $\boldsymbol{x} = \mathcal{S}(\boldsymbol{v}_i)$, we get that, for $i \in \{1, 2\}$,

$$\mathsf{wt}(\mathcal{S}(\boldsymbol{v}_i)) + \mathsf{wt}(M\mathcal{S}(\boldsymbol{v}_i)) = \mathsf{wt}(\boldsymbol{v}_i) + \mathsf{wt}(M\mathcal{S}(\boldsymbol{v}_i)) \geq m + 1 \ ,$$

which contradicts (7). □

It may obviously happen that

$$F(\boldsymbol{a}) + \langle M\mathcal{S}(\boldsymbol{v}_1), \ldots, M\mathcal{S}(\boldsymbol{v}_d) \rangle$$

is not the canonical representation of $F(V)$, and that the fact that $F(V)$ is separable can be deduced by considering other basis vector. However, finding an explicit formula generalizing the one-dimensional case seems very difficult, and an algorithmic approach seems more promising.