Exploring the Set of APN Functions in Practice

Jules Baudrin[1], Pierre Galissant[2], Léo Perrin [2]

Université Versailles Saint-Quentin, Versailles, France
Inria, Paris, France

February 13, 2026

European Research Council
Established by the European Commission

informatics mathematics

UVSQ
université PARIS-SACLAY

What Am I Talking about ?

Boolean Functions

Topic dear to Pascale, Anne, Léo, Shibam, Merlin and me

What Am I Talking about ?

Boolean Functions

Topic dear to Pascale, Anne, Léo, Shibam, Merlin and me

APN Functions

Related to cryptography, more on that in a few minutes

What Am I Talking about ?

Boolean Functions

Topic dear to Pascale, Anne, Léo, Shibam, Merlin and me

APN Functions

Related to cryptography, more on that in a few minutes

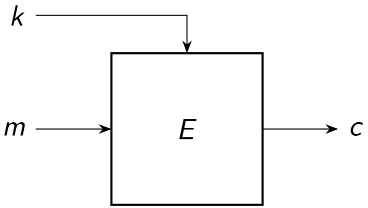
SboxU: A library for S-box analysis

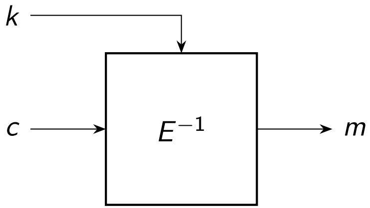
Important development project, with many team members involved, more on that throughout the presentation and a proper advert at the end

Definition of Block Ciphers

Block Cipher

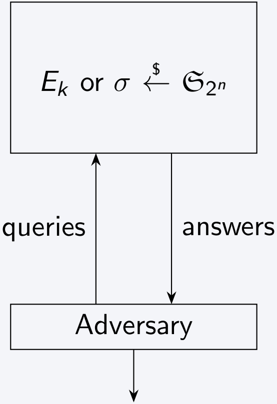
A family of keyed bijections $(E_k)_{k \in \mathcal{K}}$, $E_k : \mathbb{F}_2^n \to \mathbb{F}_2^n$





Ideal Security: Indistinguishability

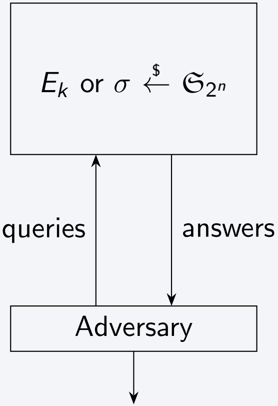
The Indistinguishability Game



Was it drawn from $(E_k)_{k \in \mathcal{K}}$?

Ideal Security: Indistinguishability

The Indistinguishability Game



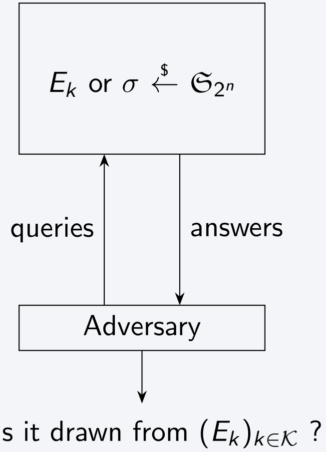
Was it drawn from $(E_k)_{k \in \mathcal{K}}$?

Why ?

- To build security proof on top of block-ciphers
- To avoid stronger attacks like key recovery

Ideal Security: Indistinguishability

The Indistinguishability Game



Why ?

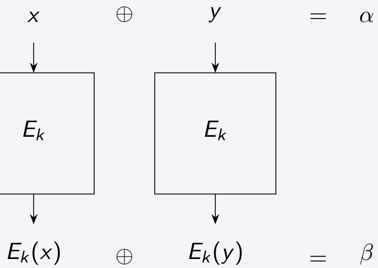
- To build security proof on top of block-ciphers
- To avoid stronger attacks like key recovery

How to find a Distinguisher

What property would help to win the indistinguishability game ?

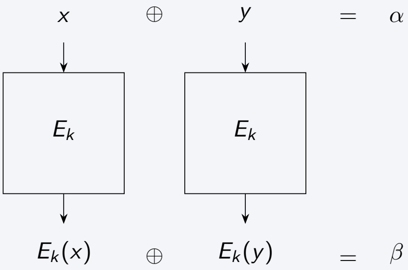
Differential Cryptanalysis

Principle (Biham and Shamir 91)



Differential Cryptanalysis

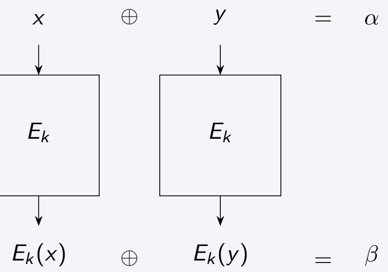
Principle (Biham and Shamir 91)



$\hookrightarrow$ How does the number of solutions $\#\{x \in \mathbb{F}_2^n \mid E_k(x + \alpha) + E_k(x) = \beta\}$ varies ?

Differential Cryptanalysis

Principle (Biham and Shamir 91)



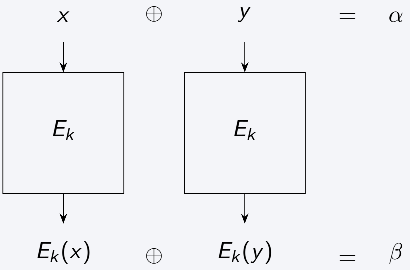
$\hookrightarrow$ How does the number of solutions $\#\{x \in \mathbb{F}_2^n \mid E_k(x+\alpha) + E_k(x) = \beta\}$ varies ?

In average over all random bijections

For $\alpha \in \mathbb{F}_2^n \setminus \{0\}$ and $\beta \in \mathbb{F}_2^n$,
$F(x+\alpha) + F(x) = \beta$ has 1 solution in x

Differential Cryptanalysis

Principle (Biham and Shamir 91)



$\hookrightarrow$ How does the number of solutions $\#\{x \in \mathbb{F}_2^n \mid E_k(x+\alpha) + E_k(x) = \beta\}$ varies ?

In average over all random bijections

For $\alpha \in \mathbb{F}_2^n \setminus \{0\}$ and $\beta \in \mathbb{F}_2^n$,
$F(x+\alpha) + F(x) = \beta$ has 1 solution in x

Differential Distinguisher

A pair $(\alpha, \beta) \in \mathbb{F}_2^n \setminus \{0\} \times \mathbb{F}_2^n$, such that $E_k(x+\alpha) + E_k(x) = \beta$ has many solution in x

How to resist Differential Cryptanalysis ?

We start from the component level:

How to resist Differential Cryptanalysis ?

We start from the component level:

- We have a better understanding of functions on small dimensions

How to resist Differential Cryptanalysis ?

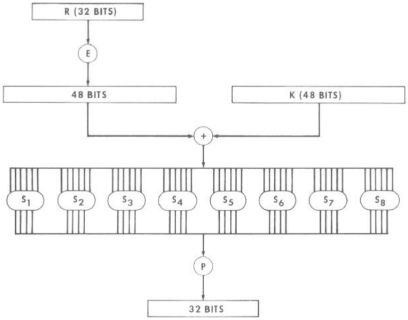
We start from the component level:

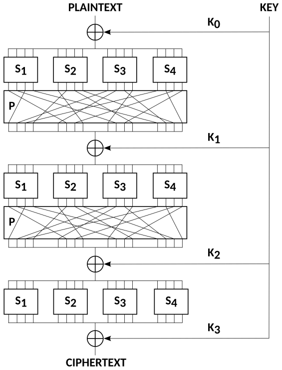
- We have a better understanding of functions on small dimensions
- We build the rest on top of 'nice' small components

How to resist Differential Cryptanalysis ?

We start from the component level:

- We have a better understanding of functions on small dimensions
- We build the rest on top of 'nice' small components





APN Functions

Definition (APN Function)

A function $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ is *Almost Perfect Non-linear* (APN) if $F(x + a) + F(x) = b$ has at most 2 solutions for all $a \neq 0, b$.

APN Functions

Definition (APN Function)

A function $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ is *Almost Perfect Non-linear* (APN) if $F(x + a) + F(x) = b$ has at most 2 solutions for all $a \neq 0, b$.

- **Perfect Nonlinear**: for linear functions, there are 0 or 2^n solutions
- **Almost**: in characteristic 2, we have an even number of solutions
- Example: $F(x) = x^3$ is APN for any n

APN Functions

Definition (APN Function)

A function $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ is *Almost Perfect Non-linear* (APN) if $F(x + a) + F(x) = b$ has at most 2 solutions for all $a \neq 0, b$.

- **Perfect Nonlinear**: for linear functions, there are 0 or 2^n solutions
- **Almost**: in characteristic 2, we have an even number of solutions
- Example: $F(x) = x^3$ is APN for any n

We need more properties for cryptography

- Sometimes, a S-box needs to be bijective
- It is usually better for it to have a higher degree

The Big APN Problem

Dillon Permutation $(n = 6)$

```
[ 0 54 48 13 15 18 53 35]
[25 63 45 52  3 20 41 33]
[59 36  2 34 10  8 57 37]
[60 19 42 14 50 26 58 24]
[39 27 21 17 16 29  1 62]
[47 40 51 56  7 43 44 38]
[31 11  4 28 61 46  5 49]
[ 9  6 23 32 30 12 55 22]
```

The Big APN Problem

Dillon Permutation ($n = 6$)

0	54	48	13	15	18	53	35
25	63	45	52	3	20	41	33
59	36	2	34	10	8	57	37
60	19	42	14	50	26	58	24
39	27	21	17	16	29	1	62
47	40	51	56	7	43	44	38
31	11	4	28	61	46	5	49
9	6	23	32	30	12	55	22

- It is the only known APN permutation since 2009

The Big APN Problem

Dillon Permutation (n = 6)

0	54	48	13	15	18	53	35
25	63	45	52	3	20	41	33
59	36	2	34	10	8	57	37
60	19	42	14	50	26	58	24
39	27	21	17	16	29	1	62
47	40	51	56	7	43	44	38
31	11	4	28	61	46	5	49
9	6	23	32	30	12	55	22

- It is the only known APN permutation since 2009
- It was found *exploring* the set of APN functions starting from the Kim mapping:

$$x^3 + x^{10} + u\,x^{24}, \text{with } \langle u \rangle = \mathbb{F}_2^n \setminus \{0\}$$

The Big APN Problem

Dillon Permutation (n = 6)

0	54	48	13	15	18	53	35
25	63	45	52	3	20	41	33
59	36	2	34	10	8	57	37
60	19	42	14	50	26	58	24
39	27	21	17	16	29	1	62
47	40	51	56	7	43	44	38
31	11	4	28	61	46	5	49
9	6	23	32	30	12	55	22

- It is the only known APN permutation since 2009
- It was found *exploring* the set of APN functions starting from the Kim mapping:

$$x^3 + x^{10} + u\,x^{24}, \text{with } \langle u \rangle = \mathbb{F}_2^n \setminus \{0\}$$

- The studied generalizations of the Kim mapping don't work

The Big APN Problem

Dillon Permutation (n = 6)

```
[ 0 54 48 13 15 18 53 35]
[25 63 45 52  3 20 41 33]
[59 36  2 34 10  8 57 37]
[60 19 42 14 50 26 58 24]
[39 27 21 17 16 29  1 62]
[47 40 51 56  7 43 44 38]
[31 11  4 28 61 46  5 49]
[ 9  6 23 32 30 12 55 22]
```

- It is the only known APN permutation since 2009
- It was found *exploring* the set of APN functions starting from the Kim mapping:

$$x^3 + x^{10} + u\, x^{24}, \text{with } \langle u \rangle = \mathbb{F}_2^n \setminus \{0\}$$

- The studied generalizations of the Kim mapping don't work

The Big APN Problem

Are there APN bijections when n is even and $n > 6$?

- x^3 is a bijection when n is odd

Goal

Goal

Can we replicate the results of Dillon in dimension 8 by *exploring* the set of APN functions ?

Goal

Can we replicate the results of Dillon in dimension 8 by *exploring* the set of APN functions ?

$\hookrightarrow$ What do we mean by exploration :

1. The computation of the **CCZ class**
2. The computation of **Switching Neighbours**

Definitions

$F : \mathbb{F}_2^n \to \mathbb{F}_2^n$, $f : \mathbb{F}_2^n \to \mathbb{F}_2$

- **Coordinates** F_i : $F = (F_1, ..., F_n)$

Definitions

$F : \mathbb{F}_2^n \to \mathbb{F}_2^n$, $f : \mathbb{F}_2^n \to \mathbb{F}_2$

- **Coordinates** F_i : $F = (F_1, ..., F_n)$
- **Degree of a Boolean Function** f : The degree of f as a polynomial over $\mathbb{F}_2[x_1, ..., x_n]/(x_1^2 + x_1, ..., x_n^2 + x_n)$

Definitions

$F : \mathbb{F}_2^n \to \mathbb{F}_2^n$, $f : \mathbb{F}_2^n \to \mathbb{F}_2$

- **Coordinates** F_i : $F = (F_1, ..., F_n)$
- **Degree of a Boolean Function** f : The degree of f as a polynomial over $\mathbb{F}_2[x_1, ..., x_n]/(x_1^2 + x_1, ..., x_n^2 + x_n)$
- **Univariate Degree of** F : The degree of F as a polynomial over $\mathbb{F}_{2^n}[X]$

Definitions

$F : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$, $f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2$

- **Coordinates** F_i : $F = (F_1, ..., F_n)$
- **Degree of a Boolean Function** f : The degree of f as a polynomial over $\mathbb{F}_2[x_1, ..., x_n]/(x_1^2 + x_1, ..., x_n^2 + x_n)$
- **Univariate Degree of** F : The degree of F as a polynomial over $\mathbb{F}_{2^n}[X]$
- **Algebraic Degree** F : The maximum of the degrees of the coordinates of F

Definitions

$F : \mathbb{F}_2^n \to \mathbb{F}_2^n$, $f : \mathbb{F}_2^n \to \mathbb{F}_2$

- **Coordinates** F_i : $F = (F_1, ..., F_n)$
- **Degree of a Boolean Function** f : The degree of f as a polynomial over $\mathbb{F}_2[x_1, ..., x_n]/(x_1^2 + x_1, ..., x_n^2 + x_n)$
- **Univariate Degree of** F : The degree of F as a polynomial over $\mathbb{F}_{2^n}[X]$
- **Algebraic Degree** F : The maximum of the degrees of the coordinates of F
- **Quadratic Function**: F is quadratic if its algebraic degree is 2
- **Graph of a Function**: $\Gamma_F = \{(x, F(x)) \,|\, x \in \mathbb{F}_2^n\}$

Definitions

$F : \mathbb{F}_2^n \to \mathbb{F}_2^n$, $f : \mathbb{F}_2^n \to \mathbb{F}_2$

- **Coordinates** F_i : $F = (F_1, ..., F_n)$
- **Degree of a Boolean Function** f : The degree of f as a polynomial over $\mathbb{F}_2[x_1, ..., x_n]/(x_1^2 + x_1, ..., x_n^2 + x_n)$
- **Univariate Degree of** F : The degree of F as a polynomial over $\mathbb{F}_{2^n}[X]$
- **Algebraic Degree** F : The maximum of the degrees of the coordinates of F
- **Quadratic Function**: F is quadratic if its algebraic degree is 2
- **Graph of a Function**: $\Gamma_F = \{(x, F(x)) \,|\, x \in \mathbb{F}_2^n\}$

A quadratic Cubic example

$F(x) = x^3 = x^2 \cdot x$ is APN and has univariate degree 3 and algebraic degree 2

Equivalence Relations - 1

Extended-Affine Equivalence

F and G are *EA-equivalent* if there exists A, B affine permutations over $\mathbb{F}_2^n$ and C an affine mapping over $\mathbb{F}_2^n$ such that $G = B \circ F \circ A + C$. In terms of graph:

Equivalence Relations - 1

Extended-Affine Equivalence

F and G are *EA-equivalent* if there exists A,B affine permutations over $\mathbb{F}_2^n$ and C an affine mapping over $\mathbb{F}_2^n$ such that $G = B \circ F \circ A + C$. In terms of graph:

$$\Gamma_G = \begin{bmatrix} A^{-1} & 0 \\ CA^{-1} & B \end{bmatrix} (\Gamma_F)$$

Equivalence Relations - 1

Extended-Affine Equivalence

F and G are *EA-equivalent* if there exists A, B affine permutations over $\mathbb{F}_2^n$ and C an affine mapping over $\mathbb{F}_2^n$ such that $G = B \circ F \circ A + C$. In terms of graph:

$$\Gamma_G = \begin{bmatrix} A^{-1} & 0 \\ CA^{-1} & B \end{bmatrix} (\Gamma_F)$$

Such affine permutation over $\mathbb{F}_2^{2n}$ is called an **EA-mapping**.

For all EA-mappings

F is APN $\implies$ $B \circ F \circ A + C$ is APN

Equivalence Relations - 1

Extended-Affine Equivalence

F and G are *EA-equivalent* if there exists A, B affine permutations over $\mathbb{F}_2^n$ and C an affine mapping over $\mathbb{F}_2^n$ such that $G = B \circ F \circ A + C$. In terms of graph:

$$\Gamma_G = \begin{bmatrix} A^{-1} & 0 \\ CA^{-1} & B \end{bmatrix} (\Gamma_F)$$

Such affine permutation over $\mathbb{F}_2^{2n}$ is called an **EA-mapping**.

For all EA-mappings

F is APN $\implies B \circ F \circ A + C$ is APN

The cube again

$x^3 + L(x)$, $x^6 = (x^2)^3$ are EA-equivalent to x^3

Equivalence Relations - 2

Carlet-Charpin-Zinoviev (CCZ) Equivalence

F and G are *CCZ-equivalent* if there exists an affine permutation $\mathcal{A}$ over $\mathbb{F}_2^{2n}$ such that $\Gamma_G = \mathcal{A}(\Gamma_F)$, i.e :

$$\exists A, B, C, D \text{ affine } , \Gamma_G = \begin{bmatrix} A & D \\ C & B \end{bmatrix} (\Gamma_F)$$

- A mapping $\mathcal{A}$ is said to be **admissible** for F if $\mathcal{A}(\Gamma_F)$ is the graph of a function
- A function is said to be **CCZ-quadratic** if it is CCZ-equivalent to a quadratic function

Equivalence Relations - 2

Carlet-Charpin-Zinoviev (CCZ) Equivalence

F and G are *CCZ-equivalent* if there exists an affine permutation $\mathcal{A}$ over $\mathbb{F}_2^{2n}$ such that $\Gamma_G = \mathcal{A}(\Gamma_F)$, i.e :

$$\exists A, B, C, D \text{ affine } , \Gamma_G = \begin{bmatrix} A & D \\ C & B \end{bmatrix} (\Gamma_F)$$

- A mapping $\mathcal{A}$ is said to be **admissible** for F if $\mathcal{A}(\Gamma_F)$ is the graph of a function
- A function is said to be **CCZ-quadratic** if it is CCZ-equivalent to a quadratic function

For all admissible mappings

F is APN $\implies$ $\mathcal{A}(\Gamma_F)$ is the graph of an APN function

Equivalence Relations - 2

Carlet-Charpin-Zinoviev (CCZ) Equivalence

F and G are *CCZ-equivalent* if there exists an affine permutation $\mathcal{A}$ over $\mathbb{F}_2^{2n}$ such that $\Gamma_G = \mathcal{A}(\Gamma_F)$, i.e :

$$\exists A, B, C, D \text{ affine } , \Gamma_G = \begin{bmatrix} A & D \\ C & B \end{bmatrix} (\Gamma_F)$$

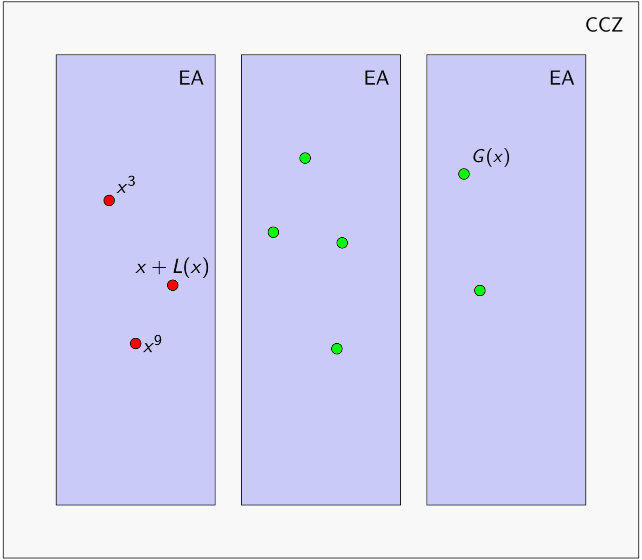
- A mapping $\mathcal{A}$ is said to be **admissible** for F if $\mathcal{A}(\Gamma_F)$ is the graph of a function
- A function is said to be **CCZ-quadratic** if it is CCZ-equivalent to a quadratic function

For all admissible mappings

F is APN $\implies$ $\mathcal{A}(\Gamma_F)$ is the graph of an APN function

- $G(x) = x^3 + (x^2 + x + 1)tr(x^3)$ is CCZ-equivalent to x^3 but not EA-equivalent

Equivalence - Summary



From now on, we assume that affine mappings are linear for the sake of simplicity

Bestiary of APN classes in numbers

A first lower bound

The number of CCZ classes of APN functions grows at least **exponentially** in the dimension (Kaspers and Zhou 2002).

Bestiary of APN classes in numbers

A first lower bound

The number of CCZ classes of APN functions grows at least **exponentially** in the dimension (Kaspers and Zhou 2002).

Numbers, before 2025

- In dimension 6 : 13 CCZ-quadratic classes, 1 non CCZ-quadratic

Bestiary of APN classes in numbers

A first lower bound

The number of CCZ classes of APN functions grows at least **exponentially** in the dimension (Kaspers and Zhou 2002).

Numbers, before 2025

- In dimension 6 : 13 CCZ-quadratic classes, 1 non CCZ-quadratic
- In dimension 7 : 448 CCZ-quadratic classes, 3 non CCZ-quadratic

Bestiary of APN classes in numbers

A first lower bound

The number of CCZ classes of APN functions grows at least **exponentially** in the dimension (Kaspers and Zhou 2002).

Numbers, before 2025

- In dimension 6 : 13 CCZ-quadratic classes, 1 non CCZ-quadratic
- In dimension 7 : 448 CCZ-quadratic classes, 3 non CCZ-quadratic
- In dimension 8 : 32,892 CCZ-quadratic classes, 1 non CCZ-quadratic....

Bestiary of APN classes in numbers

A first lower bound

The number of CCZ classes of APN functions grows at least **exponentially** in the dimension (Kaspers and Zhou 2002).

Numbers, before 2025

- In dimension 6 : 13 CCZ-quadratic classes, 1 non CCZ-quadratic
- In dimension 7 : 448 CCZ-quadratic classes, 3 non CCZ-quadratic
- In dimension 8 : 32,892 CCZ-quadratic classes, 1 non CCZ-quadratic....

In 2025

- Beierle et al. found 3.8 millions inequivalent quadratic functions...

Bestiary of APN classes in numbers

A first lower bound

The number of CCZ classes of APN functions grows at least **exponentially** in the dimension (Kaspers and Zhou 2002).

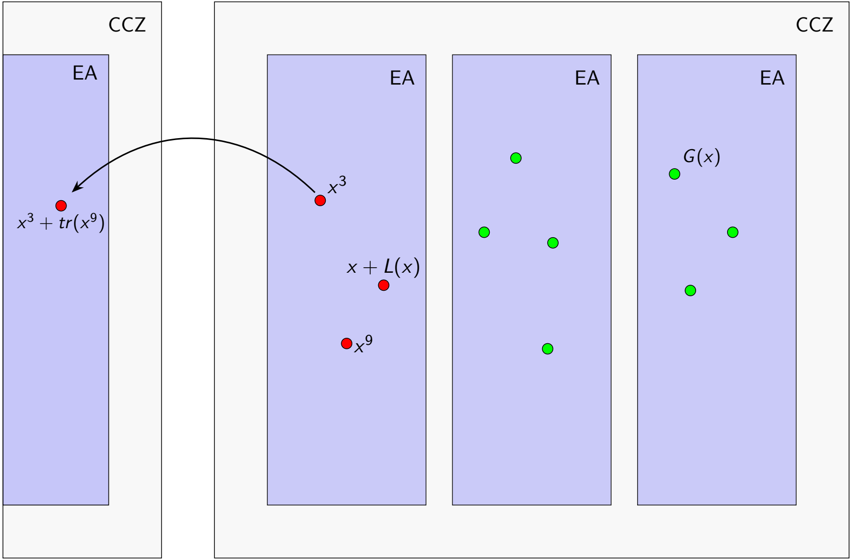
Numbers, before 2025

- In dimension 6 : 13 CCZ-quadratic classes, 1 non CCZ-quadratic
- In dimension 7 : 448 CCZ-quadratic classes, 3 non CCZ-quadratic
- In dimension 8 : 32,892 CCZ-quadratic classes, 1 non CCZ-quadratic....

In 2025

- Beierle et al. found 3.8 millions inequivalent quadratic functions...
- And they conjecture there are more !

What is a new function ?



Switching Neighbors

Switching Neighbours [EP09]

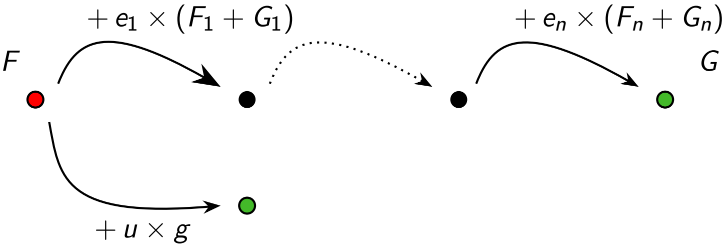
$u \in \mathbb{F}_2^n$, g a Boolean function, $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ APN, the function $F + u \times g$ is called an APN switching neighbour of F if it is an APN function.

Switching Neighbors

Switching Neighbours [EP09]

$u \in \mathbb{F}_2^n$, g a Boolean function, $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ APN, the function $F + u \times g$ is called an APN switching neighbour of F if it is an APN function.

Since $F + G = \sum_{i=1}^{n} e_i \times (F_i + G_i)$:

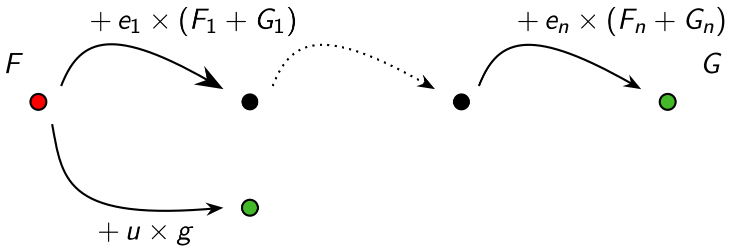


Switching Neighbors

Switching Neighbours [EP09]

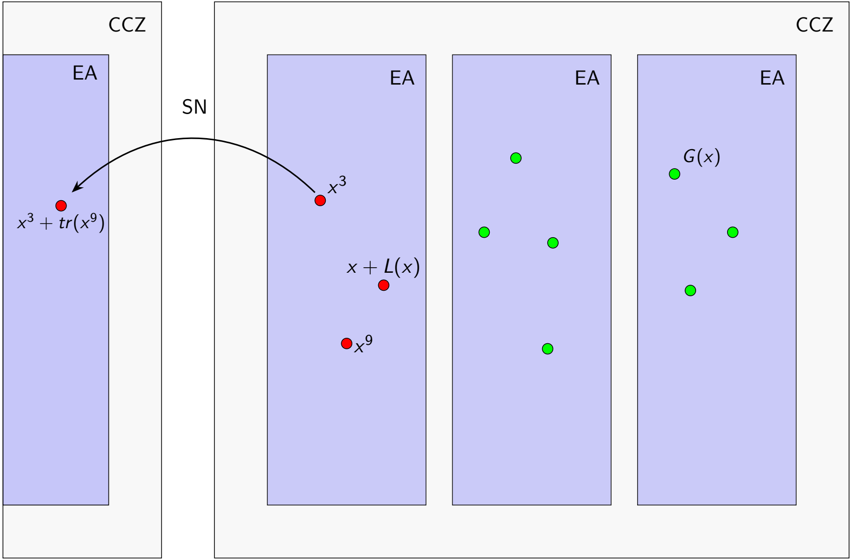
$u \in \mathbb{F}_2^n$, g a Boolean function, $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ APN, the function $F + u \times g$ is called an APN switching neighbour of F if it is an APN function.

Since $F + G = \sum_{i=1}^{n} e_i \times (F_i + G_i)$:



- Used to find the only known non-CCZ-quadratic APN in dimension 6 [EP09].
- Example: $x^3 + 1 \times tr(x^9)$, with $tr(x) = \sum_{i=0}^{n-1} x^{2^i}$

Exploration - Summary



Can we do it starting from the 3.8 millions function in dimension 8 ?

Could it be feasible ?

Problems

- Too many CCZ equivalent functions, nothing to filter them

Could it be feasible ?

Problems

- Too many CCZ equivalent functions, nothing to filter them
- Too many Switching Neighbours, no efficient algorithms to compute them

Could it be feasible ?

Problems

- Too many CCZ equivalent functions, nothing to filter them
- Too many Switching Neighbours, no efficient algorithms to compute them
- Bad perfomances with of the shelf implementations

Could it be feasible ?

Problems

- Too many CCZ equivalent functions, nothing to filter them
- Too many Switching Neighbours, no efficient algorithms to compute them
- Bad perfomances with of the shelf implementations

Our contributions

- We provide algorithms to efficiently explore the CCZ-class

Could it be feasible ?

Problems

- Too many CCZ equivalent functions, nothing to filter them
- Too many Switching Neighbours, no efficient algorithms to compute them
- Bad perfomances with of the shelf implementations

Our contributions

- We provide algorithms to efficiently explore the CCZ-class
- We provide theoretic improvements and algorithms to compute Switching Neighbours

Could it be feasible ?

Problems

- Too many CCZ equivalent functions, nothing to filter them
- Too many Switching Neighbours, no efficient algorithms to compute them
- Bad perfomances with of the shelf implementations

Our contributions

- We provide algorithms to efficiently explore the CCZ-class
- We provide theoretic improvements and algorithms to compute Switching Neighbours
- We provide public C++ implementations with SAGE bindings for all these algorithms

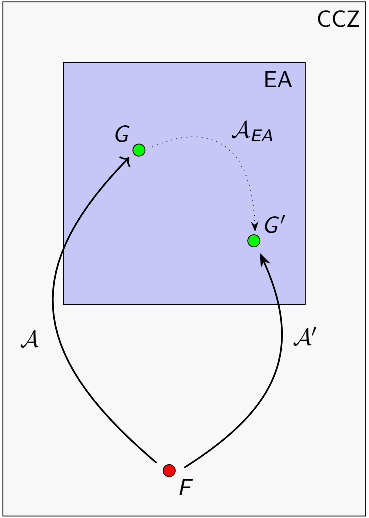
$\hookrightarrow$ From now on, everything is new !

Rest of the Talk

1. **Exploring the CCZ class**
2. **Exploring with Switching Neighbours**
3. **Experimental Results**

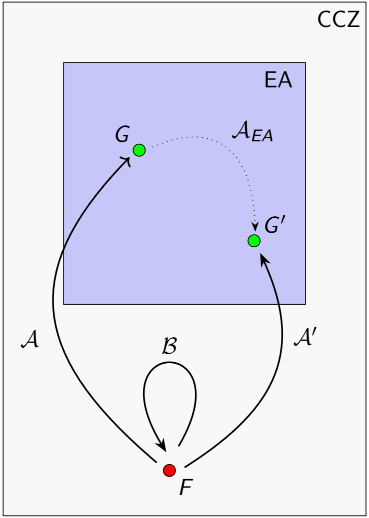
Exploring the CCZ class

Why is it harder than it seems ?



- We know how to build admissible mappings, $\mathcal{A}, \mathcal{A}'$
- Naively: $\mathcal{A}_{EA} = \mathcal{A} \circ \mathcal{A}'^{-1}$

Why is it harder than it seems ?

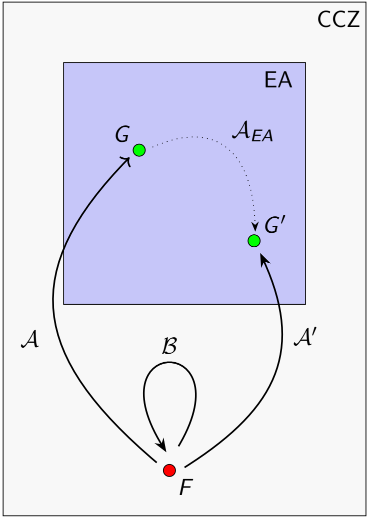


- We know how to build admissible mappings, $\mathcal{A}, \mathcal{A}'$
- Naively: $\mathcal{A}_{EA} = \mathcal{A} \circ \mathcal{A}'^{-1}$

$\mathrm{Aut}(F)$

$$(\mathcal{B} \in \mathrm{Aut}(F)) \iff (\mathcal{B}(\Gamma_F) = \Gamma_F)$$

Why is it harder than it seems ?



- We know how to build admissible mappings, $\mathcal{A}, \mathcal{A}'$
- Naively: $\mathcal{A}_{EA} = \mathcal{A} \circ \mathcal{A}'^{-1}$

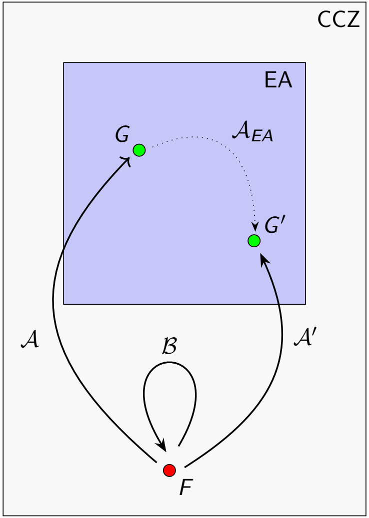
$\mathrm{Aut}(F)$

$$(\mathcal{B} \in \mathrm{Aut}(F)) \iff (\mathcal{B}(\Gamma_F) = \Gamma_F)$$

- Hence, the EA-mappings are:

$$\mathcal{A} \circ \mathcal{B} \circ \mathcal{A}'^{-1}$$

Why is it harder than it seems ?



- We know how to build admissible mappings, $\mathcal{A}, \mathcal{A}'$
- Naively: $\mathcal{A}_{EA} = \mathcal{A} \circ \mathcal{A}'^{-1}$

$\mathrm{Aut}(F)$

$(\mathcal{B} \in \mathrm{Aut}(F)) \iff (\mathcal{B}(\Gamma_F) = \Gamma_F)$

- Hence, the EA-mappings are:

$$\mathcal{A} \circ \mathcal{B} \circ \mathcal{A}'^{-1}$$

Conclusion

We have to compute $\mathrm{Aut}(F)$ to quotient, but can we do it efficiently ?

Admissible Mapping and LAT

Linear Approximation Table (LAT)

The LAT is the table of the values $W_F(a,b) = \sum_{x \in \mathbb{F}_2^n} (-1)^{a \cdot x + b \cdot F(x)}$, where $\cdot$ is the usual scalar product over $\mathbb{F}_2^n$.

Admissible Mapping and LAT

Linear Approximation Table (LAT)

The LAT is the table of the values $W_F(a, b) = \sum_{x \in \mathbb{F}_2^n} (-1)^{a \cdot x + b \cdot F(x)}$, where $\cdot$ is the usual scalar product over $\mathbb{F}_2^n$.

- $\mathcal{V} := \{(x, 0), x \in \mathbb{F}_2^n\}$
- The set of Walsh Zeroes $\mathcal{Z}_F := \{(\alpha, \beta) \mid W_F(\alpha, \beta) = 0\} \cup \{(0, 0)\}$
- $\mathcal{V} \subset \mathcal{Z}_F$

Admissible Mapping and LAT

Linear Approximation Table (LAT)

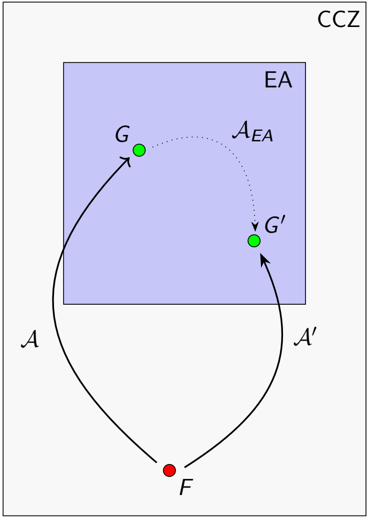
The LAT is the table of the values $W_F(a, b) = \sum_{x \in \mathbb{F}_2^n} (-1)^{a \cdot x + b \cdot F(x)}$, where $\cdot$ is the usual scalar product over $\mathbb{F}_2^n$.

- $\mathcal{V} := \{(x, 0), x \in \mathbb{F}_2^n\}$
- The set of Walsh Zeroes $\mathcal{Z}_F := \{(\alpha, \beta) \mid W_F(\alpha, \beta) = 0\} \cup \{(0, 0)\}$
- $\mathcal{V} \subset \mathcal{Z}_F$

Admissibility Criterion [CP19]

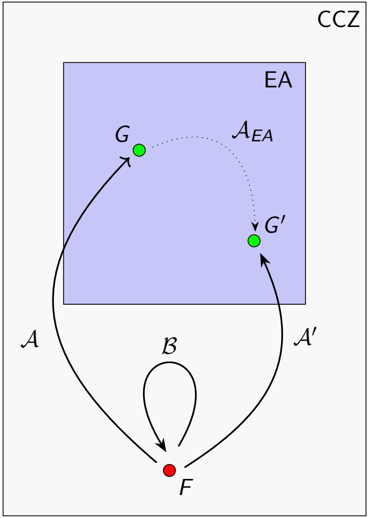
The mapping $\mathcal{A}$ is admissible for F if and only if $\mathcal{A}^\top(\mathcal{V}) \subset \mathcal{Z}_F$.

Filtering with Walsh Zeroes



- Admissible: $\mathcal{A}^\top(\mathcal{V}) \subset \mathcal{Z}_F$
- For G: $\mathcal{A}^\top(\mathcal{V}) = V$
- For G': ${\mathcal{A}'}^\top(\mathcal{V}) = V'$

Filtering with Walsh Zeroes

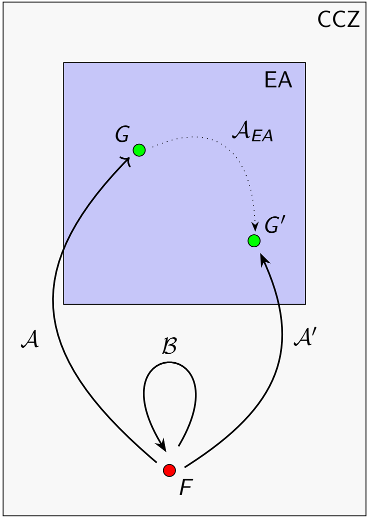


- Admissible: $\mathcal{A}^\top(\mathcal{V}) \subset \mathcal{Z}_F$
- For G: $\mathcal{A}^\top(\mathcal{V}) = V$
- For G': $\mathcal{A'}^\top(\mathcal{V}) = V'$

Proposition (refined from [CP19])

G and G' are EA-equivalent if and only if there exists $\mathcal{B} \in \mathrm{Aut}(F)$ such that $\mathcal{B}^\top(V) = V'$.

Filtering with Walsh Zeroes



- Admissible: $\mathcal{A}^\top(\mathcal{V}) \subset \mathcal{Z}_F$
- For G: $\mathcal{A}^\top(\mathcal{V}) = V$
- For G': $\mathcal{A'}^\top(\mathcal{V}) = V'$

Proposition (refined from [CP19])

G and G' are EA-equivalent if and only if there exists $\mathcal{B} \in \mathrm{Aut}(F)$ such that $\mathcal{B}^\top(V) = V'$.

Advantages

- Easier to test
- We already have an algorithm to search for the V [BPT19]

Filtering EA-equivalent functions

Filtering Algorithm

1. Compute all the vector spaces V_i of dimension n in $\mathcal{Z}_F$ (using [BPT19])

Filtering EA-equivalent functions

Filtering Algorithm

1. Compute all the vector spaces V_i of dimension n in $\mathcal{Z}_F$ (using [BPT19])
2. For all V_i and for all $\mathcal{B} \in \mathrm{Aut}(F)$, check if there exists k such that $\mathcal{B}^T(V_i) = V_k$. If it is the case, then remove V_k from the list of vector spaces.

Filtering EA-equivalent functions

Filtering Algorithm

1. Compute all the vector spaces V_i of dimension n in $\mathcal{Z}_F$ (using [BPT19])
2. For all V_i and for all $\mathcal{B} \in \mathrm{Aut}(F)$, check if there exists k such that $\mathcal{B}^T(V_i) = V_k$. If it is the case, then remove V_k from the list of vector spaces.
3. Loop over all remaining spaces V_i, build a corresponding admissible mapping, *i.e.* satisfying $\mathcal{A}^\top(\mathcal{V}) = V_i$.

Filtering EA-equivalent functions

Filtering Algorithm

1. Compute all the vector spaces V_i of dimension n in $\mathcal{Z}_F$ (using [BPT19])
2. For all V_i and for all $\mathcal{B} \in \mathrm{Aut}(F)$, check if there exists k such that $\mathcal{B}^T(V_i) = V_k$. If it is the case, then remove V_k from the list of vector spaces.
3. Loop over all remaining spaces V_i, build a corresponding admissible mapping, *i.e.* satisfying $\mathcal{A}^\top(\mathcal{V}) = V_i$.

Since we have almost only quadratic functions

How to compute the automorphism group Aut of quadratic functions ?

$\mathrm{Aut}(F) = \mathrm{Aut}_{\mathrm{EA}}(F)$

$\mathrm{Aut}_{\mathrm{EA}} \subset \mathrm{Aut}$: Automorphisms that are EA-mappings

$\mathrm{Aut}(F) = \mathrm{Aut}_{\mathrm{EA}}(F)$

$\mathrm{Aut}_{\mathrm{EA}} \subset \mathrm{Aut}$: Automorphisms that are EA-mappings

Theorem (CCZ equivalence of quadratic APN)

Let $n \geqslant 4$. Let $F, G : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be quadratic APN mappings. Then:

(i) Any automorphism of F is an EA-mapping: $\mathrm{Aut}(F) = \mathrm{Aut}_{\mathrm{EA}}(F)$ [KZ21]

(ii) if $\mathcal{A}$ is an admissible mapping satisfying $\mathcal{A}(\Gamma_F) = \Gamma_G$, then $\mathcal{A}$ is an EA-mapping.

$\mathrm{Aut}(F) = \mathrm{Aut}_{\mathrm{EA}}(F)$

$\mathrm{Aut}_{\mathrm{EA}} \subset \mathrm{Aut}$: Automorphisms that are EA-mappings

Theorem (CCZ equivalence of quadratic APN)

Let $n \geqslant 4$. Let $F, G : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be quadratic APN mappings. Then:

(i) Any automorphism of F is an EA-mapping: $\mathrm{Aut}(F) = \mathrm{Aut}_{\mathrm{EA}}(F)$ [KZ21]

(ii) if $\mathcal{A}$ is an admissible mapping satisfying $\mathcal{A}(\Gamma_F) = \Gamma_G$, then $\mathcal{A}$ is an EA-mapping.

There is only one EA-class of quadratic functions in a CCZ-class, like it is for x^3

$\mathrm{Aut}(F)$ and $\mathrm{Aut}(\pi_F)$

Definition (Ortho-derivative)

Let $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be a quadratic APN function. The *ortho-derivative* of F is the function $\pi_F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ where $\pi_F(0) = 0$ and where for any $a \in (\mathbb{F}_2^n)^*$, $\pi_F(a)$ is the only non-zero element satisfying:

$$\pi_F(a) \cdot \big(F(x+a) + F(x) + F(a) + F(0)\big) \; = \; 0 \quad \forall \, x \in \mathbb{F}_2^n \,. \tag{1}$$

$\mathrm{Aut}(F)$ and $\mathrm{Aut}(\pi_F)$

Definition (Ortho-derivative)

Let $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be a quadratic APN function. The *ortho-derivative* of F is the function $\pi_F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ where $\pi_F(0) = 0$ and where for any $a \in (\mathbb{F}_2^n)^*$, $\pi_F(a)$ is the only non-zero element satisfying:

$$\pi_F(a) \cdot \big(F(x + a) + F(x) + F(a) + F(0)\big) \; = \; 0 \quad \forall \; x \in \mathbb{F}_2^n \,. \tag{1}$$

$\mathrm{Aut}(F)$ and $\mathrm{Aut}_{\mathrm{LE}}(\pi_F)$ [CCP22]

Let $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be a quadratic APN function, if $F = B \circ F \circ A + C$, then π_F satisfies $\pi_F = B^T \circ \pi_F \circ A^{-1}$.

LAT of π_f and Computing $\mathrm{Aut}(\pi_F)$

The LAT of π_F looks like the LAT of random functions: they have large and 'rarer' coefficients.

LAT of π_f and Computing $\mathrm{Aut}(\pi_F)$

The LAT of π_F looks like the LAT of random functions: they have large and 'rarer' coefficients.

SboxU demonstration

The LAT of the Kim mapping and its ortho-derivative

Computing $\mathrm{Aut}(\pi_F)$ (as in [HK22])

- Coefficients of the same value are mapped to each other

LAT of π_f and Computing $\mathrm{Aut}(\pi_F)$

The LAT of π_F looks like the LAT of random functions: they have large and 'rarer' coefficients.

SboxU demonstration

The LAT of the Kim mapping and its ortho-derivative

Computing $\mathrm{Aut}(\pi_F)$ (as in [HK22])

- Coefficients of the same value are mapped to each other
- It put constraints on A and B quite fast since the batches of coefficients are small

LAT of π_f and Computing $\mathrm{Aut}(\pi_F)$

The LAT of π_F looks like the LAT of random functions: they have large and 'rarer' coefficients.

SboxU demonstration

The LAT of the Kim mapping and its ortho-derivative

Computing $\mathrm{Aut}(\pi_F)$ (as in [HK22])

- Coefficients of the same value are mapped to each other
- It put constraints on A and B quite fast since the batches of coefficients are small
- Very fast algorithm for ortho-derivatives in our experiments

Computing the Group of Automorphisms

Computing $\mathrm{Aut}(E)$

1. Compute the ortho-derivative π_F of the target function F.

Computing the Group of Automorphisms

Computing $\mathrm{Aut}(E)$

1. Compute the ortho-derivative π_F of the target function F.
2. Find all the pairs of linear permutations B^T and A^{-1} such that $B^T \circ \pi_F \circ A^{-1} = \pi_F$.

Computing the Group of Automorphisms

Computing $\mathrm{Aut}(E)$

1. Compute the ortho-derivative π_F of the target function F.
2. Find all the pairs of linear permutations B^T and A^{-1} such that $B^T \circ \pi_F \circ A^{-1} = \pi_F$.
3. For all $\delta \in \mathbb{F}_2^n$, compute the values of the function C such that $C(x) = F(x + \delta) + B \circ F \circ A(x)$. If C is affine, then we have just deduced an automorphism $\mathcal{A}$ of F.

Computing the Group of Automorphisms

Computing $\mathrm{Aut}(E)$

1. Compute the ortho-derivative π_F of the target function F.
2. Find all the pairs of linear permutations B^T and A^{-1} such that $B^T \circ \pi_F \circ A^{-1} = \pi_F$.
3. For all $\delta \in \mathbb{F}_2^n$, compute the values of the function C such that $C(x) = F(x + \delta) + B \circ F \circ A(x)$. If C is affine, then we have just deduced an automorphism $\mathcal{A}$ of F.

Available in sboxU

`graph_automorphisms_of_apn_quadratic`

Exploring with Switching Neighbours

1. Exploring the CCZ class

2. Exploring with Switching Neighbours

3. Experimental Results

The Switching Neighbors Criterion

Switching Neighbour Criterion [EP09]

If F is an APN function, $F + u \times g$ is an APN function if and only if :

$$g(x) + g(x + a) + g(y) + g(y + a) = 0$$

for all $x, y, a \in \mathbb{F}_2^n$ with :

$$F(x) + F(x + a) + F(y) + F(y + a) = u.$$

The Switching Neighbors Criterion

Switching Neighbour Criterion [EP09]

If F is an APN function, $F + u \times g$ is an APN function if and only if :

$$g(x) + g(x + a) + g(y) + g(y + a) = 0$$

for all $x, y, a \in \mathbb{F}_2^n$ with :

$$F(x) + F(x + a) + F(y) + F(y + a) = u.$$

Why is it a linear problem ?

The criterion boils down to $v_{x,y,a} \cdot lut(g) = 0$ with

- $lut(g) = (g(0), ..., g(2^n - 1)) \in (\mathbb{F}_2)^{2^n}$
- $v_{x,y,a} = (0, ..., 0, 1, 0, ..., 0, 1, 0, ..., 0, 1, 0, ..., 0, 1, 0, ..., 0) \in (\mathbb{F}_2)^{2^n}$

The Switching Neighbors Criterion

Switching Neighbour Criterion [EP09]

If F is an APN function, $F + u \times g$ is an APN function if and only if :

$$g(x) + g(x + a) + g(y) + g(y + a) = 0$$

for all $x, y, a \in \mathbb{F}_2^n$ with :

$$F(x) + F(x + a) + F(y) + F(y + a) = u.$$

Why is it a linear problem ?

The criterion boils down to $v_{x,y,a} \cdot lut(g) = 0$ with

- $lut(g) = (g(0), ..., g(2^n - 1)) \in (\mathbb{F}_2)^{2^n}$
- $v_{x,y,a} = (0, ..., 0, 1, 0, ..., 0, 1, 0, ..., 0, 1, 0, ..., 0, 1, 0, ..., 0) \in (\mathbb{F}_2)^{2^n}$

If M_u is the matrix composed of $v_{x,y,a}$, we need to compute $Ker(M_u)$

Some Size Problems

- Naively, we need $2^{2n} \times 2^n$ equations to have our systems.

Some Size Problems

- Naively, we need $2^{2n} \times 2^n$ equations to have our systems.
- Experimentally, we have at least 2^{2n} switching neighbours per u, are they all interesting ?

Some Size Problems

- Naively, we need $2^{2n} \times 2^n$ equations to have our systems.
- Experimentally, we have at least 2^{2n} switching neighbours per u, are they all interesting ?

For $n = 8$ with 3.8 millions functions

- Linear algebra over 2^8 matrices of size $2^8 \times 2^{16}$ per function: $\approx 2^{32}$ per function

Some Size Problems

- Naively, we need $2^{2n} \times 2^n$ equations to have our systems.
- Experimentally, we have at least 2^{2n} switching neighbours per u, are they all interesting ?

For $n = 8$ with 3.8 millions functions

- Linear algebra over 2^8 matrices of size $2^8 \times 2^{16}$ per function: $\approx 2^{32}$ per function
- $2^{16} \times 2^8$ potential new functions to test per function: $\approx 2^{45}$ without counting the CCZ class

Some Size Problems

- Naively, we need $2^{2n} \times 2^n$ equations to have our systems.
- Experimentally, we have at least 2^{2n} switching neighbours per u, are they all interesting ?

For $n = 8$ with 3.8 millions functions

- Linear algebra over 2^8 matrices of size $2^8 \times 2^{16}$ per function: $\approx 2^{32}$ per function
- $2^{16} \times 2^8$ potential new functions to test per function: $\approx 2^{45}$ without counting the CCZ class

$\hookrightarrow$ It is not going to be feasible with just better implementations

Switching Neighbours and EA equivalence

The set switches $\mathrm{SW}_F(u)$

The set of boolean functions g for which $F + u \times g$ is APN is noted $\mathrm{SW}_F(u)$. We have that:

- The set $\mathrm{SW}_F(u)$ is a vector space

Switching Neighbours and EA equivalence

The set switches $\mathrm{SW}_F(u)$

The set of boolean functions g for which $F + u \times g$ is APN is noted $\mathrm{SW}_F(u)$. We have that:

- The set $\mathrm{SW}_F(u)$ is a vector space
- If $G = AFB + C$, then:

$$(f \in \mathrm{SW}_F(u)) \iff (f \in \mathrm{SW}_G(A(u)) \circ B^{-1})$$

Switching Neighbours and EA equivalence

The set switches $\mathrm{SW}_F(u)$

The set of boolean functions g for which $F + u \times g$ is APN is noted $\mathrm{SW}_F(u)$. We have that:

- The set $\mathrm{SW}_F(u)$ is a vector space
- If $G = AFB + C$, then:

$$(f \in \mathrm{SW}_F(u)) \iff (f \in \mathrm{SW}_G(A(u)) \circ B^{-1})$$

EA-invariance

- We only need to compute the switching neighbours of one function in the EA class !
- Can be turned into a powerful invariant

Filtering Switching Neighbours

Affine Functions

- $\mathrm{Aff}(n) \subset \mathsf{SW}_F(u)$ is of dimension $n + 1$
- $\forall g \in \mathrm{Aff}(n),\ F + u \times g$ is EA-equivalent to F

Filtering Switching Neighbours

Affine Functions

- $\mathrm{Aff}(n) \subset \mathrm{SW}_F(u)$ is of dimension $n+1$
- $\forall g \in \mathrm{Aff}(n)$, $F + u \times g$ is EA-equivalent to F

Coordinates

- $\langle F_1, \ldots, F_n \rangle \cap \mathrm{SW}_F(u) \subset \mathrm{SW}_F(u)$ is of dimension $n-1$
- It also gives EA-equivalent switching neighbours

Filtering Switching Neighbours

Affine Functions

- $\mathrm{Aff}(n) \subset \mathrm{SW}_F(u)$ is of dimension $n+1$
- $\forall g \in \mathrm{Aff}(n)$, $F + u \times g$ is EA-equivalent to F

Coordinates

- $\langle F_1, \ldots, F_n \rangle \cap \mathrm{SW}_F(u) \subset \mathrm{SW}_F(u)$ is of dimension $n-1$
- It also gives EA-equivalent switching neighbours

Non-Trivial Switching Neighbours

We can decompose $\mathrm{SW}_F(u)$ in the following direct sum:

$$\mathrm{SW}_F(u) = \mathrm{SW}_F^{nt}(u) \oplus \mathrm{Aff}(n) \oplus \langle F_1, \ldots, F_n \rangle \cap \mathrm{SW}_F(u)$$

For such vector space $\mathrm{SW}_F^{nt}(u)$, we call $F + u \times \mathrm{SW}_F^{nt}(u)$ a set of non-trivial switching neighbours of F.

Computing Switching Neighbours in practice

Rephrasing of Edel and Pott's Criterion

- Let M_u be composed of $v_{x,y,a} = (0, ..., 0, 1, 0, ..., 0, 1, 0, ..., 0, 1, 0, ..., 0, 1, 0, ..., 0)$ with: $F(x) + F(x + a) + F(y) + F(y + a) = u$
- $\mathrm{SW}_F(u) = Ker(M_u)$
- Add constraints to M_u to actually compute $\mathrm{SW}_F^{nt}(u)$

Computing Switching Neighbours in practice

Rephrasing of Edel and Pott's Criterion

- Let M_u be composed of $v_{x,y,a} = (0, ..., 0, 1, 0, ..., 0, 1, 0, ..., 0, 1, 0, ..., 0, 1, 0, ..., 0)$ with: $F(x) + F(x+a) + F(y) + F(y+a) = u$
- $\mathrm{SW}_F(u) = Ker(M_u)$
- Add constraints to M_u to actually compute $\mathrm{SW}_F^{nt}(u)$

Our Algorithm

- Draw lines in $v_{x,y,a}$ at random to get a matrix M such that $Ker(M) \supset Ker(M_u)$

Computing Switching Neighbours in practice

Rephrasing of Edel and Pott's Criterion

- Let M_u be composed of $v_{x,y,a} = (0, ..., 0, 1, 0, ..., 0, 1, 0, ..., 0, 1, 0, ..., 0, 1, 0, ..., 0)$ with: $F(x) + F(x+a) + F(y) + F(y+a) = u$
- $\mathrm{SW}_F(u) = Ker(M_u)$
- Add constraints to M_u to actually compute $\mathrm{SW}_F^{nt}(u)$

Our Algorithm

- Draw lines in $v_{x,y,a}$ at random to get a matrix M such that $Ker(M) \supset Ker(M_u)$
- Check if the all the functions obtained are APN

Computing Switching Neighbours in practice

Rephrasing of Edel and Pott's Criterion

- Let M_u be composed of $v_{x,y,a} = (0, ..., 0, 1, 0, ..., 0, 1, 0, ..., 0, 1, 0, ..., 0, 1, 0, ..., 0)$ with: $F(x) + F(x+a) + F(y) + F(y+a) = u$
- $\mathrm{SW}_F(u) = Ker(M_u)$
- Add constraints to M_u to actually compute $\mathrm{SW}_F^{nt}(u)$

Our Algorithm

- Draw lines in $v_{x,y,a}$ at random to get a matrix M such that $Ker(M) \supset Ker(M_u)$
- Check if the all the functions obtained are APN
- If not, add equations by batch until all functions obtained are APN, i.e $Ker(M) = Ker(M_u)$

Computing Switching Neighbours in practice

Rephrasing of Edel and Pott's Criterion

- Let M_u be composed of $v_{x,y,a} = (0, ..., 0, 1, 0, ..., 0, 1, 0, ..., 0, 1, 0, ..., 0, 1, 0, ..., 0)$ with: $F(x) + F(x+a) + F(y) + F(y+a) = u$
- $\mathrm{SW}_F(u) = Ker(M_u)$
- Add constraints to M_u to actually compute $\mathrm{SW}_F^{nt}(u)$

Our Algorithm

- Draw lines in $v_{x,y,a}$ at random to get a matrix M such that $Ker(M) \supset Ker(M_u)$
- Check if the all the functions obtained are APN
- If not, add equations by batch until all functions obtained are APN, i.e $Ker(M) = Ker(M_u)$

As it turns out, M is of maximum rank much faster than expected. For $n = 8$ we get $Ker(M) = Ker(M_u)$ in roughly 2.5×2^n random equations.

Switching Neighbours - Summary

Our Improvements

- We have reduced by a factor 2^{3n} the number of switching neighbours per function that we need to check

Switching Neighbours - Summary

Our Improvements

- We have reduced by a factor 2^{3n} the number of switching neighbours per function that we need to check
- We have improved by almost a factor 2^n the number of equations needed to compute switching neighbours.

Switching Neighbours - Summary

Our Improvements

- We have reduced by a factor 2^{3n} the number of switching neighbours per function that we need to check
- We have improved by almost a factor 2^n the number of equations needed to compute switching neighbours.

Available in `sboxU`

- `non_trivial_sn`
- We also have very efficient implementations of linear algera !

Experimental Results

1. Exploring the CCZ class

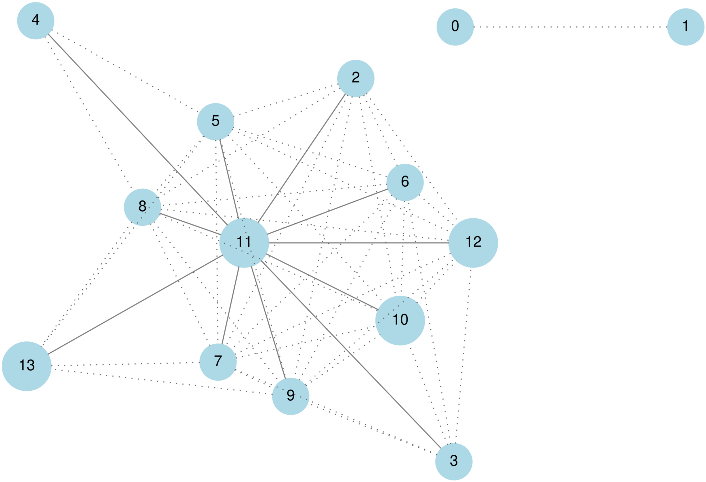
2. Exploring with Switching Neighbours

3. **Experimental Results**

For $\mathbb{F}_2^6$ - CCZ-SN Graph

- Node: CCZ class
- Edge: SN between classes
- Labels: Number in the DB

- 0: The cube
- 4: The Kim mapping
- 13: Only non CCZ-quadratic



For $\mathbb{F}_2^6$ - Graph Summary

CCZ-SN components

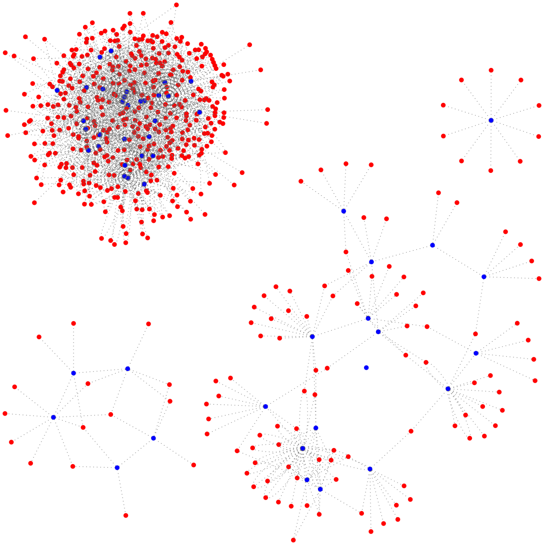
- The component of the cube - labelled 0
- The component of the Kim mapping - labelled 4
- The first graph is the identical when we consider only quadratic functions

Class 11

$F(x) = x^9 + u^4\,(x^{10} + x^{18}) + u^9\,(x^{12} + x^{20} + x^{40})$ is connected to all other CCZ-classes in its component and it is the only one !

For $\mathbb{F}_2^6$ - CCZ-Class 11

- Blue node: EA class in 11
- Red node: EA class **not** in 11
- Edge: SN EA-classes



For $\mathbb{F}_2^6$

Experimental Results

- We found all the 691 EA-classes in the union of the 13 known quadratic CCZ-equivalence classes (same as [Cal20])

For $\mathbb{F}_2^6$

Experimental Results

- We found all the 691 EA-classes in the union of the 13 known quadratic CCZ-equivalence classes (same as [Cal20])
- We have 25 EA-classes not CCZ-equivalent to quadratic (same as [Cal20])

For $\mathbb{F}_2^6$

Experimental Results

- We found all the 691 EA-classes in the union of the 13 known quadratic CCZ-equivalence classes (same as [Cal20])
- We have 25 EA-classes not CCZ-equivalent to quadratic (same as [Cal20])
- It is very important: reproductibility in experiments is key
- Computing the CCZ-SN graph can be done in 1 minute on a regular laptop

For $\mathbb{F}_2^6$

Experimental Results

- We found all the 691 EA-classes in the union of the 13 known quadratic CCZ-equivalence classes (same as [Cal20])
- We have 25 EA-classes not CCZ-equivalent to quadratic (same as [Cal20])
- It is very important: reproductibility in experiments is key
- Computing the CCZ-SN graph can be done in 1 minute on a regular laptop

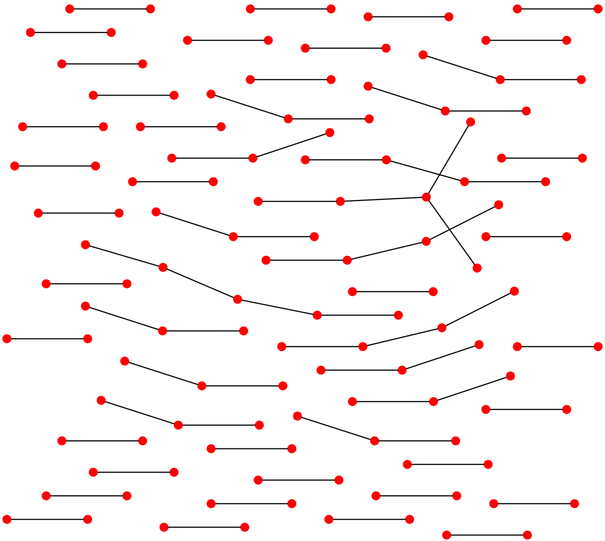
Conclusion

- All EA-classes are stored in a tinySQL database with invariants to help to test new functions
- If new APN functions exist, they are in another component
- Lightning fast performances to do experiments !

For $\mathbb{F}_2^7$ - Quadratic only

- Node: Quadratic function
- Edge: SN between functions

The size of the components does not seem to scale



For $\mathbb{F}_2^7$

Experimental Results

- 488 EA-classes of quadratic APN functions.

For $\mathbb{F}_2^7$

Experimental Results

- 488 EA-classes of quadratic APN functions.
- New : The union of their CCZ-classes contains a total of 82,409 EA-classes

For $\mathbb{F}_2^7$

Experimental Results

- 488 EA-classes of quadratic APN functions.
- New : The union of their CCZ-classes contains a total of 82,409 EA-classes
- Only 9890 functions have switching neighbours - 12 % of the entire set
- No new functions were found

Conclusion

- All EA-classes are stored in a tinySQL database with invariants to help to test new functions
- Possibly many other small CCZ-SN components
- Lightning fast performances to do experiments !

For $\mathbb{F}_2^8$ - Problems

Size Concerns

- We have 3.8 millions new starting-points, but...

For $\mathbb{F}_2^8$ - Problems

Size Concerns

- We have 3.8 millions new starting-points, but...
- The CCZ-class of these functions contain roughly 300 EA-classes

For $\mathbb{F}_2^8$ - Problems

Size Concerns

- We have 3.8 millions new starting-points, but...
- The CCZ-class of these functions contain roughly 300 EA-classes
- It takes roughly 0.6s to compute the switching neighbours of a 8-bit function

- $300 \times 0.6 \times 3.8 \times 10^6 \approx 7917$days of computations

For $\mathbb{F}_2^8$ - Problems

Size Concerns

- We have 3.8 millions new starting-points, but...
- The CCZ-class of these functions contain roughly 300 EA-classes
- It takes roughly 0.6s to compute the switching neighbours of a 8-bit function

- $300 \times 0.6 \times 3.8 \times 10^6 \approx 7917$days of computations
- 256 Byte per look-up makes about 1TB of database

For $\mathbb{F}_2^8$ - Experiments on Quadratic Functions

For quadratic functions only

- $0.6 \times 3.8 \times 10^6 \approx 26$ days of computations

For $\mathbb{F}_2^8$ - Experiments on Quadratic Functions

For quadratic functions only

- $0.6 \times 3.8 \times 10^6 \approx 26$ days of computations
- Quadratic have small ANF : 28 Bytes per function, about 250MB for the entire database

For $\mathbb{F}_2^8$ - Experiments on Quadratic Functions

For quadratic functions only

- $0.6 \times 3.8 \times 10^6 \approx 26$ days of computations
- Quadratic have small ANF : 28 Bytes per function, about 250MB for the entire database
- Parallelized on one node of the grid5000 cluster for about a day

For $\mathbb{F}_2^8$ - Experiments on Quadratic Functions

For quadratic functions only

- $0.6 \times 3.8 \times 10^6 \approx 26$ days of computations
- Quadratic have small ANF : 28 Bytes per function, about 250MB for the entire database
- Parallelized on one node of the grid5000 cluster for about a day

We found new functions !

- Among the 3.8 millions, only 1% of them have non trivial switching neighbours
- Among these switching neighbours, about 10% of them were new

For $\mathbb{F}_2^8$ - Experiments on Quadratic Functions

For quadratic functions only

- $0.6 \times 3.8 \times 10^6 \approx 26$ days of computations
- Quadratic have small ANF : 28 Bytes per function, about 250MB for the entire database
- Parallelized on one node of the grid5000 cluster for about a day

We found new functions !

- Among the 3.8 millions, only 1% of them have non trivial switching neighbours
- Among these switching neighbours, about 10% of them were new

However

These functions are all CCZ-quadratic and none is CCZ-equivalent to a bijection

For $\mathbb{F}_2^8$ - As is the tradition

[0,0,180,132,180,244,96,16,180,188,76,116,118,62,238,150,180,141,196,205,
103,30,119,62,144,161,172,173,53,68,105,40,107,243,216,112,156,68,79,167,
70,214,185,25,199,23,88,184,225,64,150,7,113,144,102,183,92,245,103,254,
186,83,225,56,180,185,152,165,58,119,118,11,148,145,244,193,108,41,108,
25,22,34,254,250,255,139,119,51,166,154,2,14,57,69,253,177,169,60,130,
39,100,177,47,202,16,141,119,218,171,118,172,65,53,153,218,70,159,115,
16,204,28,184,191,43,192,36,3,215,223,156,22,101,115,112,218,233,131,
200,6,125,89,82,188,135,75,49,70,12,128,186,237,231,135,245,198,132,
58,8,27,25,50,233,252,23,221,70,115,216,247,36,117,150,110,253,140,47,
152,122,146,64,16,178,122,232,205,39,139,81,51,153,21,143,66,12,19,109,
212,218,229,219,138,204,151,225,106,108,23,33,192,183,85,18,49,6,196,195,
152,231,65,14,31,32,166,169,217,15,143,105,12,154,58,156,136,86,146,124,
43,181,81,255,101,138,247,40,215,120,37,186,164,67,122,173,96,199,222,73]

More Exploration

New Open Problems

- Are there other components for $n = 6$ and $n = 7$?

More Exploration

New Open Problems

- Are there other components for $n = 6$ and $n = 7$?
- What are the biggest CCZ-SN component for $n = 7$, $n = 8$?

More Exploration

New Open Problems

- Are there other components for $n = 6$ and $n = 7$?
- What are the biggest CCZ-SN component for $n = 7$, $n = 8$?
- Do we need to look at the entire CCZ-class of a function for our exploration ?

More Exploration

New Open Problems

- Are there other components for $n = 6$ and $n = 7$?
- What are the biggest CCZ-SN component for $n = 7$, $n = 8$?
- Do we need to look at the entire CCZ-class of a function for our exploration ?

More Exploration

New Open Problems

- Are there other components for $n = 6$ and $n = 7$?
- What are the biggest CCZ-SN component for $n = 7$, $n = 8$?
- Do we need to look at the entire CCZ-class of a function for our exploration ?

Experiment !

More miscellaneous facts to be intuited... leading to proofs ?

More Exploration

New Open Problems

- Are there other components for $n = 6$ and $n = 7$?
- What are the biggest CCZ-SN component for $n = 7$, $n = 8$?
- Do we need to look at the entire CCZ-class of a function for our exploration ?

Experiment !

More miscellaneous facts to be intuited... leading to proofs ?

Example: CCZ-class of x^3

- For $n = 6$, there are **exactly** 3 EA-classes
- For $n = 7$, there are **exactly** 3 EA-classes
- For $n = 8$, there are **exactly** 2 EA-classes

`sboxU` v2 advert

- C++ optimized code
- Databases in tinySQL available for $n = 6$ and $n = 7$
- All the code needed to replicate our experiments
- Many more for S-box analysis

https://github.com/lpp-crypto/sboxU

- A lot of people involved from COSMIQ: Léo, Merlin, Guilhem, me
- And from faraway: Jules, Xavier, Baptiste G.
- Improved structure: it will be easier to contribute, help !
- What do YOU want to see implemented ?

Xavier Bonnetain, Léo Perrin, and Shizhu Tian.
Anomalies and vector space search: Tools for S-box analysis.
In *Advances in Cryptology – ASIACRYPT 2019*, Cham, 2019. Springer International Publishing.

Marco Calderini.
On the EA-classes of known APN functions in small dimensions.
Cryptogr. Commun., 12(5):821–840, 2020.

Anne Canteaut, Alain Couvreur, and Léo Perrin.
Recovering or testing extended-affine equivalence.
IEEE Transactions on Information Theory, 68(9):6187–6206, 2022.

Anne Canteaut and Léo Perrin.
On CCZ-equivalence, extended-affine equivalence, and function twisting.
Finite Fields and Their Applications, 56:209–246, 2019.

Yves Edel and Alexander Pott.
A new almost perfect nonlinear function which is not quadratic.

Adv. Math. Commun., 3(1):59–81, 2009.

Marie Heggebakk and Nikolay Kaleyski.
Testing linear and affine equivalence.
In *BFA 2022: The 7th International Workshop on Boolean Functions and their Applications*, Balestrand, Norway, September 2022.
https://boolean.w.uib.no/bfa-2022/.

Christian Kaspers and Yue Zhou.
The number of almost perfect nonlinear functions grows exponentially.
J. Cryptol., 34(1):4, 2021.