

How to Take a Function Apart with SboxU Léo Perrin

▶ To cite this version:

Léo Perrin. How to Take a Function Apart with SboxU. BFA 2020 - The 5th International Workshop on Boolean Functions and their Applications, Sep 2020, Loen, Norway. hal-03136551

HAL Id: hal-03136551 https://hal.inria.fr/hal-03136551

Submitted on 9 Feb 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers. L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

How to Take a Function Apart with SboxU (Also Featuring some New Results on Ortho-Derivatives)

Anne Canteaut¹, <u>Léo Perrin¹</u>

¹Inria, France

leo.perrin@inria.fr

🕒 @lpp_crypto



Boolean Functions and their Applications 2020





A wild vectorial Boolean function appears!



A wild vectorial Boolean function appears!

What do you do?

Outline

1 Basic Functionalities

- 2 CCZ-Equivalence
- 3 Ortho-Derivative

4 Conclusion

Basic Functionalities

Core Functionalities

Plan of this Section

1 Basic Functionalities

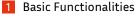
- Installation
- Core Functionalities



Basic Functionalities

Installation Core Functionalities

Plan of this Section



- Installation
- Core Functionalities
- 2 CCZ-Equivalence
- 3 Ortho-Derivative



Installation Core Functionalities

How to

- You need to have SAGE installed
- Then head to https://github.com/lpp-crypto/sboxU



Installation Core Functionalities

Sbox from SAGE vs. sboxU

There are already many functions for investigating vectorial boolean functions in SAGE:

- Class SBox from sage.crypto.sbox (or sage.crypto.mq.sbox in older versions)
- Module boolean_function from sage.crypto

Installation Core Functionalities

Sbox from SAGE vs. sboxU

There are already many functions for investigating vectorial boolean functions in SAGE:

- Class SBox from sage.crypto.sbox (or sage.crypto.mq.sbox in older versions)
- Module boolean_function from sage.crypto

SAGE SBox

- Supports output size ≠ input size
- Sub-routines written in Python or Cython
- Built-in SAGE

sboxU

- Assumes output size = input size
- Sub-routines written in Python or multi-threaded C++
- Cutting edge functionalities

Basic Functionalities

Core Functionalities

Plan of this Section



1 Basic Functionalities

- Installation
- Core Functionalities



Installation Core Functionalities

Some Tools

DDT/LAT (+ Pollock representation thereof)



Installation Core Functionalities

Some Tools

1 DDT/LAT (+ Pollock representation thereof)

Demo

2 ANF, algebraic degree



Installation Core Functionalities

Some Tools

DDT/LAT (+ Pollock representation thereof)

ANF, algebraic degree



Demo

3 Finite field arithmetic

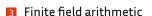


Installation Core Functionalities

Some Tools

DDT/LAT (+ Pollock representation thereof)

2 ANF, algebraic degree





Demo

Demo

4 Linear mappings



Definition and Basic Theorems How Can sboxU Help?

Plan of this Section



- 2 CCZ-Equivalence
 - Definition and Basic Theorems
 - How Can sboxU Help?
- 3 Ortho-Derivative



Definition and Basic Theorems How Can sboxU Help?

Plan of this Section



2 CCZ-Equivalence
 Definition and Basic Theorems
 How Can sboxU Help?

3 Ortho-Derivative



Definition and Basic Theorems How Can sboxU Help?

CCZ- and EA-equivalence

Definition (CCZ-Equivalence)

 $F: \mathbb{F}_2^n \to \mathbb{F}_2^m$ and $G: \mathbb{F}_2^n \to \mathbb{F}_2^m$ are C(arlet)-C(harpin)-Z(inoviev) equivalent if

 $\Gamma_{G} = \left\{ (x, G(x)), \forall x \in \mathbb{F}_{2}^{n} \right\} = L\left(\left\{ (x, F(x)), \forall x \in \mathbb{F}_{2}^{n} \right\} \right) = L(\Gamma_{F}),$

where $L: \mathbb{F}_2^{n+m} \to \mathbb{F}_2^{n+m}$ is an affine permutation.

Definition and Basic Theorems How Can sboxU Help?

CCZ- and EA-equivalence

Definition (CCZ-Equivalence)

 $F: \mathbb{F}_2^n \to \mathbb{F}_2^m$ and $G: \mathbb{F}_2^n \to \mathbb{F}_2^m$ are C(arlet)-C(harpin)-Z(inoviev) equivalent if

$$\Gamma_{G} = \left\{ (x, G(x)), \forall x \in \mathbb{F}_{2}^{n} \right\} = L\left(\left\{ (x, F(x)), \forall x \in \mathbb{F}_{2}^{n} \right\} \right) = L(\Gamma_{F})$$

where $L: \mathbb{F}_2^{n+m} \to \mathbb{F}_2^{n+m}$ is an affine permutation.

Definition (EA-Equivalence; EA-mapping)

F and G are E(xtented) A(ffine) equivalent if $G(x) = (B \circ F \circ A)(x) + C(x)$, where A, B, C are affine and A, B are permutations; so that

$$\left\{(x,G(x)),\forall x\in\mathbb{F}_2^n\right\} = \left[\begin{array}{cc}A^{-1} & 0\\CA^{-1} & B\end{array}\right]\left(\left\{(x,F(x)),\forall x\in\mathbb{F}_2^n\right\}\right) .$$

Definition and Basic Theorems How Can sboxU Help?

Some Algorithmic Problems with CCZ-Equivalence



Definition and Basic Theorems How Can sboxU Help?

Some Algorithmic Problems with CCZ-Equivalence

_	EA-class	EA-class	EA-class	EA-class	EA-class
Γ					
	F				
	F				

Definition and Basic Theorems How Can sboxU Help?

Some Algorithmic Problems with CCZ-Equivalence

EA-class	EA-class	EA-class	EA-class	EA-class
F	F ₁	F ₂		F4
			F ₃	

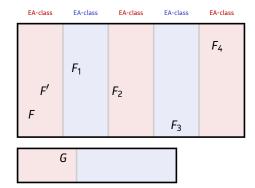
Definition and Basic Theorems How Can sboxU Help?

Some Algorithmic Problems with CCZ-Equivalence

F1	EA-class	EA-class	EA-class	EA-class	EA-class
F' F2 F F3	F4	F3	F ₂	<i>F</i> ₁	

Definition and Basic Theorems How Can sboxU Help?

Some Algorithmic Problems with CCZ-Equivalence



Definition and Basic Theorems How Can sboxU Help?

Plan of this Section

1 Basic Functionalities

2 CCZ-Equivalence

Definition and Basic Theorems

How Can sboxU Help?

3 Ortho-Derivative

4 Conclusion

Definition and Basic Theorems How Can sboxU Help?

Exploring a CCZ-class



Definition and Basic Theorems How Can sboxU Help?

Class Invariants

Definition (Differential spectrum)

Recall that $DDT_F[a, b] = \#\{x, F(x + a) + F(x) = b\}$. The differential spectrum is the number of occurrences of each number in the DDT.

Definition and Basic Theorems How Can sboxU Help?

Class Invariants

Definition (Differential spectrum)

Recall that $DDT_F[a, b] = \#\{x, F(x + a) + F(x) = b\}$. The differential spectrum is the number of occurrences of each number in the DDT.

Definition (Walsh spectrum)

Recall that $\mathcal{W}_{F}[a, b] = \sum_{x} (-1)^{a \cdot x + b \cdot F(x)}$. The Walsh spectrum is the number of occurrences of each number in the LAT. The extended Walsh spectrum considers only absolute values.

Definition and Basic Theorems How Can sboxU Help?

Class Invariants

Definition (Differential spectrum)

Recall that $DDT_F[a, b] = \#\{x, F(x + a) + F(x) = b\}$. The differential spectrum is the number of occurrences of each number in the DDT.

Definition (Walsh spectrum)

Recall that $\mathcal{W}_{F}[a, b] = \sum_{x} (-1)^{a \cdot x + b \cdot F(x)}$. The Walsh spectrum is the number of occurrences of each number in the LAT. The extended Walsh spectrum considers only absolute values.

- Differential and extended Walsh spectra are constant in a CCZ-class.
- The algebraic degree and the thickness spectrum are constant in an EA-class.



Definition and Basic Theorems Algorithmic Uses Inverting the DDT of a Quadratic Function

Plan of this Section

1 Basic Functionalities

2 CCZ-Equivalence

3 Ortho-Derivative

- Definition and Basic Theorems
- Algorithmic Uses
- Inverting the DDT of a Quadratic Function

4 Conclusion

Definition and Basic Theorems Algorithmic Uses Inverting the DDT of a Quadratic Function

Plan of this Section

1 Basic Functionalities

2 CCZ-Equivalence

3 Ortho-Derivative

Definition and Basic Theorems

- Algorithmic Uses
- Inverting the DDT of a Quadratic Function

4 Conclusion

Definition and Basic Theorems Algorithmic Uses Inverting the DDT of a Quadratic Function

Definition

Definition

Ortho-Derivative Let F be a quadratic function of \mathbb{F}_2^n . The ortho-derivatives of F are the functions of \mathbb{F}_2^n such that

$$\forall x \in \mathbb{F}_2^n, \ \pi_F(a) \cdot \left(\underbrace{F(x+a)+F(x)}_{\Delta_a F(x)}+F(a)+F(0)\right) = 0.$$

Definition and Basic Theorems Algorithmic Uses Inverting the DDT of a Quadratic Function

Definition

Definition

Ortho-Derivative Let F be a quadratic function of \mathbb{F}_2^n . The ortho-derivatives of F are the functions of \mathbb{F}_2^n such that

$$\forall x \in \mathbb{F}_2^n, \ \pi_F(a) \cdot \left(\underbrace{F(x+a) + F(x)}_{\Delta_a F(x)} + F(a) + F(0)\right) = 0 \ .$$

π_F(a) is orthogonal to the linear part of the hyperplane Im(Δ_aF)
 π_F can take any value in 0.

Definition and Basic Theorems Algorithmic Uses Inverting the DDT of a Quadratic Function

Basic Properties

Lemma (Ortho-derivatives of APN functions)

F is APN if and only if $\pi_F(a)$ is uniquely defined for all $a \in (\mathbb{F}_2^n)^*$.

¹See also A note on the properties of associated Boolean functions of quadratic APN functions by Anastasiya Gorodilova on ArXiv.

Definition and Basic Theorems Algorithmic Uses Inverting the DDT of a Quadratic Function

Basic Properties

Lemma (Ortho-derivatives of APN functions)

F is APN if and only if $\pi_F(a)$ is uniquely defined for all $a \in (\mathbb{F}_2^n)^*$.

Lemma (Interaction with EA-equivalence)

If $G = B \circ F \circ A$ where A and B are linear permutations, then

$$\pi_F = (B^T)^{-1} \circ F \circ A$$

¹See also A note on the properties of associated Boolean functions of quadratic APN functions by Anastasiya Gorodilova on ArXiv.

Definition and Basic Theorems Algorithmic Uses Inverting the DDT of a Quadratic Function

Basic Properties

Lemma (Ortho-derivatives of APN functions)

F is APN if and only if $\pi_F(a)$ is uniquely defined for all $a \in (\mathbb{F}_2^n)^*$.

Lemma (Interaction with EA-equivalence)

If $G = B \circ F \circ A$ where A and B are linear permutations, then

 $\pi_{\mathbf{F}} = (B^{\mathsf{T}})^{-1} \circ \mathbf{F} \circ A$

It seems like¹ the algebraic degree of the ortho-derivative of an APN function is always n - 2.

¹See also A note on the properties of associated Boolean functions of quadratic APN functions by Anastasiya Gorodilova on ArXiv.

Definition and Basic Theorems Algorithmic Uses Inverting the DDT of a Quadratic Function

Preimages of the Ortho-Derivative

Theorem

Linear Structures (APN case) If

$$T_F(b) = \left\{ x \in \mathbb{F}_2^n : \pi_F(x) = b
ight\},$$

then
$$T_F(b) = LS(x \mapsto b \cdot F(x)).$$

Corollary

For any b, $T_F(b)$ is a linear subspace of \mathbb{F}_2^n whose dimension has the same parity as n. Furthermore,

$$\left(\mathcal{W}_{F}[a, b]\right)^{2} \in \left\{0, 2^{n+\dim T_{F}(b)}\right\}$$

Definition and Basic Theorems Algorithmic Uses Inverting the DDT of a Quadratic Function

Plan of this Section

1 Basic Functionalities

2 CCZ-Equivalence

3 Ortho-Derivative

Definition and Basic Theorems

Algorithmic Uses

Inverting the DDT of a Quadratic Function

4 Conclusion

Definition and Basic Theorems Algorithmic Uses Inverting the DDT of a Quadratic Function

Identifying EA- and CCZ-classes

Corollary (Ortho-derivatives of APN functions)

The differential and extended Walsh spectra of the ortho-derivative of an APN function is the same within an EA-class.

Definition and Basic Theorems Algorithmic Uses Inverting the DDT of a Quadratic Function

Identifying EA- and CCZ-classes

Corollary (Ortho-derivatives of APN functions)

The differential and extended Walsh spectra of the ortho-derivative of an APN function is the same within an EA-class.

Observation

In practice, these spectra differ from one EA-class to the next!

Definition and Basic Theorems Algorithmic Uses Inverting the DDT of a Quadratic Function

Identifying EA- and CCZ-classes

Corollary (Ortho-derivatives of APN functions)

The differential and extended Walsh spectra of the ortho-derivative of an APN function is the same within an EA-class.

Observation

In practice, these spectra differ from one EA-class to the next!

We can use this to very efficiently sort large numbers of quadratic functions into distinct EA-classes.



Definition and Basic Theorems Algorithmic Uses Inverting the DDT of a Quadratic Function

Plan of this Section

1 Basic Functionalities

2 CCZ-Equivalence

3 Ortho-Derivative

- Definition and Basic Theorems
- Algorithmic Uses
- Inverting the DDT of a Quadratic Function

4 Conclusion

Definition and Basic Theorems Algorithmic Uses Inverting the DDT of a Quadratic Function

Principle

Is it possible to recover *F* given π_F ?

Definition and Basic Theorems Algorithmic Uses Inverting the DDT of a Quadratic Function

Principle

Is it possible to recover F given π_F ? Yes!

The Key Observation

We can write the scalar product $x \cdot y$ as $(\vec{x})^T \times \vec{y}$, where \times is a matrix operation.

Definition and Basic Theorems Algorithmic Uses Inverting the DDT of a Quadratic Function

.

Principle

Is it possible to recover F given π_F ? Yes!

The Key Observation

We can write the scalar product $x \cdot y$ as $(\vec{x})^T \times \vec{y}$, where \times is a matrix operation.

We represent *F* as a vector of $\mathbb{F}_2^{n2^n}$ by concatenating the *n*-bit representation of each of the 2^{*n*} values *F*(*x*):

$$\operatorname{vec}(F) = \begin{bmatrix} F_0(0) \\ F_1(0) \\ \cdots \\ F_{n-1}(0) \\ F_0(1) \\ \cdots \\ F_{n-1}(2^n - 1) \end{bmatrix}$$

Definition and Basic Theorems Algorithmic Uses Inverting the DDT of a Quadratic Function

Re-Defining Ortho-Derivatives

Let G be a function and $\zeta_a(G)$ be a matrix defined by

1 $\zeta_{G}(a)[x,x] = G(\vec{a})^{T}, \qquad \zeta_{G}(a)[x,x+a] = G(\vec{a})^{T},$ **2** $\zeta_{G}(a)[x,0] = G(\vec{a})^{T}, \qquad \zeta_{G}(a)[x,a] = G(\vec{a})^{T},$

Definition and Basic Theorems Algorithmic Uses Inverting the DDT of a Quadratic Function

Re-Defining Ortho-Derivatives

Let G be a function and $\zeta_a(G)$ be a matrix defined by

1
$$\zeta_{G}(a)[x,x] = \vec{G(a)}^{T}, \qquad \zeta_{G}(a)[x,x+a] = \vec{G(a)}^{T},$$

2 $\zeta_{G}(a)[x,0] = \vec{G(a)}^{T}, \qquad \zeta_{G}(a)[x,a] = \vec{G(a)}^{T},$

so that

Definition and Basic Theorems Algorithmic Uses Inverting the DDT of a Quadratic Function

Re-Defining Ortho-Derivatives

Let G be a function and $\zeta_a(G)$ be a matrix defined by

1 $\zeta_{G}(a)[x,x] = G(\vec{a})^{T}, \qquad \zeta_{G}(a)[x,x+a] = G(\vec{a})^{T},$ **2** $\zeta_{G}(a)[x,0] = G(\vec{a})^{T}, \qquad \zeta_{G}(a)[x,a] = G(\vec{a})^{T},$

so that

$$\zeta_{G}(a) \times \operatorname{vec}(F) = \begin{bmatrix} \frac{G(a) \cdot (F(0) + F(0 + a) + F(a) + F(0))}{G(a) \cdot (F(1) + F(1 + F(1) + F(a) + F(1))} \\ \dots \\ G(a) \cdot (F(2^{n} - 1) + F(2^{n} - 1 + a) + F(a) + F(2^{n} - 1)) \end{bmatrix},$$

from which we deduce that if π_F is an ortho-derivative of F then

$$\mathsf{vec}(F) \in \ker (\zeta(\pi_F))$$
 where $\zeta(\pi_F) = \begin{bmatrix} \zeta_0(\pi_F) \\ \dots \\ \zeta_{2^n-1}(\pi_F) \end{bmatrix}$

.

Definition and Basic Theorems Algorithmic Uses Inverting the DDT of a Quadratic Function

Inverting the DDT of a Quadratic Function

- Find a DDT,
- **2** deduce the corresponding π ,
- **B** build $\zeta(\pi)$,
- 4 find ker $(\zeta(\pi))$,
- obtain vec(F)!

²Tricks are used to get rid of redundancies in ζ , and trivial solutions.

Definition and Basic Theorems Algorithmic Uses Inverting the DDT of a Quadratic Function

Inverting the DDT of a Quadratic Function

- Find a DDT,
- **2** deduce the corresponding π ,
- **B** build $\zeta(\pi)$,
- 4 find ker $(\zeta(\pi))$,
- obtain vec(F)!

In practice, starting from "cleverly" built functions π yields $\zeta(\pi)$ with empty² kernels...

²Tricks are used to get rid of redundancies in ζ , and trivial solutions.

Plan of this Section

1 Basic Functionalities

2 CCZ-Equivalence

3 Ortho-Derivative

4 Conclusion

Conclusion

Go an use sboxU!

Conclusion

Go an use sboxU!

Thank you!