S-Box Decompositions and some Applications

Léo Perrin

January 28, 2019, Nancy

Curriculum

- **Currently**: post-doc at SECRET in Inria Paris
- **PhD**: University of Luxembourg (symmetric cryptography)
- **Masters**: double degree Centrale Lyon/KTH
(discrete math/theoretical CS)

Outline

Outline

Symmetric Cryptography

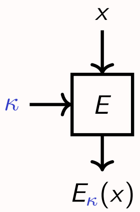
We assume that a secret key has already been shared!

Symmetric Cryptography

We assume that a secret key has already been shared!

Definition (Block Cipher)

- Input: n-bit block x
- Parameter: k-bit key κ
- Output: n-bit block $E_\kappa(x)$
- Symmetry: E and E^{-1} use the same κ

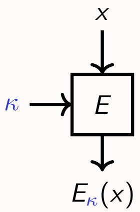


Symmetric Cryptography

We assume that a secret key has already been shared!

Definition (Block Cipher)

- Input: n-bit block x
- Parameter: k-bit key κ
- Output: n-bit block $E_\kappa(x)$
- Symmetry: E and E^{-1} use the same κ



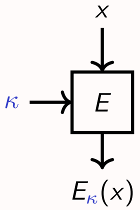
No Key Recovery. Given many pairs $(x, E_\kappa(x))$, it must be impossible to recover κ.

Symmetric Cryptography

We assume that a secret key has already been shared!

Definition (Block Cipher)

- Input: n-bit block x
- Parameter: k-bit key κ
- Output: n-bit block $E_\kappa(x)$
- Symmetry: E and E^{-1} use the same κ



No Key Recovery. Given many pairs $(x, E_\kappa(x))$, it must be impossible to recover κ.

Indistinguishability. Given an n permutation P, it must be impossible to figure out if $P = E_\kappa$ for some κ.

Security Arguments

The Specification

Contains a full design rationale, meaning we can trust the cipher because:

- we trust the security arguments of the designer
- we have a starting point for cryptanalysis

Security Arguments

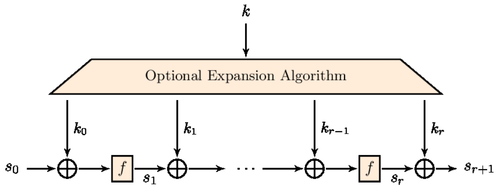
The Specification

Does not contain a full design rationale, meaning we **cannot** trust the cipher because:

- we have to start cryptanalysis from scratch
- what are they trying to hide?

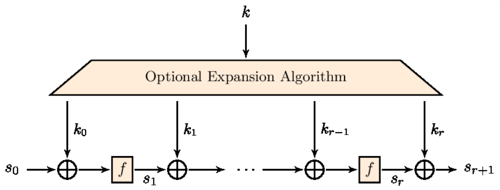
To Build a Cipher

Iterated Construction

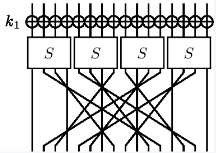


To Build a Cipher

Iterated Construction



Two different sub-components for f



Linear layer (diffusion)

S-box layer (non-linearity)

The S-box

π' = (252, 238, 221, 17, 207, 110, 49, 22, 251, 196, 250, 218, 35, 197, 4, 77, 233, 119, 240, 219, 147, 46, 153, 186, 23, 54, 241. 187, 20, 205, 95, 193, 249, 24, 101, 90, 226, 92, 239, 33, 129, 28, 60, 66, 139, 1, 142, 79, 5, 132, 2, 174, 227, 106, 143, 160, 6, 11, 237, 152, 127, 212, 211, 31, 235, 52, 44, 81, 234, 200, 72, 171, 242, 42, 104, 162, 253, 58, 206, 204, 181, 112, 14, 86, 8, 12, 118, 18, 191, 114, 19, 71, 156, 183, 93, 135, 21, 161, 150, 41, 16, 123, 154, 199, 243, 145, 120, 111, 157, 158, 178, 177, 50, 117, 25, 61, 255, 53, 138, 126, 109, 84, 198, 128, 195, 189, 13, 87, 223, 245, 36, 169, 62, 168, 67, 201, 215, 121, 214, 246, 124, 34, 185, 3, 224, 15, 236, 222, 122, 148, 176, 188, 220, 232, 40, 80, 78, 51, 10, 74, 167, 151, 96, 115, 30, 0, 98, 68, 26, 184, 56, 130, 100, 159, 38, 65, 173, 69, 70, 146, 39, 94, 85, 47, 140, 163, 165, 125, 105, 213, 149, 59, 7, 88, 179, 64, 134, 172, 29, 247, 48, 55, 107, 228, 136, 217, 231, 137, 225, 27, 131, 73, 76, 63, 248, 254, 141, 83, 170, 144, 202, 216, 133, 97, 32, 113, 103, 164, 45, 43, 9, 91, 203, 155, 37, 208, 190, 229, 108, 82, 89, 166, 116, 210, 230, 244, 180, 192, 209, 102, 175, 194, 57, 75, 99, 182).

The S-box

π' = (252, 238, 221, 17, 207, 110, 49, 22, 251, 196, 250, 218, 35, 197, 4, 77, 233, 119, 240, 219, 147, 46, 153, 186, 23, 54, 241. 187, 20, 205, 95, 193, 249, 24, 101, 90, 226, 92, 239, 33, 129, 28, 60, 66, 139, 1, 142, 79, 5, 132, 2, 174, 227, 106, 143, 160, 6, 11, 237, 152, 127, 212, 211, 31, 235, 52, 44, 81, 234, 200, 72, 171, 242, 42, 104, 162, 253, 58, 206, 204, 181, 112, 14, 86, 8, 12, 118, 18, 191, 114, 19, 71, 156, 183, 93, 135, 21, 161, 150, 41, 16, 123, 154, 199, 243, 145, 120, 111, 157, 158, 178, 177, 50, 117, 25, 61, 255, 53, 138, 126, 109, 84, 198, 128, 195, 189, 13, 87, 223, 245, 36, 169, 62, 168, 67, 201, 215, 121, 214, 246, 124, 34, 185, 3, 224, 15, 236, 222, 122, 148, 176, 188, 220, 232, 40, 80, 78, 51, 10, 74, 167, 151, 96, 115, 30, 0, 98, 68, 26, 184, 56, 130, 100, 159, 38, 65, 173, 69, 70, 146, 39, 94, 85, 47, 140, 163, 165, 125, 105, 213, 149, 59, 7, 88, 179, 64, 134, 172, 29, 247, 48, 55, 107, 228, 136, 217, 231, 137, 225, 27, 131, 73, 76, 63, 248, 254, 141, 83, 170, 144, 202, 216, 133, 97, 32, 113, 103, 164, 45, 43, 9, 91, 203, 155, 37, 208, 190, 229, 108, 82, 89, 166, 116, 210, 230, 244, 180, 192, 209, 102, 175, 194, 57, 75, 99, 182).

Importance of the S-box

If S is such that the maximum number of x such that

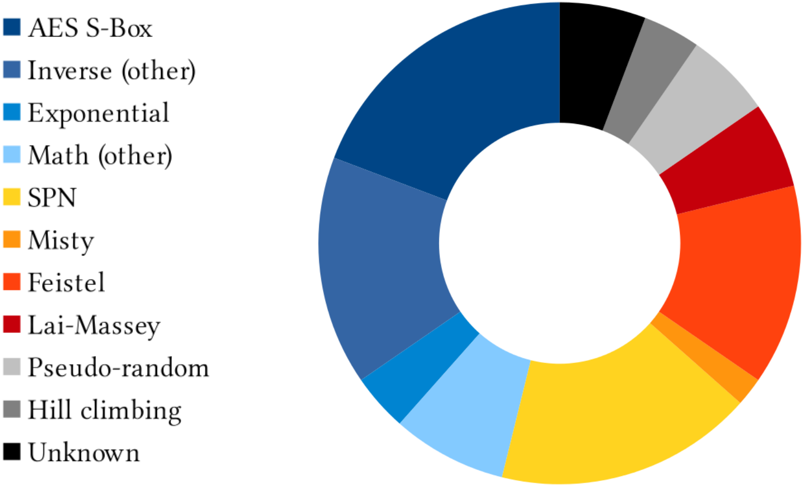
$$S(x) \oplus S(x \oplus a) = b$$

is low for all $a \neq 0$ and b then the cipher may be proved secure against differential attacks.

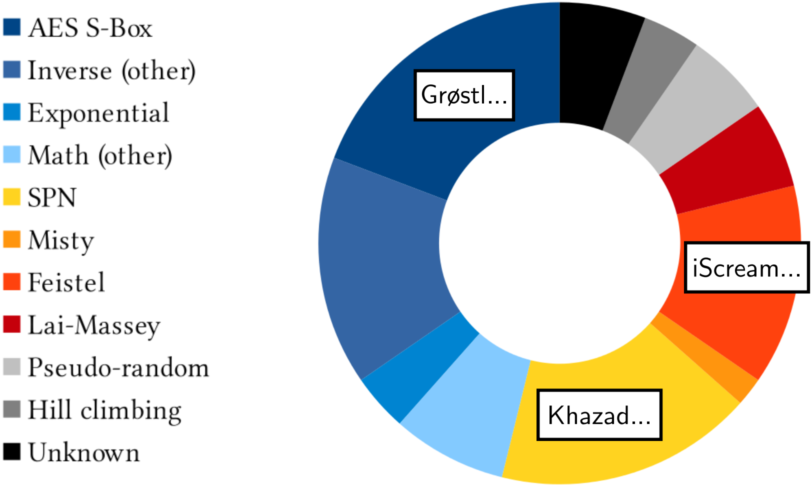
S-box Design

- AES S-Box
- Inverse (other)
- Exponential
- Math (other)
- SPN
- Misty
- Feistel
- Lai-Massey
- Pseudo-random
- Hill climbing
- Unknown

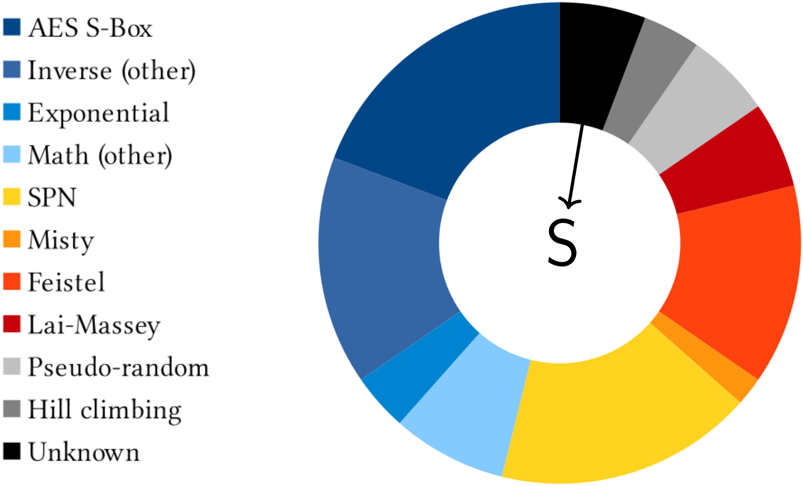
S-box Design


AES S-Box
Inverse (other)
Exponential
Math (other)
SPN
Misty
Feistel
Lai-Massey
Pseudo-random
Hill climbing
Unknown

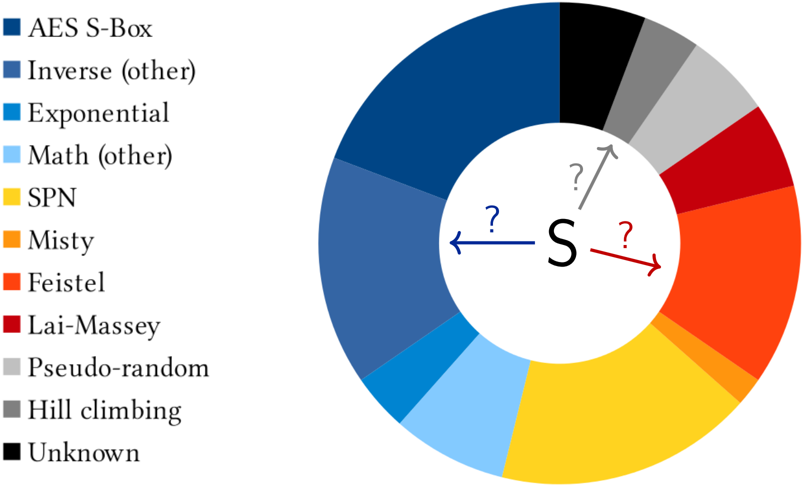
S-box Design


AES S-Box
Inverse (other)
Exponential
Math (other)
SPN
Misty
Feistel
Lai-Massey
Pseudo-random
Hill climbing
Unknown
Grøstl...
iScream...
Khazad...

S-box Reverse-Engineering



S-box Reverse-Engineering



Why Reverse-Engineer S-boxes? (1/3)

A malicious designer can hide a structure in an S-box.

Why Reverse-Engineer S-boxes? (1/3)

A malicious designer can hide a structure in an S-box.

To keep an advantage in implementation (white-box crypto)...

Why Reverse-Engineer S-boxes? (1/3)

A malicious designer can hide a structure in an S-box.

To keep an advantage in implementation (white-box crypto)...
... or an advantage in cryptanalysis (backdoor).

Dual EC: A Standardized Back Door

Daniel J. Bernstein[1,2], Tanja Lange[1], and Ruben Niederhagen[1]

eprint report 2015/767

Why Reverse-Engineer S-boxes? (2/3)

S-box based backdoors in the literature

- Rijmen, V., & Preneel, B. (1997). *A family of trapdoor ciphers*. FSE'97.
- Paterson, K. (1999). *Imprimitive Permutation Groups and Trapdoors in Iterated Block Ciphers*. FSE'99.
- Blondeau, C., Civino, R., & Sala, M. (2017). *Differential Attacks: Using Alternative Operations*. `eprint` report 2017/610.
- Bannier, A., & Filiol, E. (2017). *Partition-based trapdoor ciphers. In Partition-Based Trapdoor Ciphers*. InTech'17.

Why Reverse-Engineer S-boxes? (3/3)

Even without malicious intent, an unexpected structure can be a problem.

$\Longrightarrow$ We need tools to **reverse-engineer** S-boxes!

Design and Analysis

Analysis

- GLUON-64 hash function (FSE'14)
- PRINCE block cipher (FSE'15)
- TWINE block cipher (FSE'15)

Design

- SPARX block cipher (Asiacrypt'16)
- SPARKLE permutation, ESCH hash function, SCHWAEMM authenticated cipher (NIST submission)
- Purposefully *hard* functions (Asiacrypt'17)
- MOE block cipher (submitted to EC)

S-box Reverse-Engineering

When the S-box has a BC structure

Feistel network (SAC'15, FSE'16), SPN (ToSC'17)

When it doesn't

- Analysis of Skipjack (Crypto'15)
- Structures in the Russian S-box (Eurocrypt'16, ToSC'17, ToSC'19)
- Cryptanalysis of a Theorem (Crypto'16, IEEE Trans. Inf. Th.'17, FFA'19, CC'19)

S-box Reverse-Engineering

When the S-box has a BC structure

Feistel network (SAC'15, FSE'16), SPN (ToSC'17)

When it doesn't

- Analysis of Skipjack (Crypto'15)
- **Structures in the Russian S-box** (Eurocrypt'16, ToSC'17, ToSC'19)
- **Cryptanalysis of a Theorem** (Crypto'16, IEEE Trans. Inf. Th.'17, FFA'19, CC'19)

Outline

Outline

We can recover an actual decomposition using patterns in the LAT.

1. TU-decomposition: what is it and how to apply it?
2. First results on the Russian S-box
3. Its intended decomposition (I think)

Kuznyechik/Streebog

Streebog

Type	Hash function
Publication	2012

Kuznyechik

Type	Block cipher
Publication	2015

Kuznyechik/Streebog

Streebog

Type	Hash function
Publication	2012

Kuznyechik

Type	Block cipher
Publication	2015

Common ground

- Both are standard symmetric primitives in Russia.
- Both were designed by the FSB (TC26).
- Both use the same 8×8 S-box, π.

Basic Tools for Analysing S-boxes

Let $S : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$ be an S-box.

Basic Tools for Analysing S-boxes

Let $S : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$ be an S-box.

Definition (DDT)

The *Difference Distribution Table* of S is a matrix of size $2^n \times 2^n$ such that

$$\mathrm{DDT}[a, b] = \#\{x \in \mathbb{F}_2^n \mid S(x \oplus a) \oplus S(x) = b\}.$$

Basic Tools for Analysing S-boxes

Let $S : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be an S-box.

Definition (DDT)

The *Difference Distribution Table* of S is a matrix of size $2^n \times 2^n$ such that

$$\text{DDT}[a, b] \;=\; \#\{x \in \mathbb{F}_2^n \mid S(x \oplus a) \oplus S(x) = b\}.$$

Definition (LAT)

The *Linear Approximations Table* of S is a matrix of size $2^n \times 2^n$ such that

$$\text{LAT}[a, b] \;=\; \sum_{x \in \mathbb{F}_2^n} (-1)^{x \cdot a \oplus S(x) \cdot b} \;.$$

Example

$$S = [4, 2, 1, 6, 0, 5, 7, 3]$$

The DDT of S.

$$\begin{bmatrix} 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 \\ 0 & 0 & 4 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 \\ 0 & 4 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 \end{bmatrix}$$

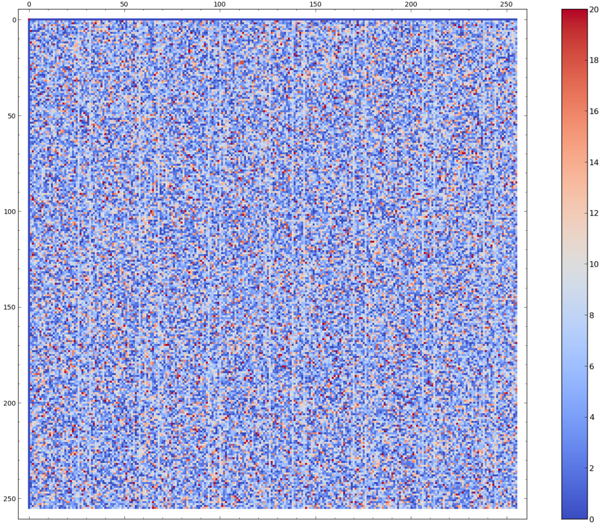
$$\#\{x \in \mathbb{F}_2^n \mid S(x \oplus a) \oplus S(x) = b\}$$

The LAT of S.

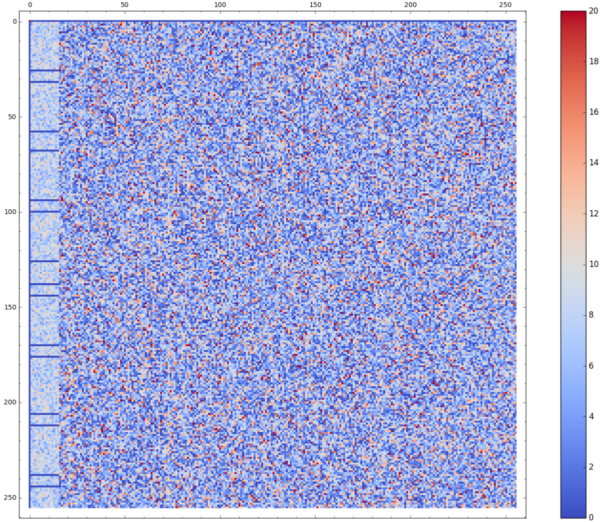
$$\begin{bmatrix} 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 4 & 0 & 0 & 4 & -4 \\ 0 & 4 & 4 & 0 & 0 & 4 & -4 & 0 \\ 0 & 4 & 0 & 4 & 0 & -4 & 0 & 4 \\ 0 & 4 & 0 & -4 & 0 & -4 & 0 & -4 \\ 0 & -4 & 4 & 0 & 0 & -4 & -4 & 0 \\ 0 & 0 & -4 & 4 & 0 & 0 & -4 & -4 \\ 0 & 0 & 0 & 0 & -8 & 0 & 0 & 0 \end{bmatrix}$$

$$\sum_{x \in \mathbb{F}_2^n} (-1)^{x \cdot a \oplus S(x) \cdot b} .$$

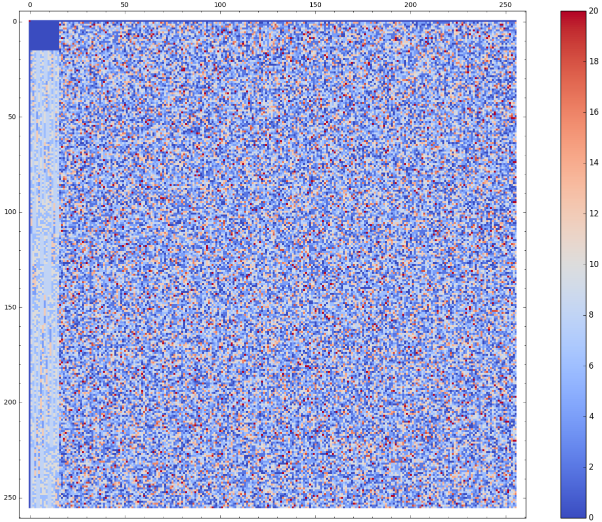
The LAT of π



The LAT of π (reordered columns)



The LAT of $\eta \circ \pi \circ \mu$



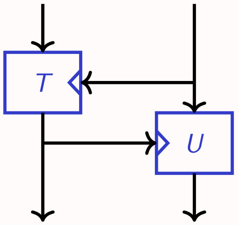
The TU-Decomposition

Definition

The TU-decomposition is a decomposition algorithm working against S-boxes with vector spaces of zeroes in their LAT.

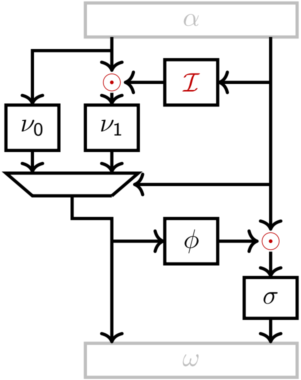
Theorem

"Square of zeroes" in the LAT. $\Leftrightarrow$



- *T and U are mini-block ciphers*
- *μ and η are linear permutations.*

First Complete Decomposition of π [BPU16]



$\odot$ Multiplication in $\mathbb{F}_{2^4}$

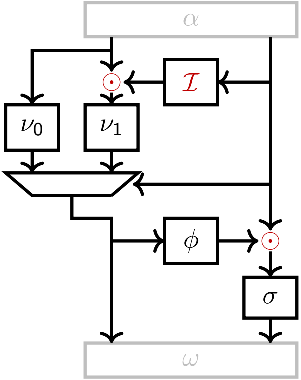
$\mathcal{I}$ Inversion in $\mathbb{F}_{2^4}$

ν_0, ν_1, σ 4×4 permutations

ϕ 4×4 function

α, ω Linear permutations

First Complete Decomposition of π [BPU16]



- $\odot$ Multiplication in $\mathbb{F}_{2^4}$
- $\mathcal{I}$ Inversion in $\mathbb{F}_{2^4}$
- ν_0, ν_1, σ 4×4 permutations
- ϕ 4×4 function
- α, ω Linear permutations

Ugly, but it would not be there if π were random.

Hardware Performance

Structure	Area (μm^2)	Delay (ns)
Naive implementation	3889.6	362.52
With TU-decomposition	1530.1	46.11

Knowledge of this decomposition divides:

- the area by 2.5, and
- the delay by 8

Conclusion for Kuznyechik/Streebog?

The Russian S-box was built with a TU-decomposition...

Conclusion for Kuznyechik/Streebog?

The Russian S-box was built with a TU-decomposition...

... or was it?

Reopening a Cold Case (Twice)

Detour through Belarus [PU16]

We identified some similar properties between π and the S-box of the standard of Belarus... Which turned out to be based on a discrete logarithm.

Reopening a Cold Case (Twice)

Detour through Belarus [PU16]

We identified some similar properties between π and the S-box of the standard of Belarus... Which turned out to be based on a discrete logarithm.

New Patterns [Per18]

$$\pi\left(0 \oplus \langle 01, 0a, 44, 92\rangle\right) = c8 \oplus \langle 02, 04, 10, 20\rangle$$
$$\pi\left(0 \oplus \langle 05, 22, 49, 8b\rangle\right) = 20 \oplus \langle 01, 0a, 44, 92\rangle\ .$$

Reopening a Cold Case (Twice)

Detour through Belarus [PU16]

We identified some similar properties between π and the S-box of the standard of Belarus... Which turned out to be based on a discrete logarithm.

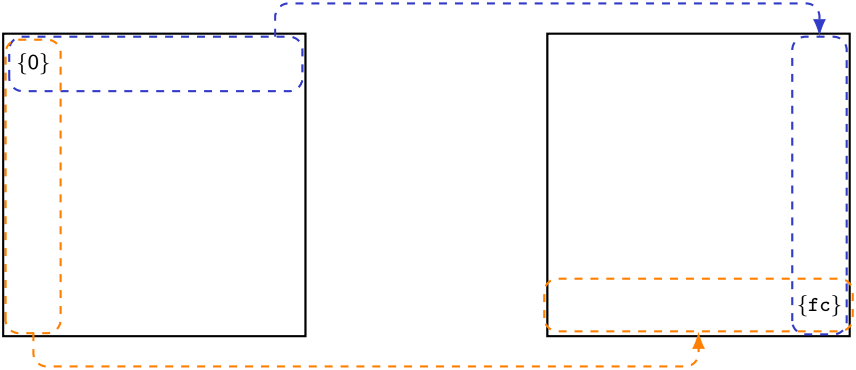
New Patterns [Per18]

$$\pi\left(0 \oplus \langle 01, 0a, 44, 92\rangle\right) = c8 \oplus \langle 02, 04, 10, 20\rangle$$
$$\pi\left(0 \oplus \langle 05, 22, 49, 8b\rangle\right) = 20 \oplus \langle 01, 0a, 44, 92\rangle \ .$$

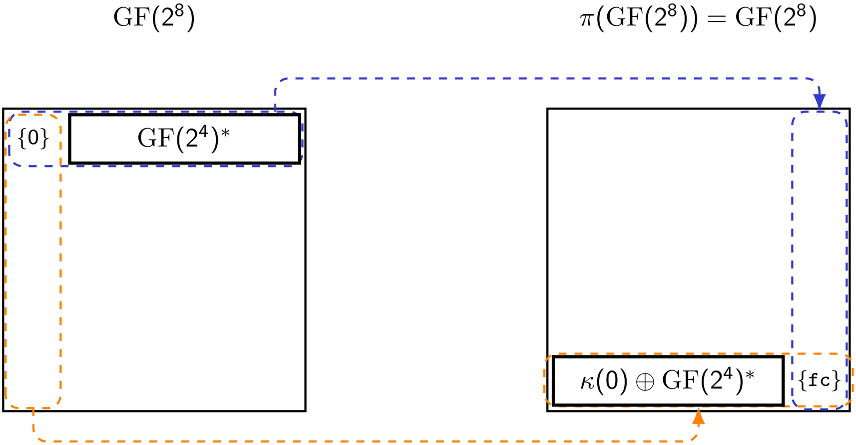
- $\langle 01, 0a, 44, 92\rangle \ \oplus \ \langle 05, 22, 49, 8b\rangle \ = \ \mathbb{F}_2^8$
- $(c8 \oplus \langle 05, 22, 49, 8b\rangle) \ \oplus \ (20 \oplus \langle 01, 0a, 44, 92\rangle) \ = \ \mathbb{F}_2^8$
- $(c8 \oplus \langle 05, 22, 49, 8b\rangle) \ \cap \ (20 \oplus \langle 01, 0a, 44, 92\rangle) \ = \ \pi(0) = \texttt{fc}$

Cosets to Cosets

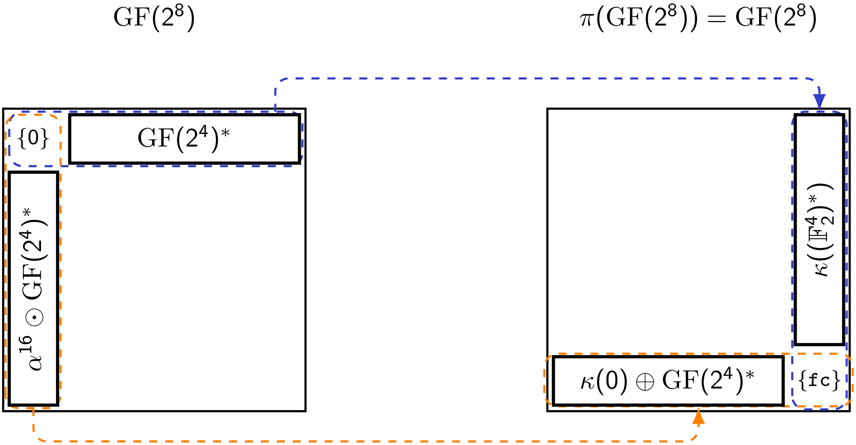
$\mathrm{GF}(2^8)$ $\pi(\mathrm{GF}(2^8)) = \mathrm{GF}(2^8)$



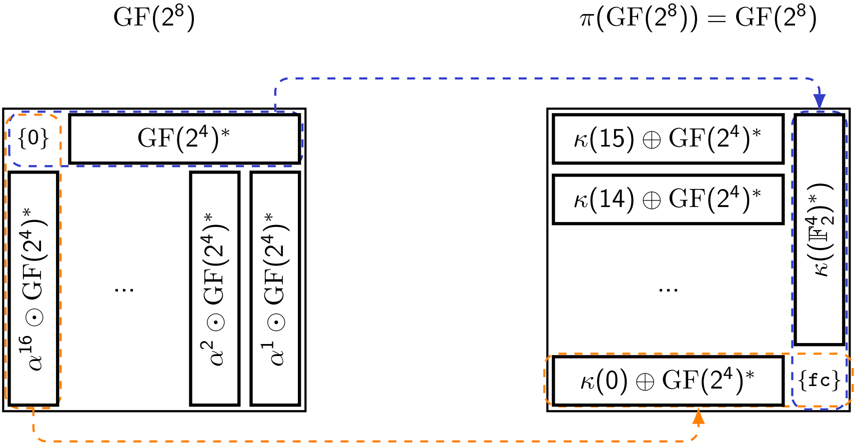
Cosets to Cosets



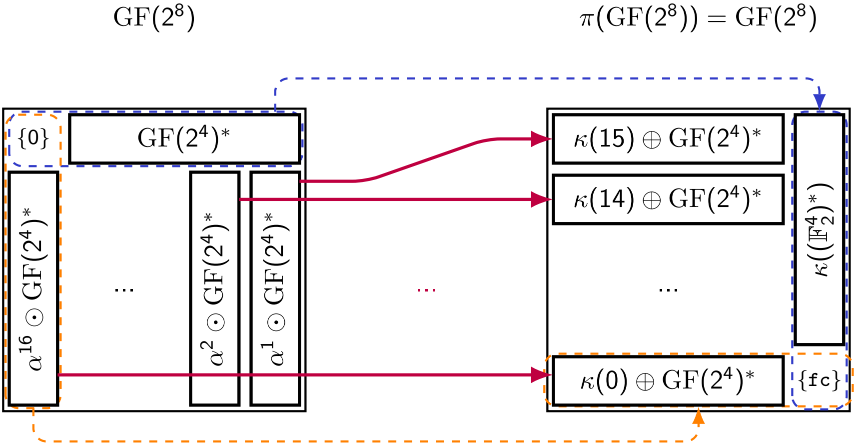
Cosets to Cosets



Cosets to Cosets



Cosets to Cosets



The TKlog [Per18]

A TKlog operates on $\mathrm{GF}(2^{2m})$ and uses:

- α: a generator of $\mathrm{GF}(2^{2m})$,
- κ: an affine function $\mathbb{F}_2^m \to \mathrm{GF}(2^{2m})$ with $\kappa(\mathbb{F}_2^m) \oplus \mathrm{GF}(2^m) = \mathrm{GF}(2^{2m})$,
- s: a permutation of $\mathbb{Z}/(2^m - 1)\mathbb{Z}$.

The corresponding TKlog is denoted $\mathscr{T}_{\kappa,s}$ and it works as follows:

$$\begin{cases} \mathscr{T}_{\kappa,s}(0) & = \kappa(0) \ , \\ \mathscr{T}_{\kappa,s}\left((\alpha^{2^m+1})^j\right) & = \kappa(2^m - j), \quad \text{for } \ 1 \le j \le 2^m - 1 \ , \\ \mathscr{T}_{\kappa,s}\left(\alpha^{i+(2^m+1)j}\right) & = \kappa(2^m - i) \oplus \left(\alpha^{2^m+1}\right)^{s(j)}, \ \text{for } 0 < i, 0 \le j < 2^m - 1 \ . \end{cases}$$

Case of π

- $p = X^8 + X^4 + X^3 + X^2 + 1$,
- $s = [0, 12, 9, 8, 7, 4, 14, 6, 5, 10, 2, 11, 1, 3, 13]$,
- $\kappa(x) = \Lambda(x) \oplus \texttt{0xfc}$,
- $\Lambda(1) = \texttt{0x12}$, $\Lambda(2) = \texttt{0x26}$, $\Lambda(4) = \texttt{0x24}$, $\Lambda(8) = \texttt{0x30}$

Case of π

- $p = X^8 + X^4 + X^3 + X^2 + 1$,
- $s = [0, 12, 9, 8, 7, 4, 14, 6, 5, 10, 2, 11, 1, 3, 13]$,
- $\kappa(x) = \Lambda(x) \oplus \texttt{0xfc}$,
- $\Lambda(1) = \texttt{0x12}$, $\Lambda(2) = \texttt{0x26}$, $\Lambda(4) = \texttt{0x24}$, $\Lambda(8) = \texttt{0x30}$

$$\#\text{TKlogs} = \underbrace{16}_{p} \times \underbrace{15!}_{s} \times \underbrace{\prod_{i=4}^{7}(2^8 - 2^i)}_{\Lambda} \times \underbrace{2^8}_{\kappa(0)} \approx 2^{82.6}$$

$$\#\text{8-bit perm.} = 2^{1684} \; ; \; \#\text{Affine perm.} = \underbrace{2^8}_{\text{cstte}} \times \underbrace{\prod_{i=0}^{7}(2^8 - 2^i)}_{\text{linear part}} \approx 2^{70.2} \, .$$

The Linear Layer of Streebog (1/2)

5.4 Линейное преобразование множества двоичных векторов

Линейное преобразование l множества двоичных векторов V_{64} задается умножением справа на матрицу A над полем $GF(2)$, строки которой записаны ниже последовательно в шестнадцатеричном виде. Строка матрицы с номером j, $j = 0,...,63$, записанная в виде $a_{j,15} ... a_{j,0}$, где $a_{j,i} \in \mathbb{Z}_{16}$, $i = 0, ..., 15$, есть $\mathrm{Vec}_4(a_{j,15}) \| ... \| \mathrm{Vec}_4(a_{j,0})$.

8e20faa72ba0b470	47107ddd9b505a38	ad08b0e0c3282d1c	d8045870ef14980e
6c022c38f90a4c07	3601161cf205268d	1b8e0b0e798c13c8	83478b07b2468764
a011d380818e8f40	5086e740ce47c920	2843fd2067adea10	14aff010bdd87508
0ad97808d06cb404	05e23c0468365a02	8c711e02341b2d01	46b60f011a83988e
90dab52a387ae76f	486dd4151c3dfdb9	24b86a840e90f0d2	125c354207487869
092e94218d243cba	8a174a9ec8121e5d	4585254f64090fa0	accc9ca9328a8950
9d4df05d5f661451	c0a878a0a1330aa6	60543c50de970553	302a1e286fc58ca7
18150f14b9ec46dd	0c84890ad27623e0	0642ca05693b9f70	0321658cba93c138
86275df09ce8aaa8	439da0784e745554	afc0503c273aa42a	d960281e9d1d5215
e230140fc0802984	71180a8960409a42	b60c05ca30204d21	5b068c651810a89e
456c34887a3805b9	ac361a443d1c8cd2	561b0d22900e4669	2b838811480723ba
9bcf4486248d9f5d	c3e9224312c8c1a0	effa11af0964ee50	f97d86d98a327728
e4fa2054a80b329c	727d102a548b194e	39b008152acb8227	9258048415eb419d
492c024284fbaec0	aa16012142f35760	550b8e9e21f7a530	a48b474f9ef5dc18
70a6a56e2440598e	3853dc371220a247	1ca76e95091051ad	0edd37c48a08a6d8
07e095624504536c	8d70c431ac02a736	c83862965601dd1b	641c314b2b8ee083

The Linear Layer of Streebog (2/2)

It is actually a matrix multiplication in $\mathrm{GF}(2^8)$:

$$\begin{bmatrix} 83 & 47 & 8b & 07 & b2 & 46 & 87 & 64 \\ 46 & b6 & 0f & 01 & 1a & 83 & 98 & 8e \\ ac & cc & 9c & a9 & 32 & 8a & 89 & 50 \\ 03 & 21 & 65 & 8c & ba & 93 & c1 & 38 \\ 5b & 06 & 8c & 65 & 18 & 10 & a8 & 9e \\ f9 & 7d & 86 & d9 & 8a & 32 & 77 & 28 \\ a4 & 8b & 47 & 4f & 9e & f5 & dc & 18 \\ 64 & 1c & 31 & 4b & 2b & 8e & e0 & 83 \end{bmatrix}.$$

The Linear Layer of Streebog (2/2)

It is actually a matrix multiplication in $\mathrm{GF}(2^8)$:

$$\begin{bmatrix} 83 & 47 & 8b & 07 & b2 & 46 & 87 & 64 \\ 46 & b6 & 0f & 01 & 1a & 83 & 98 & 8e \\ ac & cc & 9c & a9 & 32 & 8a & 89 & 50 \\ 03 & 21 & 65 & 8c & ba & 93 & c1 & 38 \\ 5b & 06 & 8c & 65 & 18 & 10 & a8 & 9e \\ f9 & 7d & 86 & d9 & 8a & 32 & 77 & 28 \\ a4 & 8b & 47 & 4f & 9e & f5 & dc & 18 \\ 64 & 1c & 31 & 4b & 2b & 8e & e0 & 83 \end{bmatrix}.$$

The polynomial used is the same as in π.

The Linear Layer of Streebog (2/2)

It is actually a matrix multiplication in $\mathrm{GF}(2^8)$:

$$\begin{bmatrix} 83 & 47 & 8b & 07 & b2 & 46 & 87 & 64 \\ 46 & b6 & 0f & 01 & 1a & 83 & 98 & 8e \\ ac & cc & 9c & a9 & 32 & 8a & 89 & 50 \\ 03 & 21 & 65 & 8c & ba & 93 & c1 & 38 \\ 5b & 06 & 8c & 65 & 18 & 10 & a8 & 9e \\ f9 & 7d & 86 & d9 & 8a & 32 & 77 & 28 \\ a4 & 8b & 47 & 4f & 9e & f5 & dc & 18 \\ 64 & 1c & 31 & 4b & 2b & 8e & e0 & 83 \end{bmatrix}.$$

The polynomial used is the same as in π.

A new security analysis is badly needed!

The Linear Layer of Streebog (2/2)

It is actually a matrix multiplication in $\mathrm{GF}(2^8)$:

$$\begin{bmatrix} 83 & 47 & 8b & 07 & b2 & 46 & 87 & 64 \\ 46 & b6 & 0f & 01 & 1a & 83 & 98 & 8e \\ ac & cc & 9c & a9 & 32 & 8a & 89 & 50 \\ 03 & 21 & 65 & 8c & ba & 93 & c1 & 38 \\ 5b & 06 & 8c & 65 & 18 & 10 & a8 & 9e \\ f9 & 7d & 86 & d9 & 8a & 32 & 77 & 28 \\ a4 & 8b & 47 & 4f & 9e & f5 & dc & 18 \\ 64 & 1c & 31 & 4b & 2b & 8e & e0 & 83 \end{bmatrix}.$$

The polynomial used is the same as in π.

A new security analysis is badly needed!

Reverse-engineering works!

Outline

Outline

We can obtain new mathematical results using decompositions.

1. The big APN problem and its only known solutions
2. Decomposing and generalizing this solution as butterflies
3. Generalizing a property of butterflies

The Big APN Problem

Definition (APN function)

A function $S : \mathbb{F}_2^n \to \mathbb{F}_2^n$ is Almost Perfect Non-linear (APN) if

$$S(x \oplus a) \oplus S(x) = b$$

has 0 or 2 solutions for all $a \neq 0$ and for all b.

The Big APN Problem

Definition (APN function)

A function $S : \mathbb{F}_2^n \to \mathbb{F}_2^n$ is Almost Perfect Non-linear (APN) if

$$S(x \oplus a) \oplus S(x) = b$$

has 0 or 2 solutions for all $a \neq 0$ and for all b.

Big APN Problem

Are there APN permutations operating on $\mathbb{F}_2^n$ where n is even? [NK95]

Dillon et al.'s Permutation

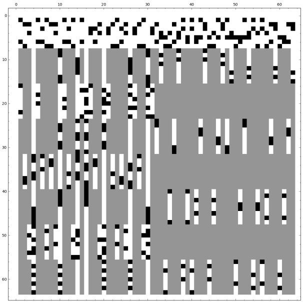
Only One Known Solution!

For $n = 6$, Dillon et al. [BDKM09] found an APN permutation.

Dillon et al.'s Permutation

Only One Known Solution!

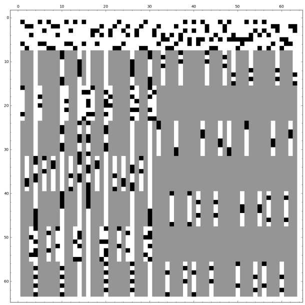
For $n = 6$, Dillon et al. [BDKM09] found an APN permutation.

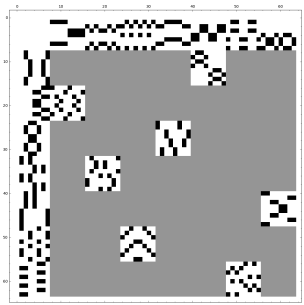


Dillon et al.'s Permutation

Only One Known Solution!

For $n = 6$, Dillon et al. [BDKM09] found an APN permutation.

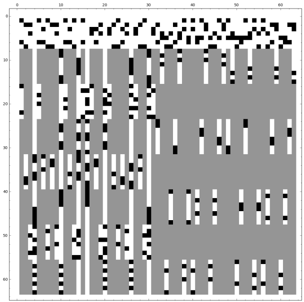


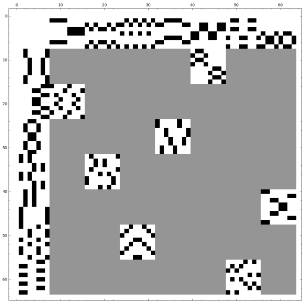


Dillon et al.'s Permutation

Only One Known Solution!

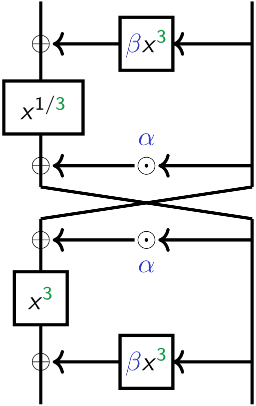
For $n = 6$, Dillon et al. [BDKM09] found an APN permutation.





It is possible to make a TU-decomposition! [PUB16]

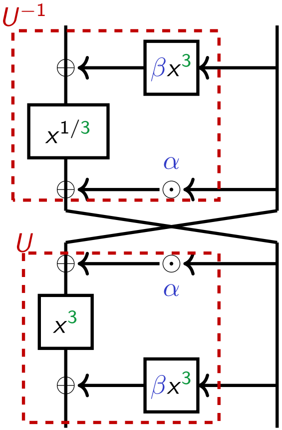
On the Butterfly Structure



Definition (Open Butterfly $H^3_{\alpha,\beta}$)

This permutation is an open butterfly [PUB16].

On the Butterfly Structure



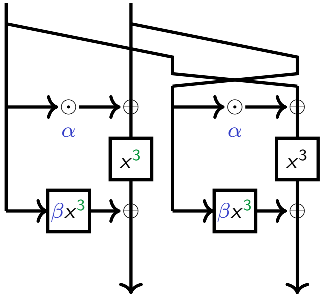
Definition (Open Butterfly $H^3_{\alpha,\beta}$)

This permutation is an open butterfly [PUB16].

Lemma

Dillon's permutation is affine-equivalent to $H^3_{w,1}$, where $Tr(w) = 0$.

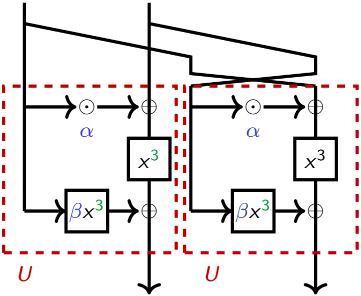
Closed Butterflies



Definition (Closed butterfly $V^3_{\alpha,\beta}$)

This quadratic function is a closed butterfly.

Closed Butterflies



Definition (Closed butterfly $V^3_{\alpha,\beta}$)

This quadratic function is a closed butterfly.

Lemma (Equivalence)

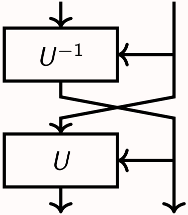
Open and closed butterflies with the same parameters are CCZ-equivalent.

Properties of Butterflies

Let $n \leq 3$ be odd. Butterflies...

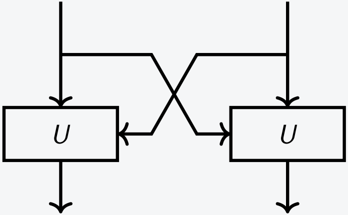
- ... are APN but only for $n = 3$ [CDP17, CPT18]
- ... are differentially-4 (the best) for $n > 3$
- ... have the best non-linearity
- ... are rather cheap to implement

Open Butterfly



$2n$-bit permutation.
Algebraic degree n (or $n+1$).

Closed Butterfly



$2n$-bit function for $n \leq 3$ odd.
Algebraic degree 2.

Equivalence Relations (1/2)

Definition (Affine-Equivalence)

F and G are *affine equivalent* if $G(x) = (B \circ F \circ A)(x)$, where A, B are affine permutations.

Equivalence Relations (1/2)

Definition (Affine-Equivalence)

F and G are *affine equivalent* if $G(x) = (B \circ F \circ A)(x)$, where A, B are affine permutations.

Equivalently, we need to have

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

Equivalence Relations (2/2)

Definition (CCZ-Equivalence [CCZ98])

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

$$\Gamma_G = \{(x, G(x)), \forall x \in \mathbb{F}_2^n\} = \mathcal{L}\left(\{(x, F(x)), \forall x \in \mathbb{F}_2^n\}\right) = \mathcal{L}(\Gamma_F) ,$$

where $\mathcal{L} : \mathbb{F}_2^{n+m} \rightarrow \mathbb{F}_2^{n+m}$ is an affine permutation.
For example, F and F^{-1} are CCZ-equivalent.

Equivalence Relations (2/2)

Definition (CCZ-Equivalence [CCZ98])

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

$$\Gamma_G = \left\{(x, G(x)), \forall x \in \mathbb{F}_2^n\right\} \ = \ \mathcal{L}\left(\left\{(x, F(x)), \forall x \in \mathbb{F}_2^n\right\}\right) = \mathcal{L}(\Gamma_F) \ ,$$

where $\mathcal{L} : \mathbb{F}_2^{n+m} \rightarrow \mathbb{F}_2^{n+m}$ is an affine permutation.
For example, F and F^{-1} are CCZ-equivalent.

CCZ-equivalence preserves some properties (differential and linear) but **not** others (algebraic degree).

Equivalence Relations (2/2)

Definition (CCZ-Equivalence [CCZ98])

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

$$\Gamma_G = \left\{(x, G(x)), \forall x \in \mathbb{F}_2^n\right\} = \mathcal{L}\left(\left\{(x, F(x)), \forall x \in \mathbb{F}_2^n\right\}\right) = \mathcal{L}(\Gamma_F) ,$$

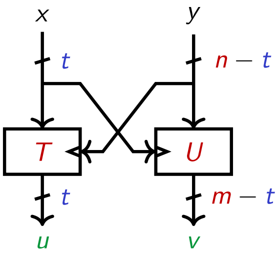
where $\mathcal{L} : \mathbb{F}_2^{n+m} \to \mathbb{F}_2^{n+m}$ is an affine permutation.
For example, F and F^{-1} are CCZ-equivalent.

CCZ-equivalence preserves some properties (differential and linear) but **not** others (algebraic degree).

The TU-decomposition plays a crucial role in CCZ-equivalence.

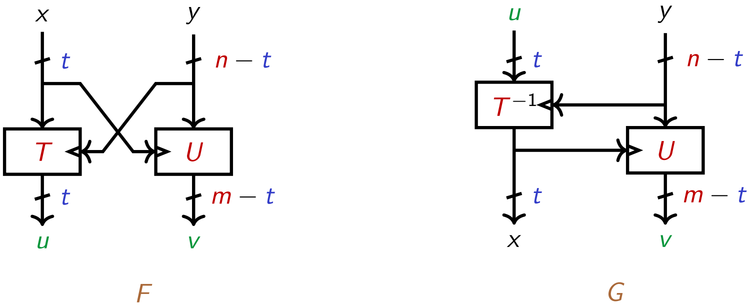
Twist

Any function $F : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$ can be projected on $\mathbb{F}_2^t \times \mathbb{F}_2^{m-t}$.



Twist

Any function $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ can be projected on $\mathbb{F}_2^t \times \mathbb{F}_2^{m-t}$.



If T is a permutation for all secondary inputs, then we define the t-twist equivalent of F as G, where

$$G(x, y) = \left(T_y^{-1}(x), U_{T_y^{-1}(x)}(y)\right)$$

for all $(x, y) \in \mathbb{F}_2^t \times \mathbb{F}_2^{n-t}$.

TU-Decomposition and CCZ-Equivalence

Theorem ([CP19])

If F and G are CCZ-equivalent then either their equivalence is trivial or it involves a t-twist.

TU-Decomposition and CCZ-Equivalence

Theorem ([CP19])

If F and G are CCZ-equivalent then either their equivalence is trivial or it involves a t-twist.

In other words, if F is non-trivially CCZ-equivalent to something else then it must have a TU-decomposition!

Outline

Conclusion

Decompositions play a crucial role in cryptography!

- When designing
- When implementing
- When attacking

Conclusion

Decompositions play a crucial role in cryptography!

- When designing
- When implementing
- When attacking

They allow us to bring cryptographic techniques to other fields of mathematics.

Open Problems (Symmetric Cryptography)

Russian Shenanigans

Is it possible to use the latest decomposition of the Russian S-box to attack the corresponding algorithms?

Open Problems (Symmetric Cryptography)

Russian Shenanigans

Is it possible to use the latest decomposition of the Russian S-box to attack the corresponding algorithms?

DES

What are the decompositions in the S-boxes of the DES (that we don't know of)? Could we use them in attacks?

Open Problems (Discrete Mathematics)

TU-decomposition in GF

The TU-decomposition and the twist are defined over $\mathbb{F}_2^n$. Can we find a nice representation over $\mathrm{GF}(2^n)$?

Open Problems (Discrete Mathematics)

TU-decomposition in GF

The TU-decomposition and the twist are defined over $\mathbb{F}_2^n$. Can we find a nice representation over $\mathrm{GF}(2^n)$?

Big APN Problem

Is there an APN permutation of an even number of bits ($n \geq 8$)?

Open Problems (Discrete Mathematics)

TU-decomposition in GF

The TU-decomposition and the twist are defined over $\mathbb{F}_2^n$. Can we find a nice representation over $\mathrm{GF}(2^n)$?

Big APN Problem

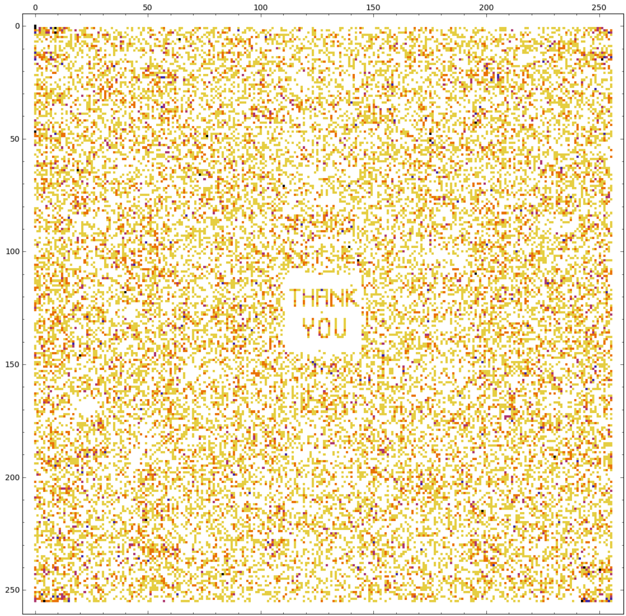
Is there an APN permutation of an even number of bits ($n \geq 8$)?

Other Decomposition

Are there other decompositions as general as the TU-decomposition? Are other mathematical structures explained by an underlying decomposition?

The Last S-Box

14	11	60	6d	e9	10	e3	2	b	90	d	17	c5	b0	9f	c5
d8	da	be	22	8	f3	4	a9	fe	f3	f5	fc	bc	30	be	26
bb	88	85	46	f4	2e	e	fd	76	fe	b0	11	4e	de	35	bb
30	4b	30	d6	dd	df	df	d4	90	7a	d8	8c	6a	89	30	39
e9	1	da	d2	85	87	d3	d4	ba	2b	d4	9f	9c	38	8c	55
d3	86	bb	db	ec	e0	46	48	bf	46	1b	1c	d7	d9	1b	e0
23	d4	d7	7f	16	3f	3	3	44	c3	59	10	2a	da	ed	e9
8e	d8	d1	db	cb	cb	c3	c7	38	22	34	3d	db	85	23	7c
24	d1	d8	2e	fc	44	8	38	c8	c7	39	4c	5f	56	2a	cf
d0	e9	d2	68	e4	e3	e9	13	e2	c	97	e4	60	29	d7	9b
d9	16	24	94	b3	e3	4c	4c	4f	39	e0	4b	bc	2c	d3	94
81	96	93	84	91	d0	2e	d6	d2	2b	78	ef	d6	9e	7b	72
ad	c4	68	92	7a	d2	5	2b	1e	d0	dc	b1	22	3f	c3	c3
88	b1	8d	b5	e3	4e	d7	81	3	15	17	25	4e	65	88	4e
e4	3b	81	81	fa	1	1d	4	22	0	6	1	27	68	27	2e
3b	83	c7	cc	25	9b	d8	d5	1c	1f	e5	59	7f	3f	3f	ef

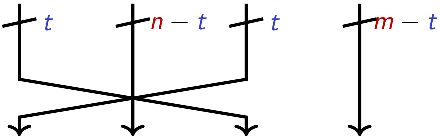

THANK
YOU

Swap Matrices

The swap matrix permuting $\mathbb{F}_2^{n+m}$ is defined for $t \leq \min(n, m)$ as

$$M_t = \begin{bmatrix} 0 & 0 & I_t & 0 \\ 0 & I_{n-t} & 0 & 0 \\ I_t & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{m-t} \end{bmatrix} .$$

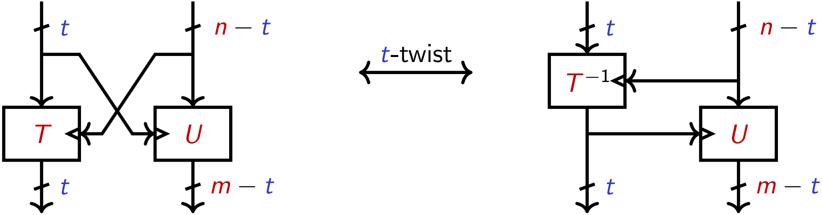
It has a simple interpretation:



For all $t \leq \min(n, m)$, M_t is an **orthogonal** and **symmetric involution**.

Swap Matrices and Twisting

$\mathbf{F} : \mathbb{F}_2^{\mathbf{n}} \rightarrow \mathbb{F}_2^{\mathbf{m}}$ $\mathbf{G} : \mathbb{F}_2^{\mathbf{n}} \rightarrow \mathbb{F}_2^{\mathbf{m}}$



$$\Gamma_F = \left\{ (x, F(x)), \forall x \in \mathbb{F}_2^n \right\} \quad \xleftrightarrow{M_t} \quad \Gamma_G = \left\{ (x, G(x)), \forall x \in \mathbb{F}_2^n \right\}$$

$$\mathcal{W}_F(u) = \mathcal{W}_G(M_t(u))$$

Twisting and CCZ-Class

Lemma

Twisting preserves the CCZ-equivalence class.

Main Result

Theorem

If $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ and $G : \mathbb{F}_2^n \to \mathbb{F}_2^m$ are CCZ-equivalent, then

$$\Gamma_G = (B \times M_t \times A)(\Gamma_F) ,$$

where A and B are EA-mappings and where

$$t = \dim\left(proj_{\mathcal{V}^\perp}\left((A^T \times M_t \times B^T)(\mathcal{V})\right)\right) .$$

Corollary

If a function is CCZ-equivalent but not EA-equivalent to another function, then they have to be EA-equivalent to functions for which a t-twist is possible.

K. A. Browning, J.F. Dillon, R.E. Kibler, and M. T. McQuistan.
APN Polynomials and Related Codes.
J. of Combinatorics, Information and System Sciences, 34(1-4):135–159, 2009.

Alex Biryukov, Léo Perrin, and Aleksei Udovenko.
Reverse-engineering the S-box of streebog, kuznyechik and STRIBOBr1.
In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part I*, volume 9665 of *LNCS*, pages 372–402. Springer, Heidelberg, May 2016.

Claude Carlet, Pascale Charpin, and Victor Zinoviev.
Codes, bent functions and permutations suitable for DES-like cryptosystems.
Designs, Codes and Cryptography, 15(2):125–156, 1998.

Anne Canteaut, Sébastien Duval, and Léo Perrin.
A generalisation of Dillon's APN permutation with the best known differential and nonlinear properties for all fields of size 2^{4k+2}.

IEEE Transactions on Information Theory, 63(11):7575–7591, Nov 2017.

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

Anne Canteaut, Léo Perrin, and Shizhu Tian.
If a generalised butterfly is APN then it operates on 6 bits.
Cryptology ePrint Archive, Report 2018/1036, 2018.
https://eprint.iacr.org/2018/1036.

Kaisa Nyberg and Lars R. Knudsen.
Provable security against a differential attack.
Journal of Cryptology, 8(1):27–37, 1995.

Léo Perrin.
Partitions in the S-box of Streebog and Kuznyechik.
To appear (IACR ToSC), 2018.

Léo Perrin and Aleksei Udovenko.

Exponential s-boxes: a link between the s-boxes of BelT and Kuznyechik/Streebog.
IACR Trans. Symm. Cryptol., 2016(2):99–124, 2016.
http://tosc.iacr.org/index.php/ToSC/article/view/567.

Léo Perrin, Aleksei Udovenko, and Alex Biryukov.
Cryptanalysis of a theorem: Decomposing the only known solution to the big APN problem.
In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 93–122. Springer, Heidelberg, August 2016.