On S-Box Reverse-Engineering: from Cryptanalysis to the Big APN Problem

Léo Perrin

DTU, Lyngby
perrin dot leo at gmail

4th of July 2017
Boolean Functions and Their Applications

The content of this talk is based on joint works with Biryukov, Canteaut, Duval, Khovratovich and Udovenko, and my PhD thesis.

If you only know the Look-Up Table of an S-Box, what can you do?

If you only know the Look-Up Table of an S-Box, what can you do?

Random?

Was it picked uniformly at random?

If you only know the Look-Up Table of an S-Box, what can you do?

Random?

Was it picked uniformly at random?

Structured?

Was it built using a particular structure ?

S-Box?

An S-Box is a small non-linear function mapping m bits to n usually specified via its look-up table.

S-Box?

An S-Box is a small non-linear function mapping m bits to n usually specified via its look-up table.

- Typically, $n = m, n \in \{4, 8\}$
- Used by many block ciphers/hash functions/stream ciphers.
- Necessary for the wide trail strategy.

Example

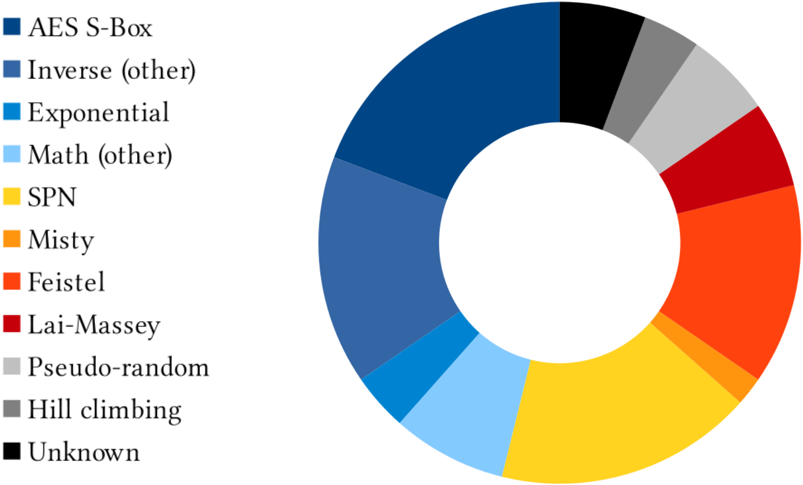
π' = (252, 238, 221, 17, 207, 110, 49, 22, 251, 196, 250, 218, 35, 197, 4, 77, 233, 119, 240, 219, 147, 46, 153, 186, 23, 54, 241. 187, 20, 205, 95, 193, 249, 24, 101, 90, 226, 92, 239, 33, 129, 28, 60, 66, 139, 1, 142, 79, 5, 132, 2, 174, 227, 106, 143, 160, 6, 11, 237, 152, 127, 212, 211, 31, 235, 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).

Screen capture from [GOST, 2015].

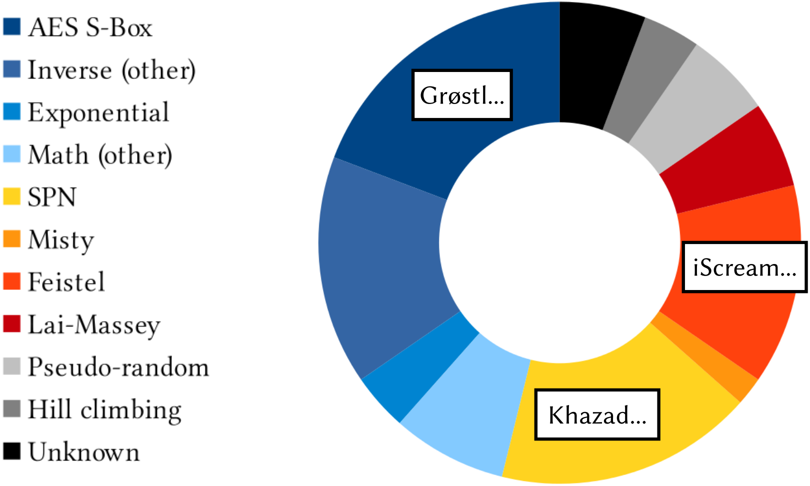
S-Box Design

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

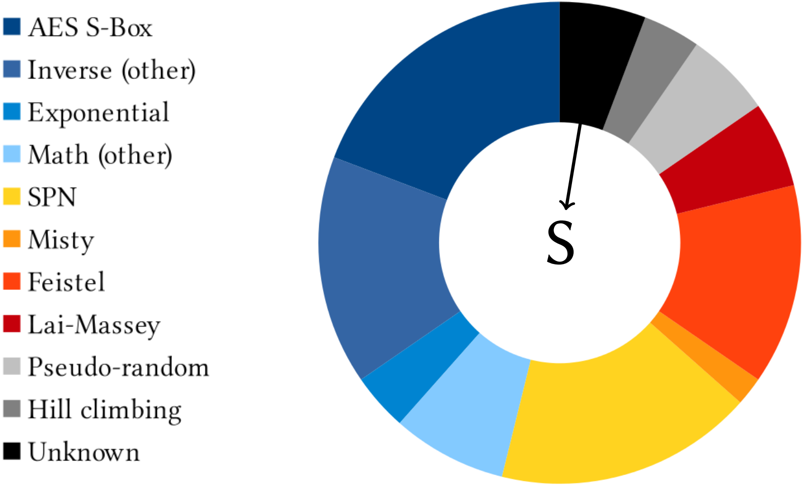
S-Box Design


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

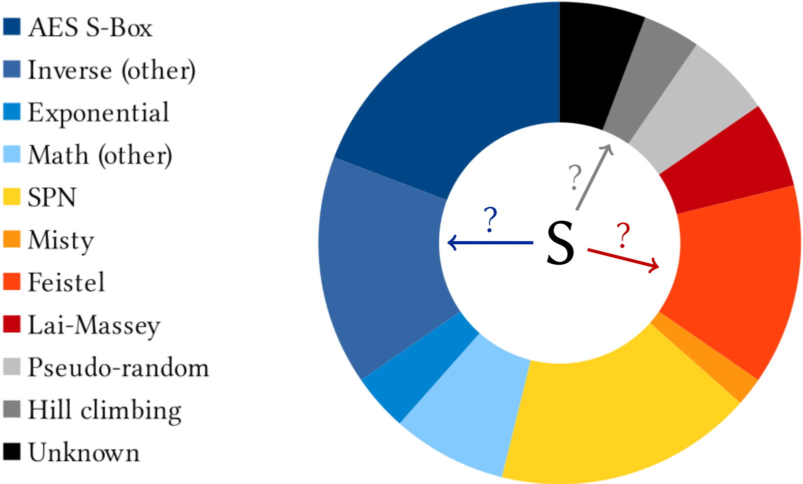
S-Box Design


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



Motivation

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

Motivation

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

To keep an advantage in implementation (WB crypto)...

Motivation

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

To keep an advantage in implementation (WB crypto)...
... or an advantage in cryptanalysis (backdoor)?

Outline

Plan of this Section

The Two Tables

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

The Two Tables

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

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

The Two Tables

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

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

Definition (LAT)

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

$$\mathrm{LAT}[a, b] \;=\; \#\{x \in \mathbb{F}_2^n \mid x \cdot a = S(x) \cdot b\} - 2^{n-1} \;=\; \frac{\mathcal{W}_S(a, b)}{2}$$

Coefficient Distribution in the DDT

If an n-bit S-Box is bijective, then its DDT coefficients behave like independent and identically distributed random variables following a Poisson distribution:

$$\Pr\left[\mathrm{DDT}[a,b] = 2z\right] = \frac{e^{-1/2}}{2^{z} z!} .$$

Coefficient Distribution in the DDT

If an n-bit S-Box is bijective, then its DDT coefficients behave like independent and identically distributed random variables following a Poisson distribution:

$$\Pr\left[\mathrm{DDT}[a,b] = 2z\right] = \frac{e^{-1/2}}{2^z z} .$$

- Always even, ≥ 0
- Typically between 0 and 16 (for $n =$)
- Lower is better.

Coefficient Distribution in the LAT

If an n-bit S-Box is bijective, then its LAT coefficients behave like independent and identically distributed random variables following this distribution:

$$\Pr\left[\mathrm{LAT}[a,b] = 2z\right] = \frac{\binom{2^{n-1}}{2^{n-2+z}}}{\binom{2^n}{2^{n-1}}} .$$

Coefficient Distribution in the LAT

If an n-bit S-Box is bijective, then its LAT coefficients behave like independent and identically distributed random variables following this distribution:

$$\Pr\left[\mathrm{LAT}[a,b] = 2z\right] = \frac{\binom{2^{n-1}}{2^{n-2+z}}}{\binom{2^n}{2^{n-1}}}.$$

- Always even, signed.
- Typically between -40 and 40 (for $n = 8$).
- Lower absolute value is better.

Looking Only at the Maximum

δ	$\log_2 (\Pr [\max(\mathcal{D}) \leq \delta])$
14	-0.006
12	-0.094
10	-1.329
8	-16.148
6	-164.466
4	-1359.530

DDT

ℓ	$\log_2 (\Pr [\max(\mathcal{L}) \leq \ell])$
38	-0.084
36	-0.302
34	-1.008
32	-3.160
30	-9.288
28	-25.623
26	-66.415
24	-161.900
22	-371.609

LAT

Probability that the maximum coefficient in the DDT/LAT of an 8-bit permutation is at most equal to a certain threshold.

Looking Only at the Maximum

δ	$\log_2(\Pr[\max(\mathcal{D}) \leq \delta])$
14	-0.006
12	-0.094
10	-1.329
8	-16.148
6	-164.466
4	-1359.530

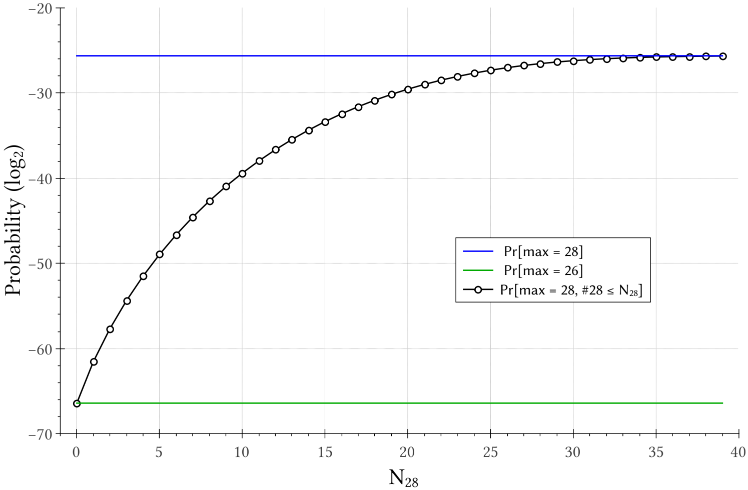
DDT

ℓ	$\log_2(\Pr[\max(\mathcal{L}) \leq \ell])$
38	-0.084
36	-0.302
34	-1.008
32	-3.160
30	-9.288
28	-25.623
26	-66.415
24	-161.900
22	-371.609

LAT

Probability that the maximum coefficient in the DDT/LAT of an 8-bit permutation is at most equal to a certain threshold.

Taking Number of Maximum Values into Account

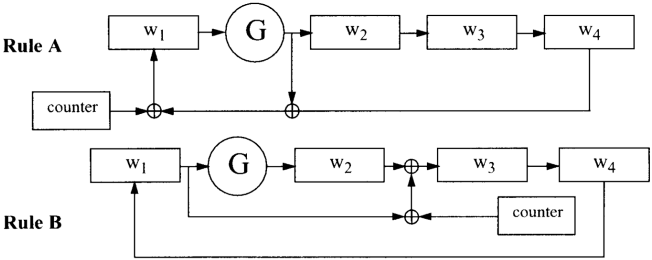


Application of this Analysis?

We applied this method on the S-Box of Skipjack.

What is Skipjack?

Type	Block cipher
Bloc	64 bits
Key	80 bits
Authors	NSA
Publication	1998 (classified at first)

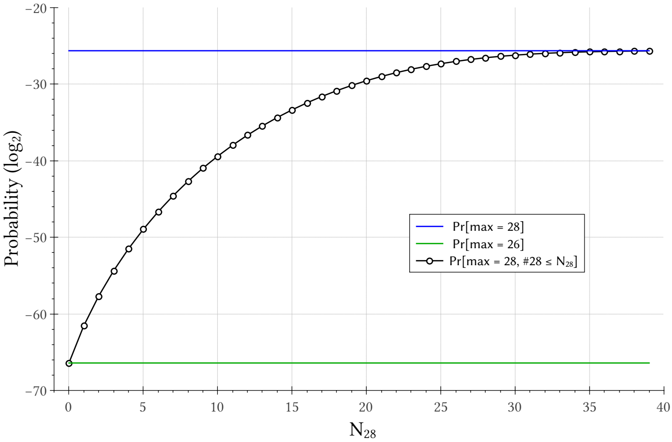


Reverse-Engineering the S-Box of Skipjack

Skipjack uses F, a permutation of $\mathbb{F}_2^8$ with $\max(\text{LAT}) = 28$ and $\#28 = 3$.

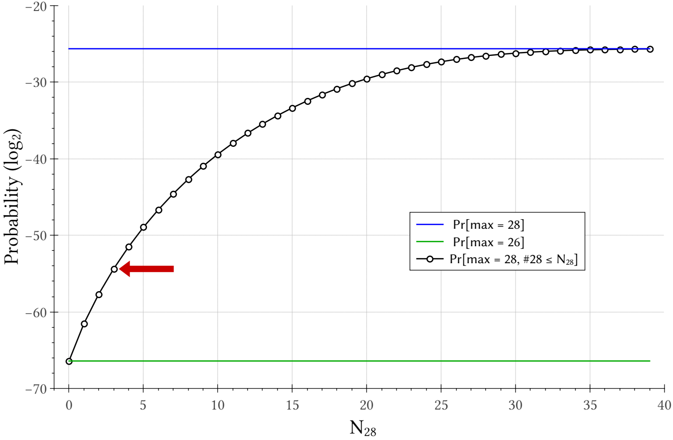
Reverse-Engineering the S-Box of Skipjack

Skipjack uses F, a permutation of $\mathbb{F}_2^8$ with max(LAT) = 28 and #28 = 3.



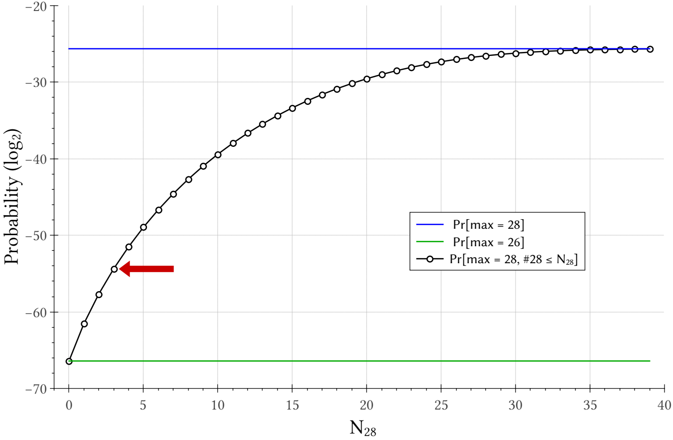
Reverse-Engineering the S-Box of Skipjack

Skipjack uses F, a permutation of $\mathbb{F}_2^8$ with max(LAT) = 28 and #28 = 3.



Reverse-Engineering the S-Box of Skipjack

Skipjack uses F, a permutation of $\mathbb{F}_2^8$ with $\max(\text{LAT}) = 28$ and $\#28 = 3$.



$$\Pr\left[\max(\text{LAT}) = 28 \text{ and } \#28 \leq 3\right] \approx \mathbf{2^{-55}}$$

What Can We Deduce?

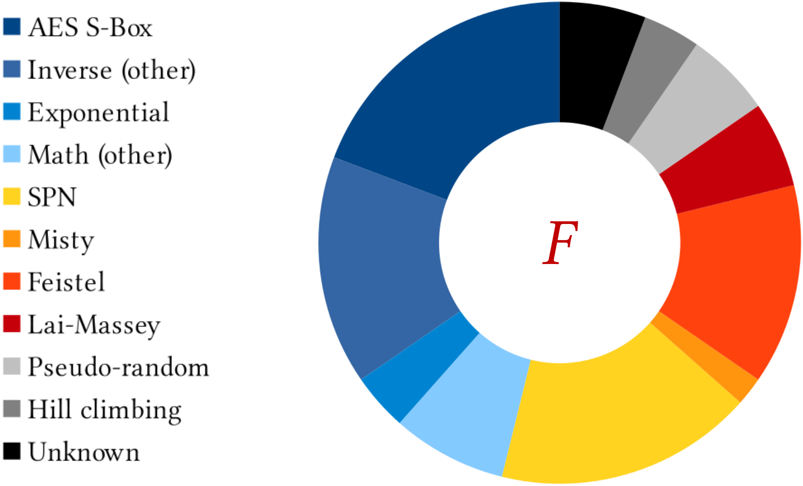
- F has not been picked uniformly at random.
- F has not been picked among a feasibly large set of random S-Boxes.
- Its linear properties were optimized (though poorly).

What Can We Deduce?

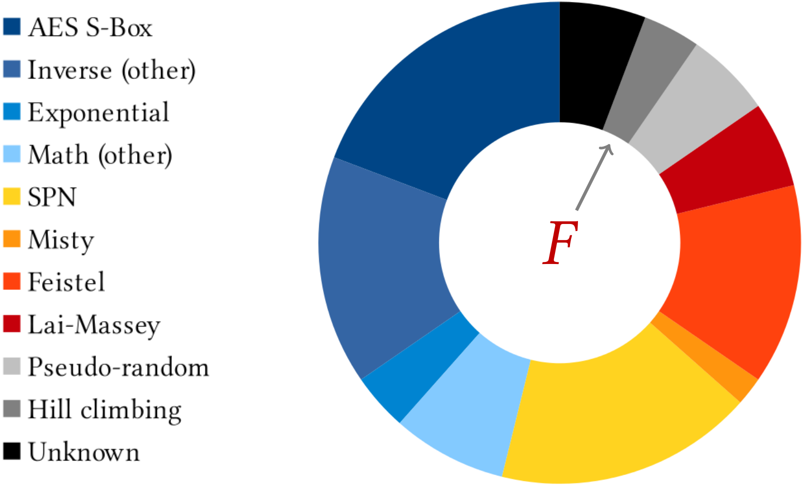
- F has not been picked uniformly at random.
- F has not been picked among a feasibly large set of random S-Boxes.
- Its linear properties were optimized (though poorly).

The S-Box of Skipjack was built using a dedicated algorithm.

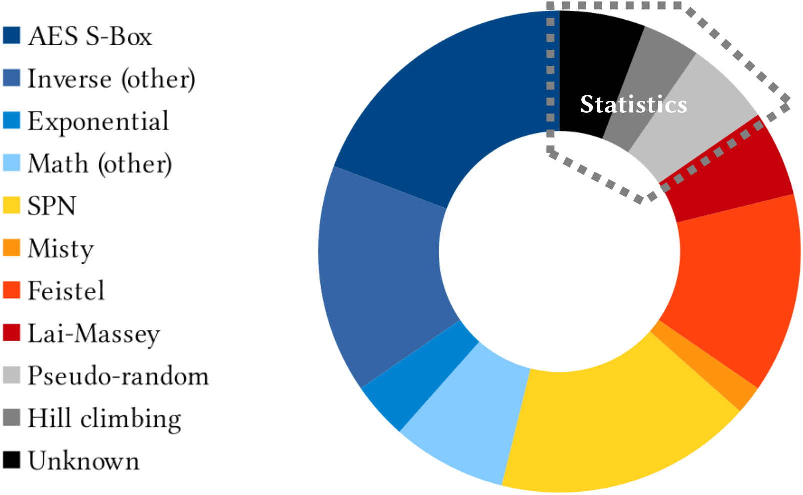
Conclusion on Skipjack



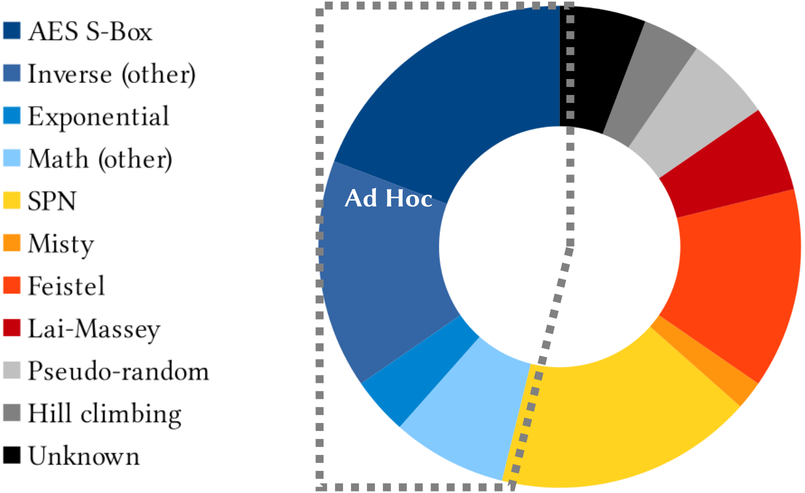
Conclusion on Skipjack



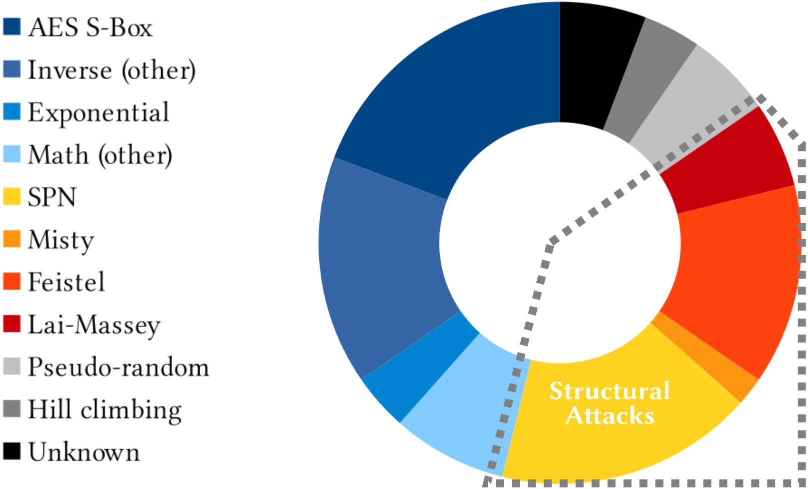
Different Techniques



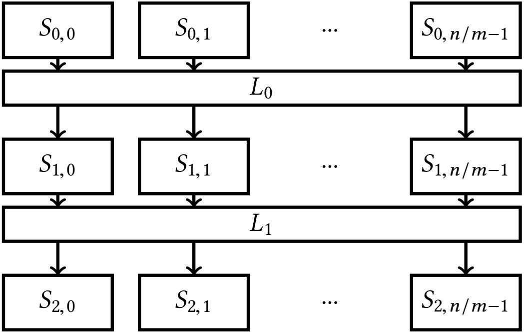
Different Techniques



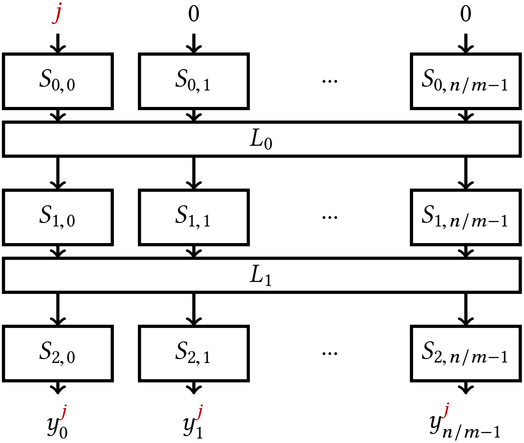
Different Techniques



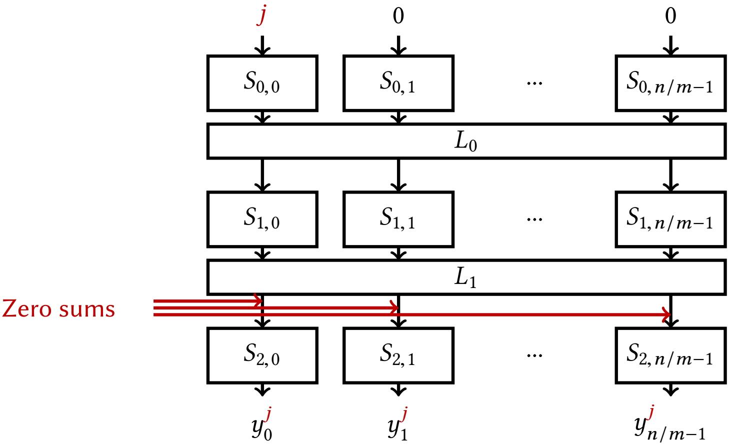
Attacks Against SPN (1/2)



Attacks Against SPN (1/2)

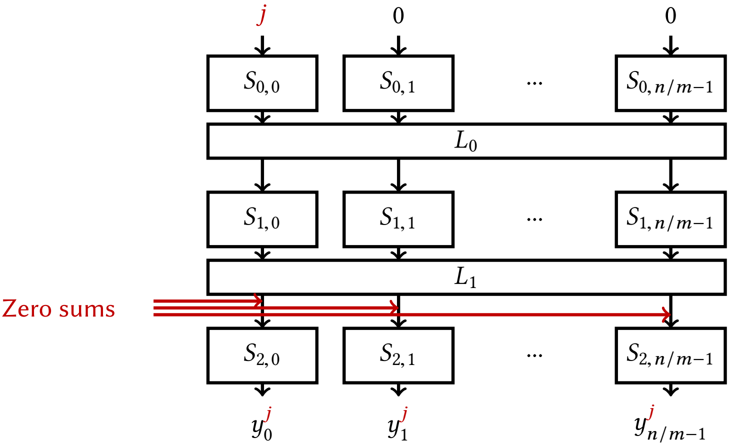


Attacks Against SPN (1/2)



$\bigoplus_{j=0}^{2^m-1} S_{2,i}(y_i^j) = 0$, for all i.

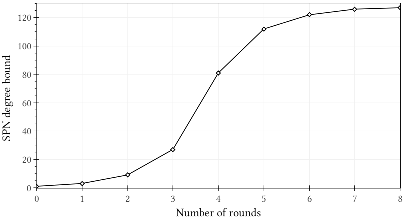
Attacks Against SPN (1/2)



$\bigoplus_{j=0}^{2^m-1} S_{2,i}(y_i^j) = 0$, for all i. Repeat for different constant then solve system [Biryukov, Shamir, 2001]

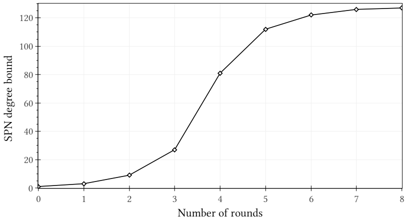
Attacks Against SPN (2/2)

Works against more than 3 rounds if $\deg(S(AS)^{r-1})$ is low enough.



Attacks Against SPN (2/2)

Works against more than 3 rounds if $\deg(S(AS)^{r-1})$ is low enough.



Degree Bound (SPN) [Biryukov et al., 2017]

Let σ operate on m bits, $\deg(\sigma) = m - 1$, and n be the block size.
Rhoughly speaking, $\deg\left(S(AS)^{r-1}\right) < n - 1$ as long as

$$(m-1)^{\lfloor r/2 \rfloor} < n \, .$$

Attacks Against Feistel Networks

Degree Bound (Feistel Network) [Perrin and Udovenko, 2016]

Let $\{F_i\}_{i<r}$ be permutations of $\mathbb{F}_2^{n/2}$ of degree d and let $\mathcal{F}^r(F)$ denote the r-round n-bit Feistel Network with round function F_i. If

$$d^{\lfloor r/2 \rfloor - 1} + d^{\lceil r/2 \rceil - 1} < n \, ,$$

then some degree $n-1$ terms in the ANF of $\mathcal{F}^r(F)$ are missing.

What Does it Take to Have Full Degree?

The degree based distinguishers for SPNs and Feistel networks can be seen as particular cases of this lemma.

Lemma

Let $F : \mathbb{F}_2^n \rightarrow \mathbb{F}_2$ be a Boolean function and let $G : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$ be a permutation. Then:

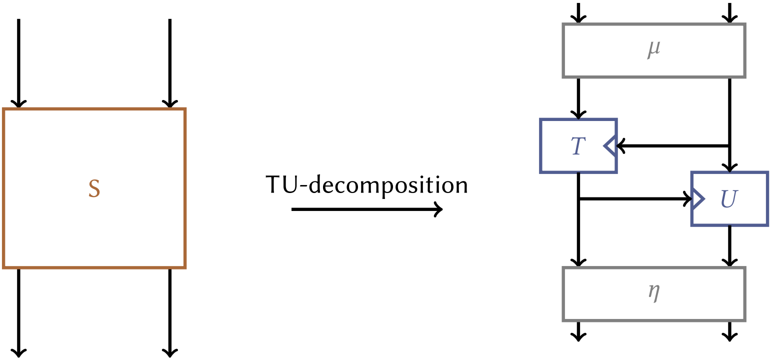
$$\deg(F \circ G) = n - 1 \implies \deg(F) + \deg(G^{-1}) \geq n \ .$$

Outline

Plan of this Section

What is the TU-Decomposition?

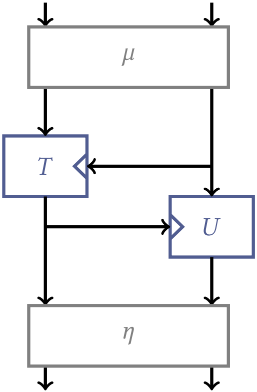
The *TU-decomposition* is a decomposition algorithm working against vast groups of algorithms: 3-round Feistel, Dillon's APN permutation, SAS, ...



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

TU-Decomposition in a Nutshell

Let $\mathcal{L}$ be the LAT of the target $S : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$.

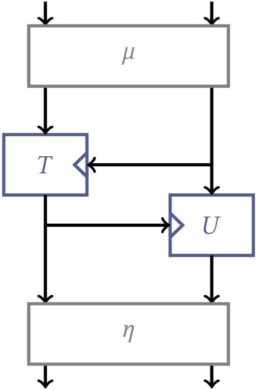


TU-Decomposition in a Nutshell

Let $\mathcal{L}$ be the LAT of the target $S : \mathbb{F}_2^n \to \mathbb{F}_2^n$.

1. Identify vector spaces $\mathcal{U}$ and $\mathcal{V}$ of dimension $n/2$ such that:

$$\mathcal{L}(a, b) = 0,\ \forall (a, b) \in \mathcal{U} \times \mathcal{V}\ .$$



TU-Decomposition in a Nutshell

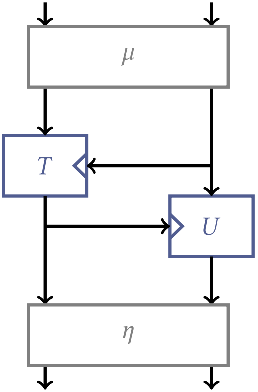
Let $\mathcal{L}$ be the LAT of the target $S : \mathbb{F}_2^n \to \mathbb{F}_2^n$.

1. Identify vector spaces $\mathcal{U}$ and $\mathcal{V}$ of dimension $n/2$ such that:

$$\mathcal{L}(a, b) = 0,\ \forall (a, b) \in \mathcal{U} \times \mathcal{V}\ .$$

2. Deduce linear permutations μ' and η' such that

$$\mathcal{L}(\mu'(a), \eta'(b)) = 0,\ \forall (a, b) \in \mathbb{F}_2^{n/2} \times \mathbb{F}_2^{n/2}$$



TU-Decomposition in a Nutshell

Let $\mathcal{L}$ be the LAT of the target $S : \mathbb{F}_2^n \to \mathbb{F}_2^n$.

1. Identify vector spaces $\mathcal{U}$ and $\mathcal{V}$ of dimension $n/2$ such that:

$$\mathcal{L}(a, b) = 0,\ \forall (a, b) \in \mathcal{U} \times \mathcal{V}\ .$$

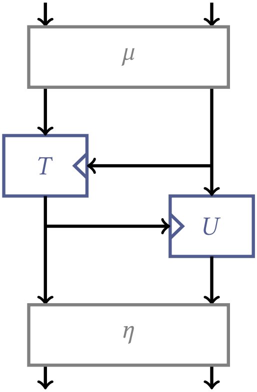
2. Deduce linear permutations μ' and η' such that

$$\mathcal{L}(\mu'(a), \eta'(b)) = 0,\ \forall (a, b) \in \mathbb{F}_2^{n/2} \times \mathbb{F}_2^{n/2}$$

3. Built new LAT $\mathcal{L}'$ such that

$$\mathcal{L}'(a, b) = \mathcal{L}(\mu'(a), \eta'(b))$$

and recover S' with LAT $\mathcal{L}'$. Deduce μ, η.



TU-Decomposition in a Nutshell

Let $\mathcal{L}$ be the LAT of the target $S : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$.

1. Identify vector spaces $\mathcal{U}$ and $\mathcal{V}$ of dimension $n/2$ such that:

$$\mathcal{L}(a,b) = 0,\ \forall (a,b) \in \mathcal{U} \times \mathcal{V}\ .$$

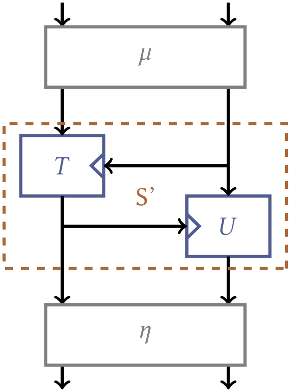
2. Deduce linear permutations μ' and η' such that

$$\mathcal{L}(\mu'(a), \eta'(b)) = 0,\ \forall (a,b) \in \mathbb{F}_2^{n/2} \times \mathbb{F}_2^{n/2}$$

3. Built new LAT $\mathcal{L}'$ such that

$$\mathcal{L}'(a,b) = \mathcal{L}(\mu'(a), \eta'(b))$$

and recover S' with LAT $\mathcal{L}'$. Deduce μ, η.



Bootstrapping TU-Decomposition

OK... But how do we find $\mathcal{U}$ and $\mathcal{V}$?

Bootstrapping TU-Decomposition

OK... But how do we find $\mathcal{U}$ and $\mathcal{V}$?

For now: we just look at the LAT and hope for the best!

Kuznyechik/Stribog

Stribog

Type Hash function

Publication [GOST, 2012]

Kuznyechik

Type Block cipher

Publication [GOST, 2015]

Kuznyechik/Stribog

Stribog

Type	Hash function
Publication	[GOST, 2012]

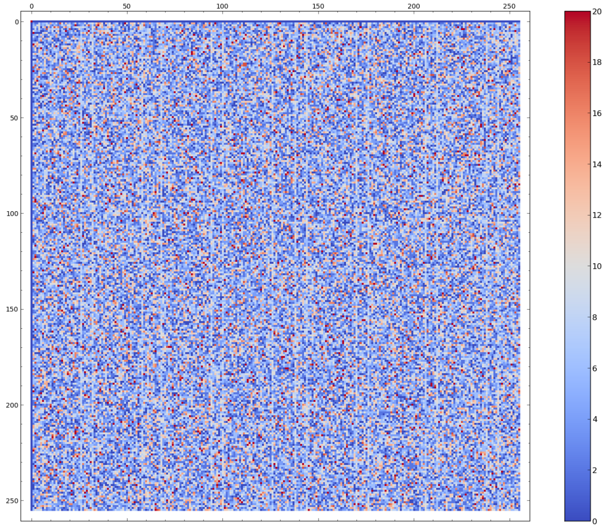
Kuznyechik

Type	Block cipher
Publication	[GOST, 2015]

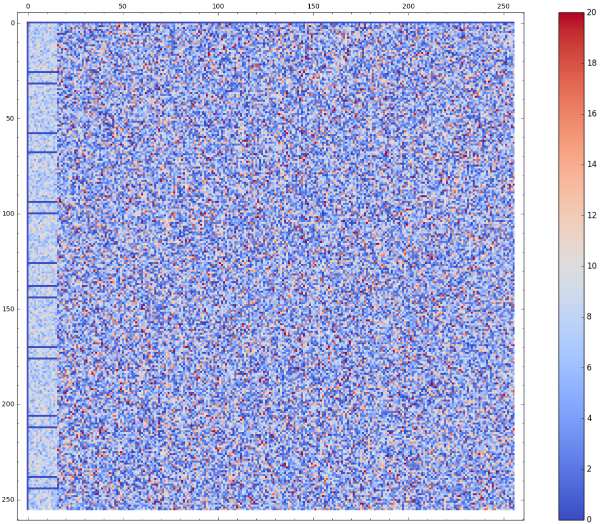
Common ground

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

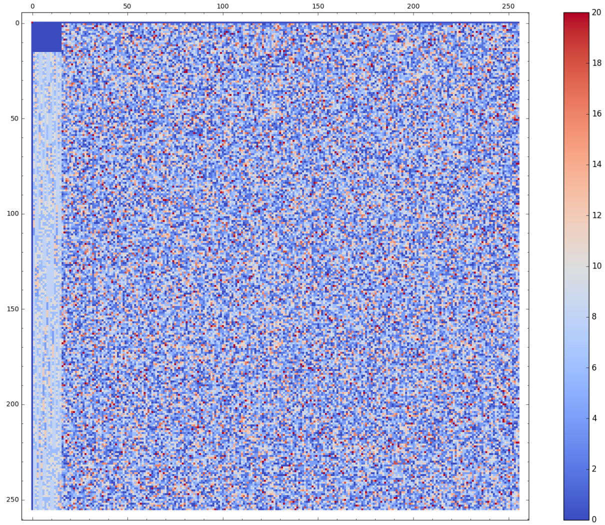
The LAT of the S-Box of Kuznyechik



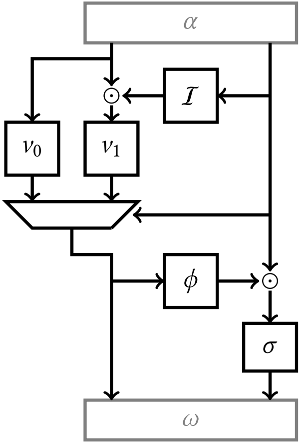
Applying one Linear Layer



Applying two Linear Layers



Final Decomposition Number 1



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

α Linear permutation

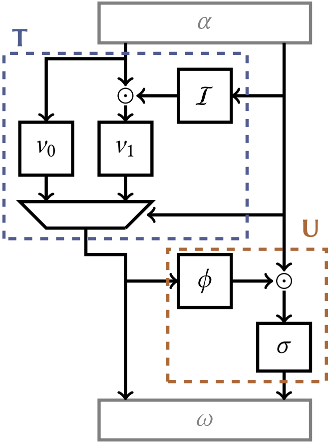
$\mathcal{I}$ Inversion in $\mathbb{F}_{2^4}$

ν_0, ν_1, σ 4×4 permutations

ϕ 4×4 function

ω Linear permutation

Final Decomposition Number 1



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

Conclusion for Kuznyechik/Stribog?

The Russian S-Box was built like a strange Feistel...

Conclusion for Kuznyechik/Stribog?

The Russian S-Box was built like a strange Feistel...

... or was it?

Conclusion for Kuznyechik/Stribog?

The Russian S-Box was built like a strange Feistel...

... or was it?

Belarussian inspiration

- The last standard of Belarus [Bel. St. Univ., 2011] uses an 8-bit S-box,
- somewhat similar to π...

Conclusion for Kuznyechik/Stribog?

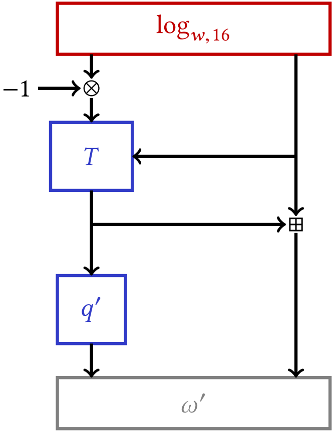
The Russian S-Box was built like a strange Feistel...

... or was it?

Belarussian inspiration

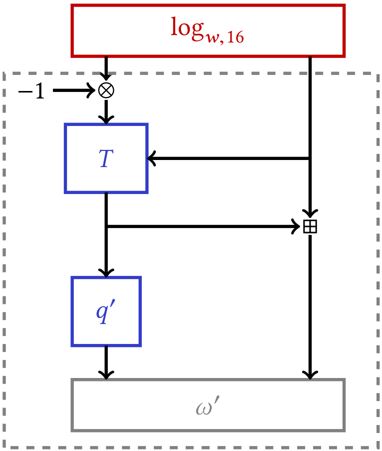
- The last standard of Belarus [Bel. St. Univ., 2011] uses an 8-bit S-box,
- somewhat similar to π...
- ... based on a finite field exponential!

Final Decomposition Number 2 (!)



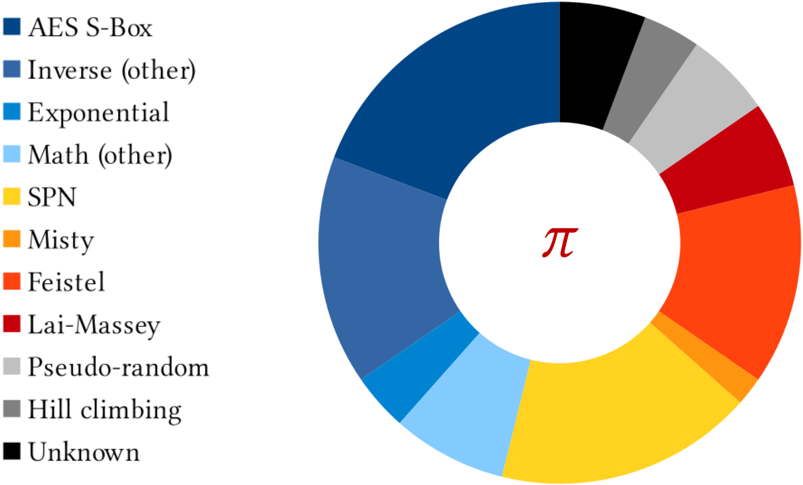
	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
T_0	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
T_1	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
T_2	0	1	2	3	4	5	6	7	8	9	a	b	c	d	f	e
T_3	0	1	2	3	4	5	6	7	8	9	a	b	c	f	d	e
T_4	0	1	2	3	4	5	6	7	8	9	a	b	f	c	d	e
T_5	0	1	2	3	4	5	6	7	8	9	a	f	b	c	d	e
T_6	0	1	2	3	4	5	6	7	8	9	f	a	b	c	d	e
T_7	0	1	2	3	4	5	6	7	8	f	9	a	b	c	d	e
T_8	0	1	2	3	4	5	6	7	f	8	9	a	b	c	d	e
T_9	0	1	2	3	4	5	6	f	7	8	9	a	b	c	d	e
T_a	0	1	2	3	4	5	f	6	7	8	9	a	b	c	d	e
T_b	0	1	2	3	4	f	5	6	7	8	9	a	b	c	d	e
T_c	0	1	2	3	f	4	5	6	7	8	9	a	b	c	d	e
T_d	0	1	2	f	3	4	5	6	7	8	9	a	b	c	d	e
T_e	0	1	f	2	3	4	5	6	7	8	9	a	b	c	d	e
T_f	0	f	1	2	3	4	5	6	7	8	9	a	b	c	d	e

Final Decomposition Number 2 (!)

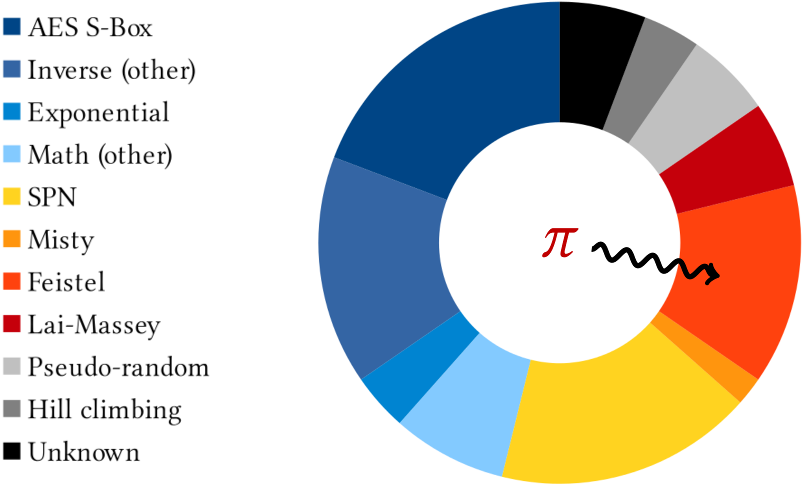


	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
T_0	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
T_1	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
T_2	0	1	2	3	4	5	6	7	8	9	a	b	c	d	f	e
T_3	0	1	2	3	4	5	6	7	8	9	a	b	c	f	d	e
T_4	0	1	2	3	4	5	6	7	8	9	a	b	f	c	d	e
T_5	0	1	2	3	4	5	6	7	8	9	a	f	b	c	d	e
T_6	0	1	2	3	4	5	6	7	8	9	f	a	b	c	d	e
T_7	0	1	2	3	4	5	6	7	8	f	9	a	b	c	d	e
T_8	0	1	2	3	4	5	6	7	f	8	9	a	b	c	d	e
T_9	0	1	2	3	4	5	6	f	7	8	9	a	b	c	d	e
T_a	0	1	2	3	4	5	f	6	7	8	9	a	b	c	d	e
T_b	0	1	2	3	4	f	5	6	7	8	9	a	b	c	d	e
T_c	0	1	2	3	f	4	5	6	7	8	9	a	b	c	d	e
T_d	0	1	2	f	3	4	5	6	7	8	9	a	b	c	d	e
T_e	0	1	f	2	3	4	5	6	7	8	9	a	b	c	d	e
T_f	0	f	1	2	3	4	5	6	7	8	9	a	b	c	d	e

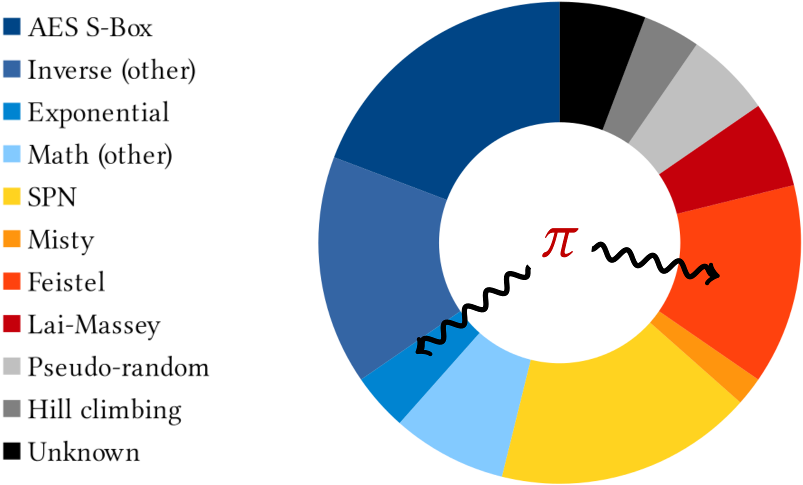
Conclusion on Kuznyechik/Stribog



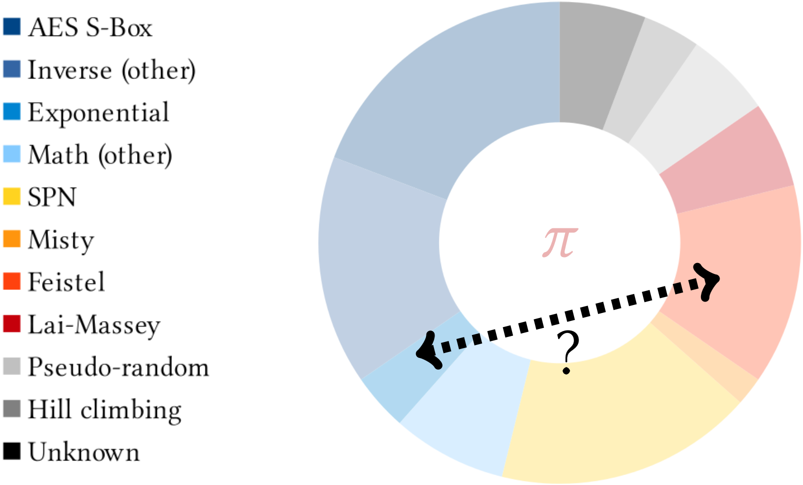
Conclusion on Kuznyechik/Stribog



Conclusion on Kuznyechik/Stribog



Conclusion on Kuznyechik/Stribog



Outline

Plan of this Section

The Big APN Problem

Definition (APN function)

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

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

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

The Big APN Problem

Definition (APN function)

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

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

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?

Dillon et al.'s Permutation

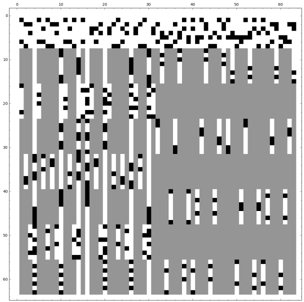
Only One Known Solution!

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

Dillon et al.'s Permutation

Only One Known Solution!

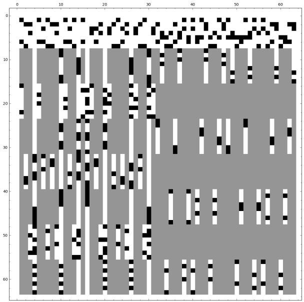
For $n = 6$, Dillon et al. found an APN permutation.

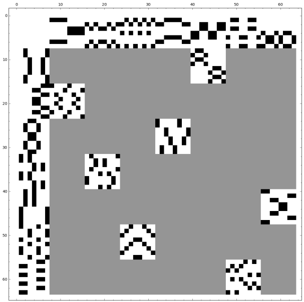


Dillon et al.'s Permutation

Only One Known Solution!

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

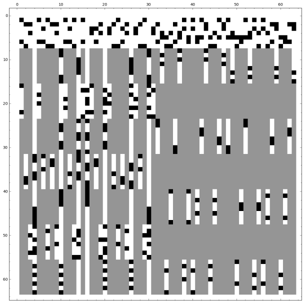


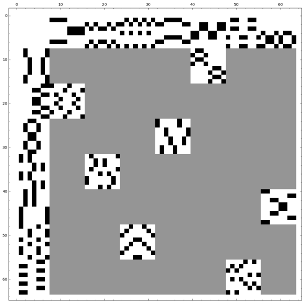


Dillon et al.'s Permutation

Only One Known Solution!

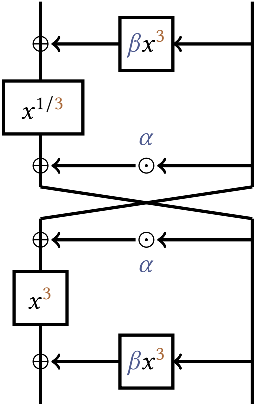
For $n = 6$, Dillon et al. found an APN permutation.





It is possible to make a TU-decomposition!

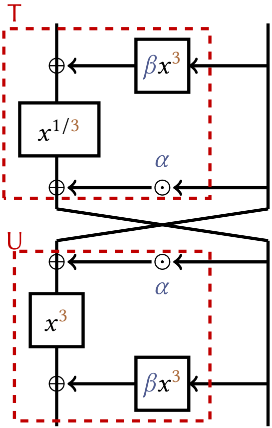
On the Butterfly Structure



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

This permutation is an open butterfly.

On the Butterfly Structure



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

This permutation is an open butterfly.

Lemma

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

CCZ-equivalence (1/2)

Definition (CCZ-equivalence)

Let F and G be functions of $\mathbb{F}_2^n$. They are CCZ-equivalent if there exists a linear permutation L of $\mathbb{F}_2^n \times \mathbb{F}_2^n$ such that

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

CCZ-equivalence (1/2)

Definition (CCZ-equivalence)

Let F and G be functions of $\mathbb{F}_2^n$. They are CCZ-equivalent if there exists a linear permutation L of $\mathbb{F}_2^n \times \mathbb{F}_2^n$ such that

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

Properties

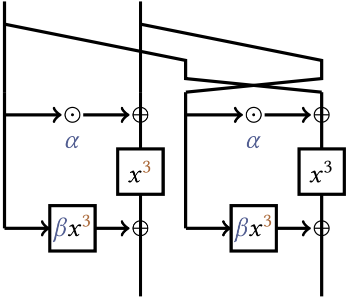
CCZ-equivalence preserves:

- the distribution of the coefficients in the LAT (Walsh spectrum),
- the distribution of the coefficients in the DDT.

It does not preserve:

- the position of the DDT/LAT coefficients
- the algebraic degree.

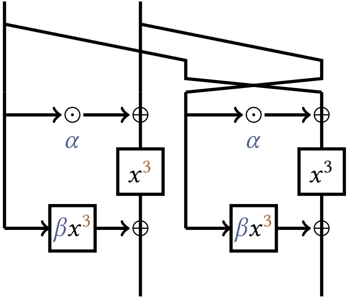
Closed Butterflies



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

This quadratic function is a closed butterfly.

Closed Butterflies



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

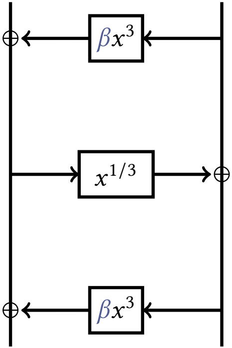
This quadratic function is a closed butterfly.

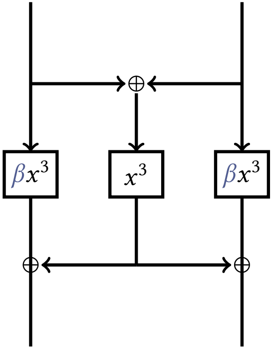
Lemma (Equivalence)

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

Butterflies and Feistel Networks

When $\alpha = 1$, butterflies can be greatly simplified.





Some Properties of Butterflies

Theorem (Properties of butterflies [Canteaut et al., 2017])

Let $\mathsf{V}^3_{\alpha,\beta}$ *and* $\mathsf{H}^3_{\alpha,\beta}$ *be butterflies operating on* $2n$ *bits,* n *odd. Then:*

- $\deg\left(\mathsf{V}^3_{\alpha,\beta}\right) = 2$,
- *if* $n = 3$, $\mathrm{Tr}(\alpha) = 0$ *and* $\beta + \alpha^3 \in \{\alpha, 1/\alpha\}$, *then*
$$\max(DDT) = \mathbf{2},\ \max(\mathcal{W}) = 2^{n+1} \ \textit{and}\ \deg\left(\mathsf{H}^3_{\alpha,\beta}\right) = n+1\,,$$
- *if* $\beta = (1+\alpha)^3$, *then*
$$\max(DDT) = 2^{n+1},\ \max(\mathcal{W}) = 2^{(3n+1)/2} \ \textit{and}\ \deg\left(\mathsf{H}^3_{\alpha,\beta}\right) = n\,,$$
- *otherwise,*
$$\max(DDT) = 4,\ \max(\mathcal{W}) = 2^{n+1} \ \textit{and}\ \deg\left(\mathsf{H}^3_{\alpha,\beta}\right) \in \{n, n+1\}$$
and $\deg\left(\mathsf{H}^3_{\alpha,\beta}\right) = n$ *if and only if*
$$1 + \alpha\beta + \alpha^4 = (\beta + \alpha + \alpha^3)^2\,.$$

Outline

Plan of this Section

Conclusion

- We can recover the majority of known S-Box structures and derive new results about Skipjack and Kuznyechik.

Conclusion

- We can recover the majority of known S-Box structures and derive new results about Skipjack and Kuznyechik.
- We can generalize the permutation of Dillon et al...

Conclusion

- We can recover the majority of known S-Box structures and derive new results about Skipjack and Kuznyechik.
- We can generalize the permutation of Dillon et al...
- but we can prove that our generalizations are never APN (except in the known case).

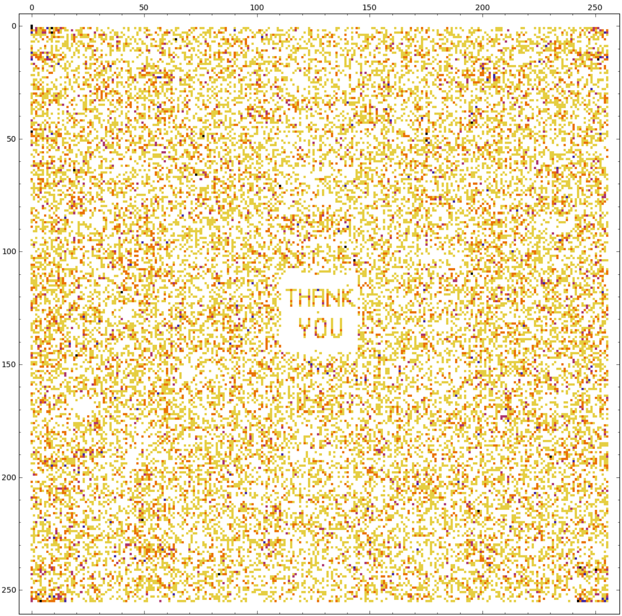
Conclusion

- We can recover the majority of known S-Box structures and derive new results about Skipjack and Kuznyechik.
- We can generalize the permutation of Dillon et al...
- but we can prove that our generalizations are never APN (except in the known case).
- There are still S-Boxes with unknown building strategies (CMEA, CSS)!

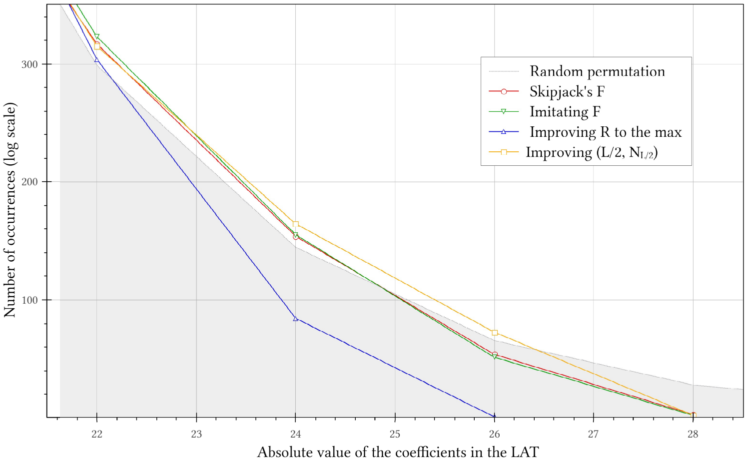
The Last S-Box

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

Conclusion



Details About Skipjack



Proof of Full Degree Condition

If $\deg(F \circ G) = n - 1$, then $\exists i \leq n$ such that $\bigoplus_{x \in C_i} (F \circ G)(x) = 1$.

Proof of Full Degree Condition

If $\deg(F \circ G) = n - 1$, then $\exists i \leq n$ such that $\bigoplus_{x \in C_i} (F \circ G)(x) = 1$.

Let $I_i : \mathbb{F}_2^n \to \mathbb{F}_2$ be such that $I_i(x) = 1 \Leftrightarrow x \in C_i$:

$$\bigoplus_{x \in C_i} (F \circ G)(x) \;=\; \bigoplus_{x \in \mathbb{F}_2^n} F\Big(G(x)\Big) \times I_i(x) \;,$$

Proof of Full Degree Condition

If $\deg(F \circ G) = n - 1$, then $\exists i \leq n$ such that $\bigoplus_{x \in C_i} (F \circ G)(x) = 1$.

Let $I_i : \mathbb{F}_2^n \to \mathbb{F}_2$ be such that $I_i(x) = 1 \Leftrightarrow x \in C_i$:

$$\bigoplus_{x \in C_i} (F \circ G)(x) \ = \ \bigoplus_{x \in \mathbb{F}_2^n} F\Big(G(x)\Big) \times I_i(x) \ ,$$

and let $y = G(x)$. Then:

$$\bigoplus_{x \in C_i} (F \circ G)(x) \ = \ \bigoplus_{y \in \mathbb{F}_2^n} F(y) \times I_i\Big(G^{-1}(y)\Big) \ .$$

Proof of Full Degree Condition

If $\deg(F \circ G) = n - 1$, then $\exists i \leq n$ such that $\bigoplus_{x \in C_i} (F \circ G)(x) = 1$.

Let $I_i : \mathbb{F}_2^n \to \mathbb{F}_2$ be such that $I_i(x) = 1 \Leftrightarrow x \in C_i$:

$$\bigoplus_{x \in C_i} (F \circ G)(x) = \bigoplus_{x \in \mathbb{F}_2^n} F\Big(G(x)\Big) \times I_i(x) ,$$

and let $y = G(x)$. Then:

$$\bigoplus_{x \in C_i} (F \circ G)(x) = \bigoplus_{y \in \mathbb{F}_2^n} F(y) \times I_i\Big(G^{-1}(y)\Big) .$$

This sum is equal to 1 if and only if $x \mapsto F(x) \times I_i\Big(G^{-1}(x)\Big)$ has degree n.

Proof of Full Degree Condition

If $\deg(F \circ G) = n - 1$, then $\exists i \leq n$ such that $\bigoplus_{x \in C_i} (F \circ G)(x) = 1$.

Let $I_i : \mathbb{F}_2^n \to \mathbb{F}_2$ be such that $I_i(x) = 1 \Leftrightarrow x \in C_i$:

$$\bigoplus_{x \in C_i} (F \circ G)(x) = \bigoplus_{x \in \mathbb{F}_2^n} F\big(G(x)\big) \times I_i(x)\ ,$$

and let $y = G(x)$. Then:

$$\bigoplus_{x \in C_i} (F \circ G)(x) = \bigoplus_{y \in \mathbb{F}_2^n} F(y) \times I_i\big(G^{-1}(y)\big)\ .$$

This sum is equal to 1 if and only if $x \mapsto F(x) \times I_i\big(G^{-1}(x)\big)$ has degree n.
I_i is affine ($I_i(x) = 1 + x_i$).

Proof of Full Degree Condition

If $\deg(F \circ G) = n - 1$, then $\exists i \leq n$ such that $\bigoplus_{x \in C_i} (F \circ G)(x) = 1$.

Let $I_i : \mathbb{F}_2^n \to \mathbb{F}_2$ be such that $I_i(x) = 1 \Leftrightarrow x \in C_i$:

$$\bigoplus_{x \in C_i} (F \circ G)(x) \;=\; \bigoplus_{x \in \mathbb{F}_2^n} F\Big(G(x)\Big) \times I_i(x)\,,$$

and let $y = G(x)$. Then:

$$\bigoplus_{x \in C_i} (F \circ G)(x) \;=\; \bigoplus_{y \in \mathbb{F}_2^n} F(y) \times I_i\Big(G^{-1}(y)\Big)\,.$$

This sum is equal to 1 if and only if $x \mapsto F(x) \times I_i\Big(G^{-1}(x)\Big)$ has degree n.
I_i is affine ($I_i(x) = 1 + x_i$). Thus, the sum can be equal to 1 only if

$$\deg(F) + \deg(G^{-1}) \geq n\,.$$

$\square$

Bibliography I

Bel. St. Univ. (2011).
"Information technologies. Data protection. Cryptographic algorithms for encryption and integrity control.".
State Standard of Republic of Belarus (STB 34.101.31-2011).
http://apmi.bsu.by/assets/files/std/belt-spec27.pdf.

Biryukov, A., Khovratovich, D., and Perrin, L. (2017).
Multiset-algebraic cryptanalysis of reduced Kuznyechik, Khazad, and secret SPNs.
IACR Transactions on Symmetric Cryptology, 2016(2):226–247.

Canteaut, A., Duval, S., and Perrin, L. (2017).
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, (to appear).

GOST (2012).
Gost r 34.11-2012: Streebog hash function.
https://www.streebog.net/.

Bibliography II

GOST (2015).
(GOST R 34.12–2015) information technology – cryptographic data security – block ciphers.
http://tc26.ru/en/standard/gost/GOST_R_34_12_2015_ENG.pdf.

Perrin, L. and Udovenko, A. (2016).
Algebraic insights into the secret feistel network.
In Peyrin, T., editor, *Fast Software Encryption – FSE 2016*, volume 9783 of *Lecture Notes in Computer Science*, pages 378–398. Springer, Heidelberg.