Reverse-Engineering the S-Box of Streebog, Kuznyechik and Stribob

Alex Biryukov, Léo Perrin, Aleksei Udovenko

SnT, University of Luxembourg

August 18, 2015

Our target

Definition (π)

π is the S-Box used by:

- **Streebog** (GOST latest hash function),
- **Kuznyechik** (GOST latest block cipher),
- **STRIBOBr1** (CAESAR candidate). Not used by STRIBOBr2.

Our target

Definition (π)

π is the S-Box used by:

- **Streebog** (GOST latest hash function),
- **Kuznyechik** (GOST latest block cipher),
- **STRIBOBr1** (CAESAR candidate). Not used by STRIBOBr2.

Properties

- Permutation of 8 bits

Our target

Definition (π)

π is the S-Box used by:

- **Streebog** (GOST latest hash function),
- **Kuznyechik** (GOST latest block cipher),
- **STRIBOBr1** (CAESAR candidate). Not used by STRIBOBr2.

Properties

- Permutation of 8 bits
- $\mathbf{P}[\text{differential properties}] \leq 2^{-82.7}$

Our target

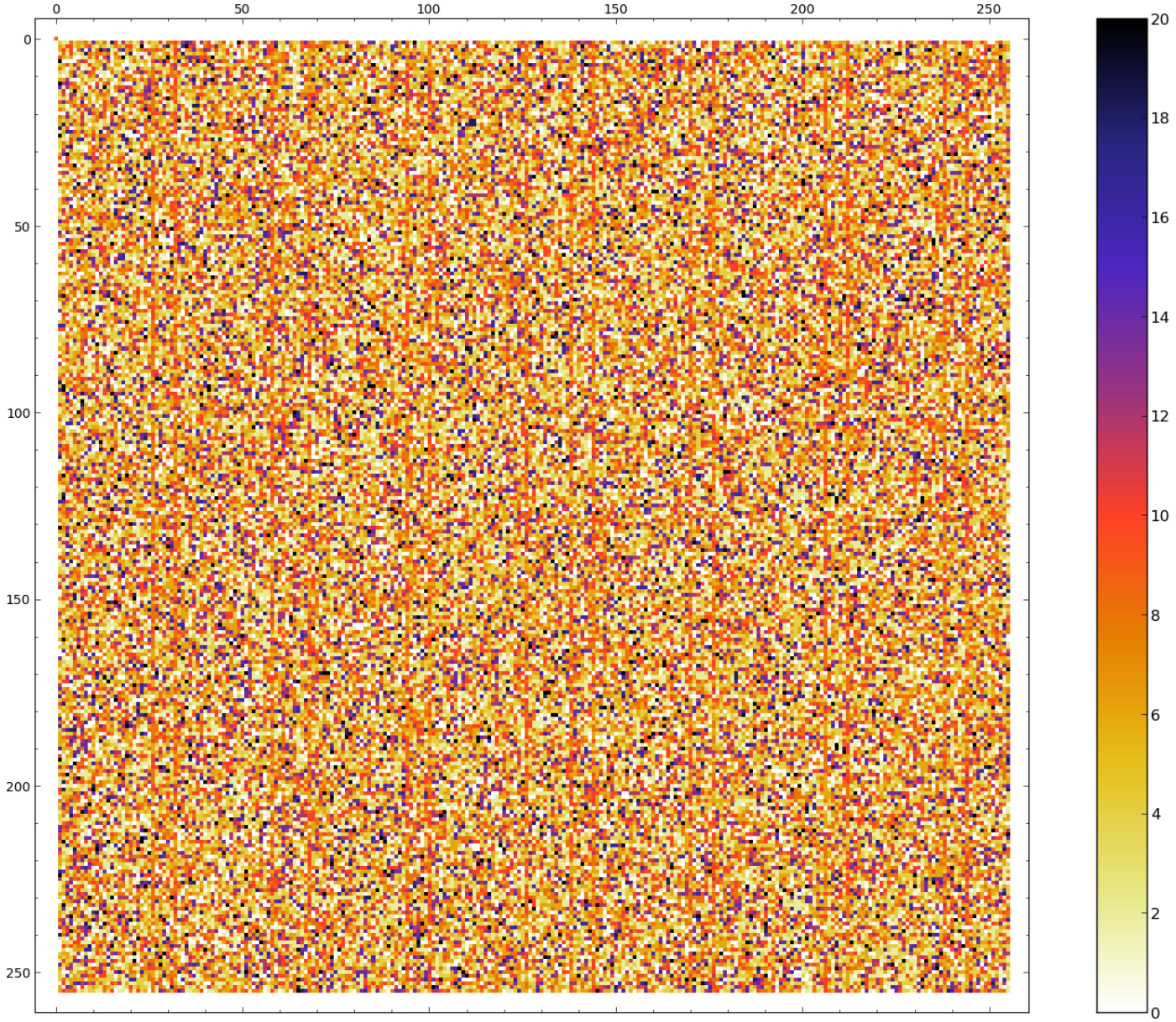
Definition (π)

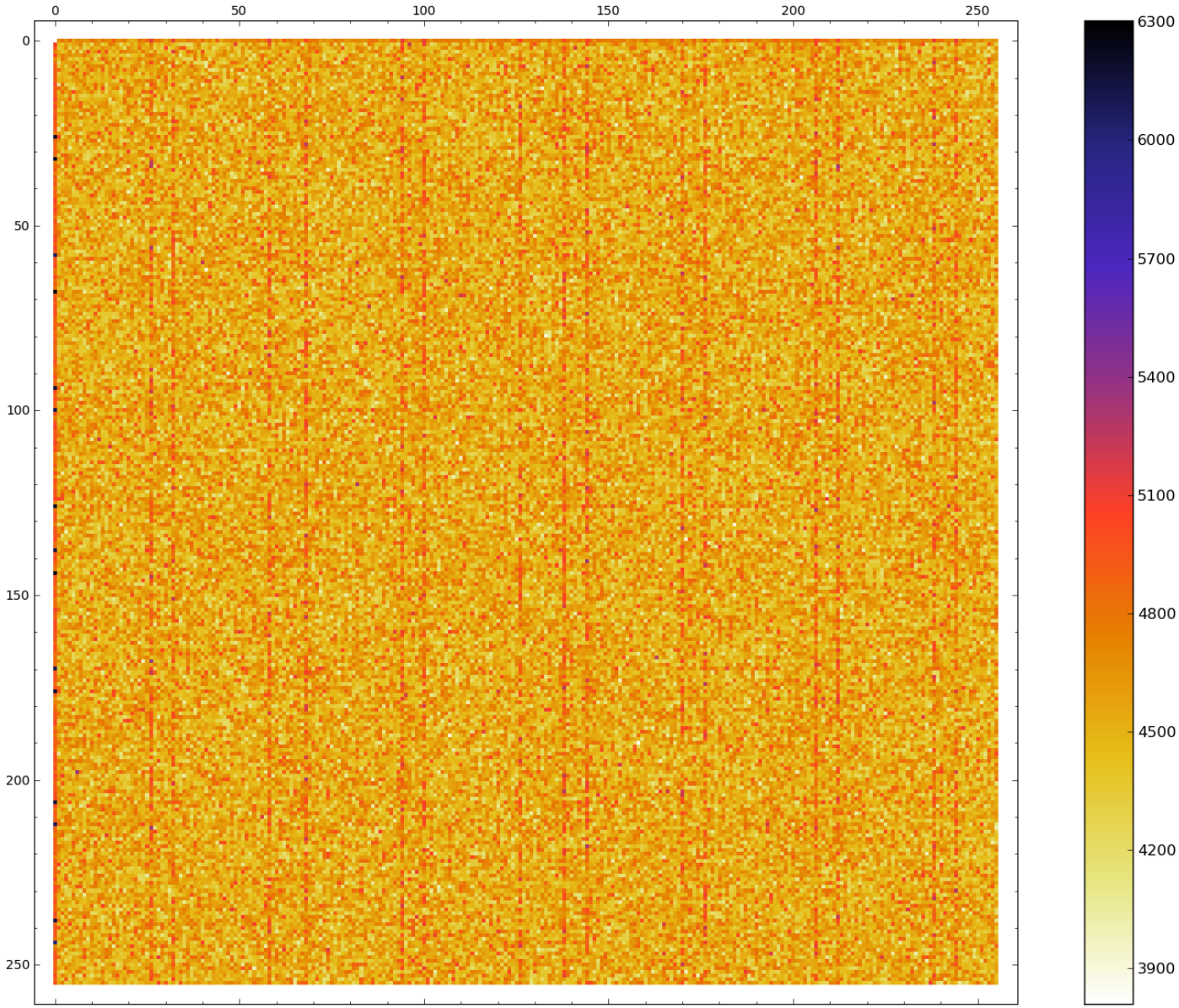
π is the S-Box used by:

- **Streebog** (GOST latest hash function),
- **Kuznyechik** (GOST latest block cipher),
- **STRIBOBr1** (CAESAR candidate). Not used by STRIBOBr2.

Properties

- Permutation of 8 bits
- $\mathbf{P}[\text{differential properties}] \leq 2^{-82.7}$
- ... No explanation whatsoever.


0
50
100
150
200
250
0
50
100
150
200
250
0
2
4
6
8
10
12
14
16
18
20


0
50
100
150
200
250
6300
6000
5700
5400
5100
4800
4500
4200
3900

Reverse-Engineering the S-Box

Pull on the thread!

Reverse-Engineering the S-Box

Pull on the thread!

1. Lines/dot

Reverse-Engineering the S-Box

Pull on the thread!

1. Lines/dot
2. A vector space

Reverse-Engineering the S-Box

Pull on the thread!

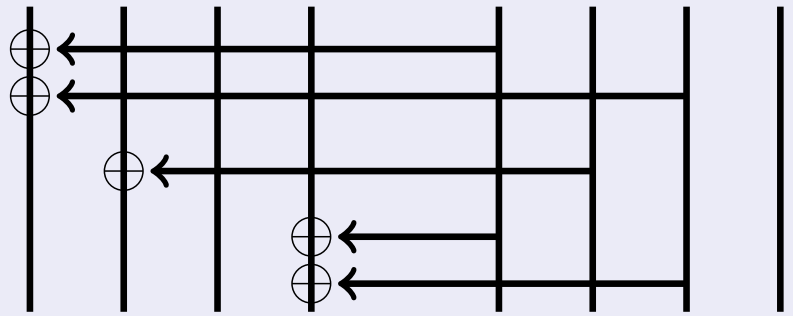
1. Lines/dot
2. A vector space
3. Partial linear layers

Reverse-Engineering the S-Box

Pull on the thread!

1. Lines/dot
2. A vector space
3. Partial linear layers

Linear layer L^*

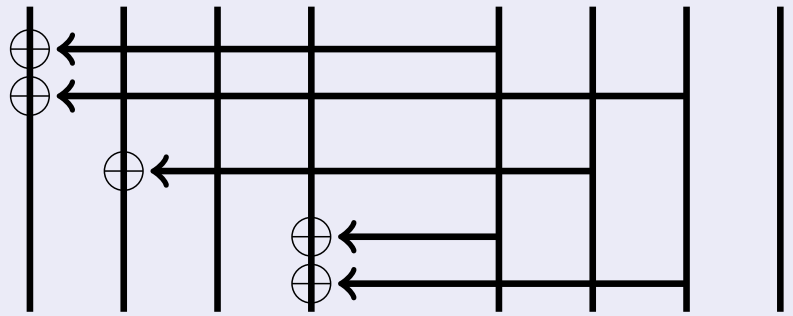


Reverse-Engineering the S-Box

Pull on the thread!

1. Lines/dot
2. A vector space
3. Partial linear layers
4. Integral properties

Linear layer L^*

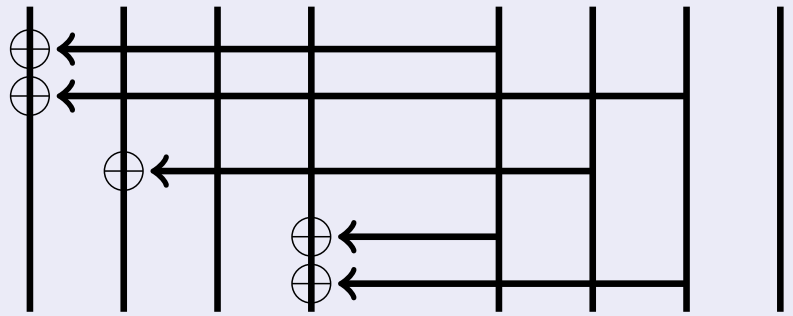


Reverse-Engineering the S-Box

Pull on the thread!

1. Lines/dot
2. A vector space
3. Partial linear layers
4. Integral properties
5. High level structure

Linear layer L^*

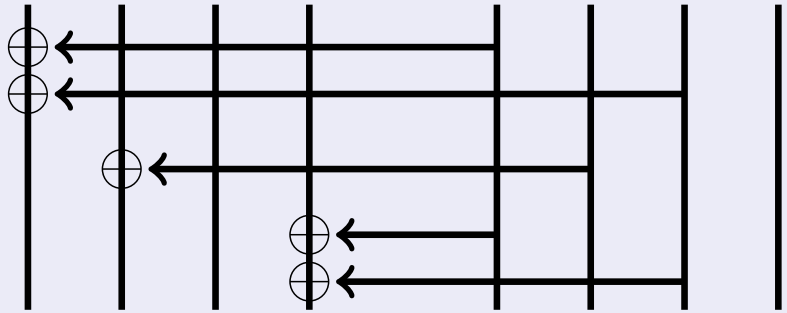


Reverse-Engineering the S-Box

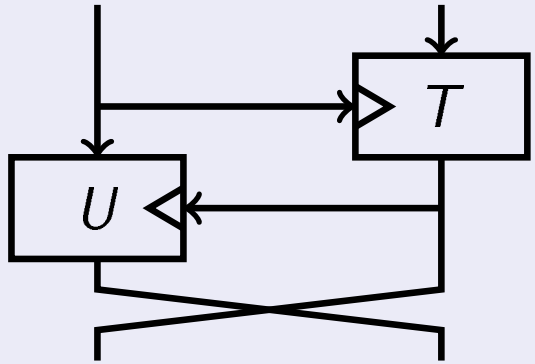
Pull on the thread!

1. Lines/dot
2. A vector space
3. Partial linear layers
4. Integral properties
5. High level structure

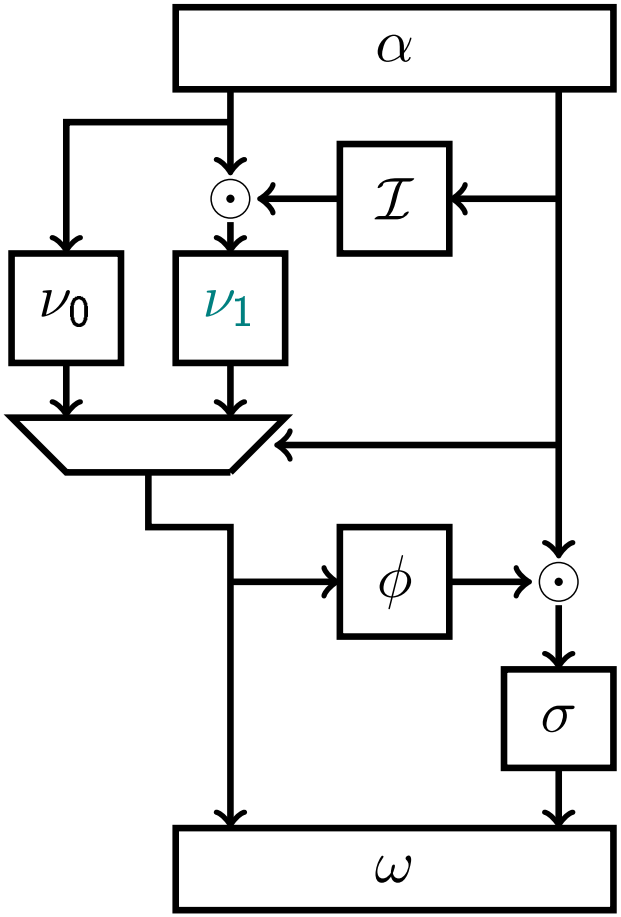
Linear layer L^*



Structure of $L^* \circ \pi^{-1} \circ L^*$

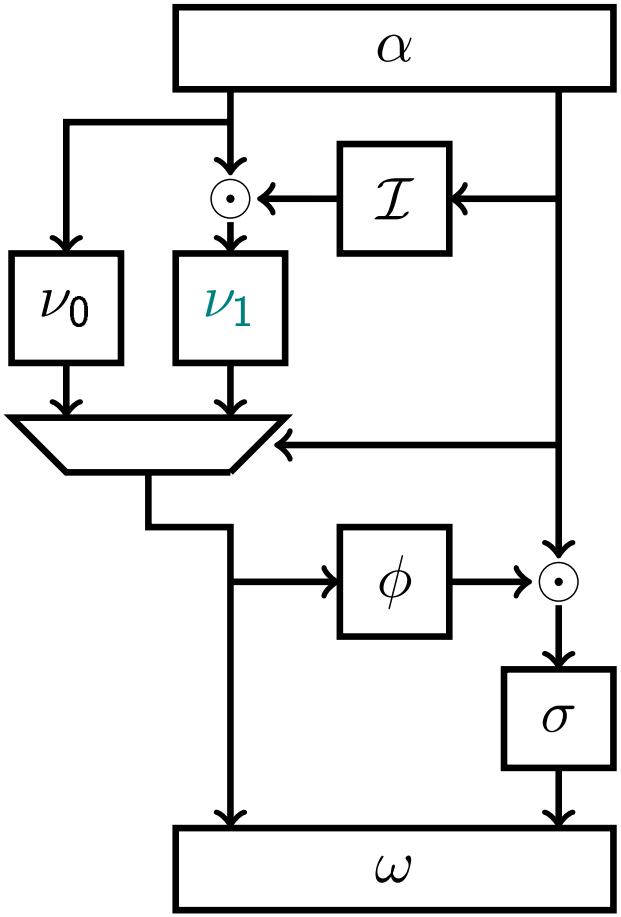


Final Decomposition



- $\odot$ 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



- $\odot$ 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

$$P[\nu_1(x \oplus \texttt{0x9}) \oplus \nu_1(x) = \texttt{0x2}] = \mathbf{1}$$

Conclusion

Hardware Implementation

Structure	Area (μm^2)	Delay (ns)
naive implementation	3889.6	362.52
using the decomposition	1530.1	46.11

Conclusion

Hardware Implementation

Structure	Area (μm^2)	Delay (ns)
naive implementation	3889.6	362.52
using the decomposition	1530.1	46.11

https://eprint.iacr.org/2015/812.pdf

Thank you!