An algorithm for bounding non-minimum weight differentials in 2-round LSX-ciphers

Vitaly Kiryukhin

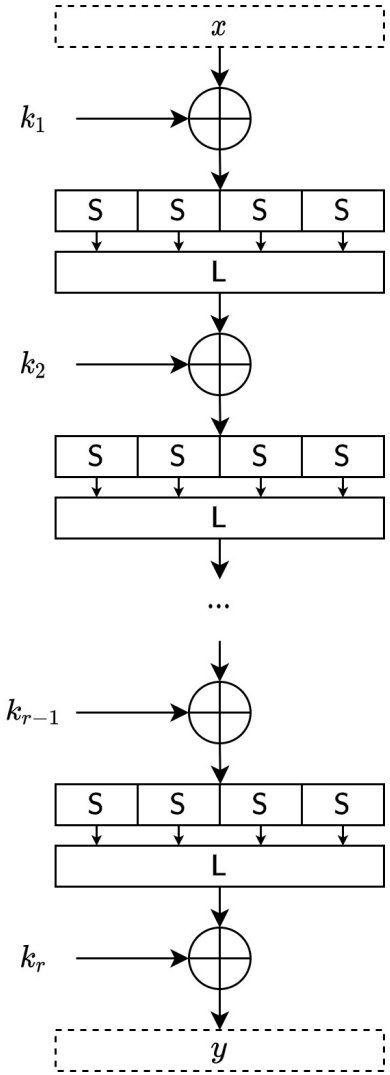
JSC «InfoTeCS», LLC «SFB Lab»

CTCrypt 2020

September 16, 2020

vitaly.kiryukhin@infotecs.ru

LSX-cipher



- R similar rounds
- XOR with round key
- nonlinear layer: n bijective parallel m-bit Sboxes
- linear layer

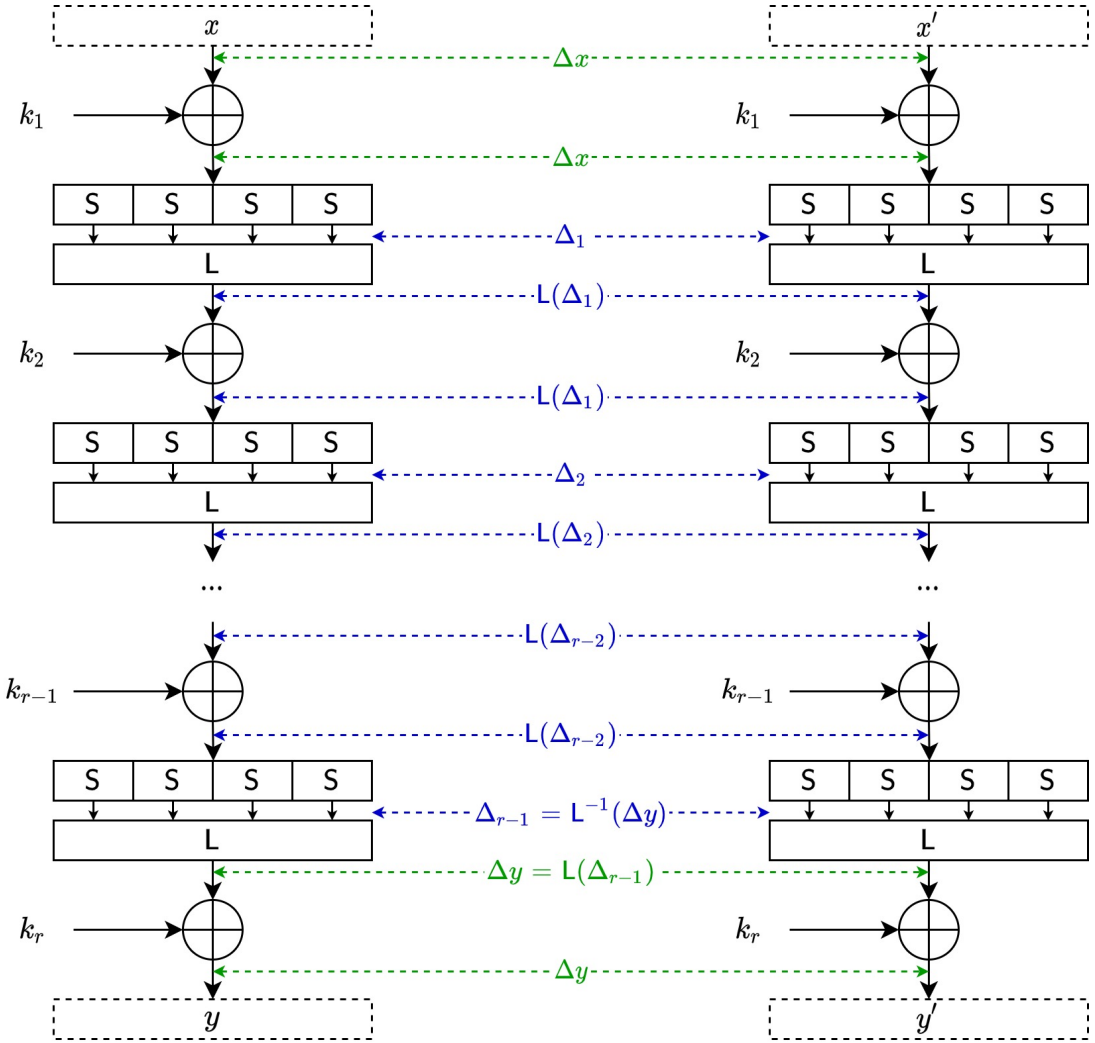
AES, Kuznyechik, Khazad and many others.

Markov assumption

We assume that all round keys are **independent** and uniformly distributed

[Lai X., Massey J.L., Murphy S. – *Markov ciphers and differential cryptanalysis* – 1991]

Difference propagation



Change of the difference $\Delta x = x \oplus x'$:

- XOR — not change
- S — non-bijective
- L — bijective

Differentials and trails

Differential trail (characteristic) = sequence of differences:

$$\Omega = \Delta x \to \Delta_1 \to \Delta_2 \to \ldots \to \Delta y$$

Differentials and trails

Differential trail (characteristic) = sequence of differences:

$$\Omega = \Delta x \to \Delta_1 \to \Delta_2 \to \ldots \to \Delta y$$

Differential = set of differential trails:

$$(\Delta x, \Delta y) = \{\Omega_i : \Delta x \to \Delta_1^{(i)} \to \Delta_2^{(i)} \to \ldots \to \Delta y\}$$

Security of the cipher

Informally:

The cipher is secure against differential cryptanalysis $\iff$

$\nexists(\Delta x, \Delta y) \neq 0$: conditional probability $\Pr(\Delta x \to \Delta y) \gg 2^{-N}$.

Security of the cipher

Informally:

The cipher is secure against differential cryptanalysis $\iff$

$\nexists(\Delta x, \Delta y) \neq 0$: conditional probability $\Pr(\Delta x \to \Delta y) \gg 2^{-N}$.

- What should we take for this probability?

Security of the cipher

Informally:

The cipher is secure against differential cryptanalysis $\iff$

$\nexists(\Delta x, \Delta y) \neq 0$: conditional probability $\Pr(\Delta x \to \Delta y) \gg 2^{-N}$.

- What should we take for this probability?
- How to compute it?

Security of the cipher

Informally:

The cipher is secure against differential cryptanalysis $\iff$

$\nexists(\Delta x, \Delta y) \neq 0$: conditional probability $\Pr(\Delta x \to \Delta y) \gg 2^{-N}$.

- What should we take for this probability?
- How to compute it?
- How to estimate it?

DP and EDP

The differential probability

$$\mathrm{DP}\left(\Delta x, \Delta y\right) = \Pr\left(f(x) \oplus f(x \oplus \Delta x) = \Delta y\right)$$

DP and EDP

The differential probability

$$\mathrm{DP}\left(\Delta x, \Delta y\right) = \Pr\left(f(x) \oplus f(x \oplus \Delta x) = \Delta y\right)$$

The **expected** probability of the trail Ω

$$\mathrm{EDCP}\left(\Omega\right) = 2^{-\kappa} \sum_{K \in \mathbf{F}_2^{\kappa}} \Pr\left(\Delta_1{=}x_1 \oplus x_1' \text{ and } \Delta_2{=}x_2 \oplus x_2' \text{ ... and } \Delta y{=}y \oplus y'\right)$$

DP and EDP

The differential probability

$$\mathrm{DP}\left(\Delta x, \Delta y\right) = \Pr\left(f(x) \oplus f(x \oplus \Delta x) = \Delta y\right)$$

The **expected** probability of the trail Ω

$$\mathrm{EDCP}\left(\Omega\right) = 2^{-\kappa} \sum_{K \in \mathbf{F}_2^\kappa} \Pr\left(\Delta_1{=}x_1 \oplus x_1' \text{ and } \Delta_2{=}x_2 \oplus x_2' \text{ ... and } \Delta y{=}y \oplus y'\right)$$

The **expected** probability of the differential $(\Delta x, \Delta y)$

$$\mathrm{EDP}\left(\Delta x, \Delta y\right) = 2^{-\kappa} \sum_{K \in \mathbf{F}_2^\kappa} \Pr\left(E_K(x) \oplus E_K(x \oplus \Delta x) = \Delta y\right)$$

Relations EDP and EDCP

$$\mathrm{EDP}\left(\Delta x, \Delta y\right) = \sum_{\Omega:\Delta x \to \ldots \to \Delta y} \mathrm{EDCP}\left(\Delta x \to \ldots \to \Delta y\right)$$

[Lai X., Massey J.L., Murphy S. – *Markov ciphers and differential cryptanalysis* – 1991]

Small example: toy cipher

$\mathsf{s} : \mathbf{F}_2^5 \to \mathbf{F}_2^5$

$\mathsf{s} = [20, 17, 31, 22, 29, 27, 13, 1, 21, 15, 4, 9, 11, 10, 7, 3, 14, 19, 5, 0, 12, 18, 23, 28, 16, 6, 25, 8, 30, 2, 26, 24]$

$$p = \max_{\Delta x, \Delta y \neq 0} \mathrm{DP}\left(\Delta x, \Delta y\right) = \max_{\Delta x, \Delta y \neq 0} \Pr\left(\mathsf{s}(x) \oplus \mathsf{s}(x \oplus \Delta x) = \Delta y\right) =$$

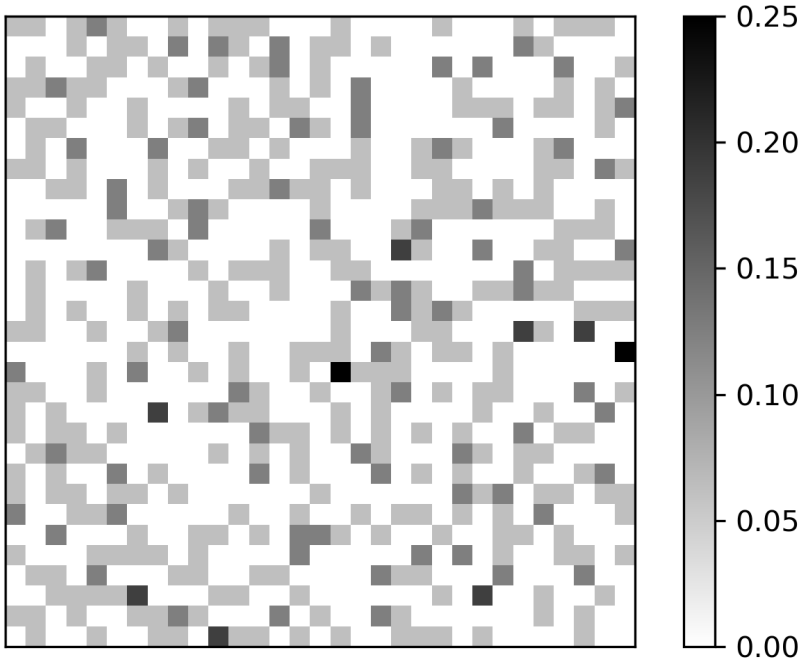
$$= \frac{8}{32} = 2^{-2}$$

Small example: toy cipher

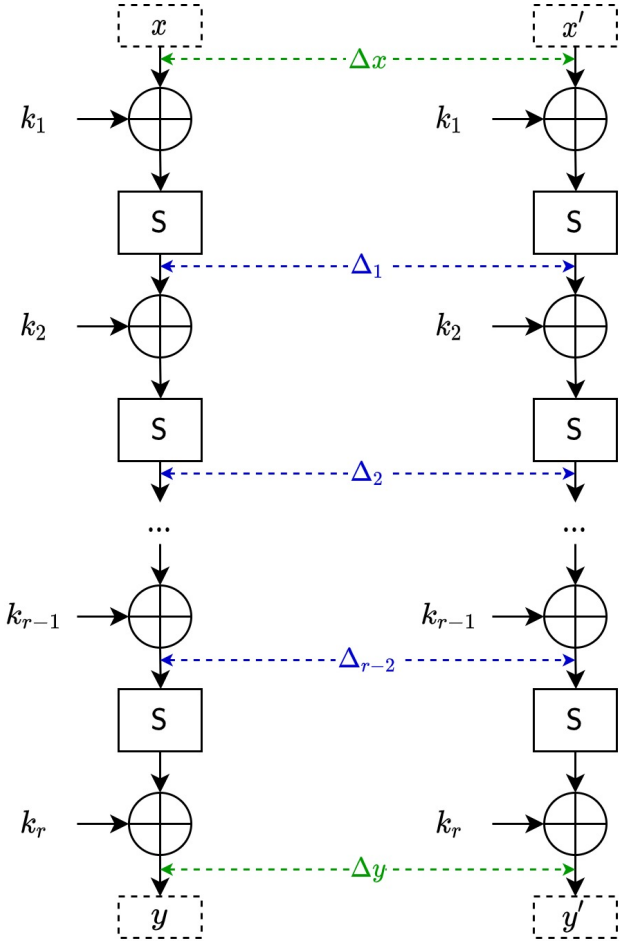
DDT

$\mathsf{s} = [20, 17, 31, 22, 29, 27, 13, 1, 21, 15, 4, 9, 11, 10, 7, 3, 14, 19, 5, 0, 12, 18, 23, 28, 16, 6, 25, 8, 30, 2, 26, 24]$

$$\mathrm{DDT}[\Delta x][\Delta y] = \frac{\#\{x : \mathsf{s}(x) \oplus \mathsf{s}(x \oplus \Delta x) = \Delta y\}}{2^5}$$



Small example: toy cipher



$$E_K(x) = y =$$

$$= \mathsf{s}(\ldots \mathsf{s}(\mathsf{s}(x \oplus k_1) \oplus k_2) \ldots \oplus k_{r-1}) \oplus k_r$$

Small example: toy cipher

Upper bound on differential characteristic

$$\max_{\Omega \neq 0} EDCP(\Delta x \to \ldots \to \Delta y) \leq p^R \leq p^\Theta$$

R – rounds;

Θ – minimal number of active Sboxes;

Rounds	1	2	3	4	
p^θ	2^{-2}	2^{-4}	2^{-6}	2^{-8}	decreasing

Small example: toy cipher

Differential characteristic

$$R = 1 : \Omega = \Delta x \to \Delta y \qquad = 12 \to 11$$

$$R = 2 : \Omega = \Delta x \to \Delta_1 \to \Delta y \qquad = 12 \to 11 \to 1\mathtt{f}$$

$$R = 3 : \Omega = \Delta x \to \Delta_1 \to \Delta_2 \to \Delta y \qquad = 12 \to 11 \to 1\mathtt{f} \to \mathtt{b}$$

$$R = 4 : \Omega = \Delta x \to \Delta_1 \to \Delta_2 \to \Delta_3 \to \Delta y = 16 \to 12 \to 11 \to 1\mathtt{f} \to \mathtt{b}$$

Rounds	1	2	3	4	
p^θ	2^{-2}	2^{-4}	2^{-6}	2^{-8}	decreasing
$\max EDCP(\Delta x \to \ldots \to \Delta y)$	2^{-2}	2^{-4}	$2^{-6.4}$	$2^{-9.4}$	decreasing, $\leq p^\theta$

Small example: toy cipher

Differential

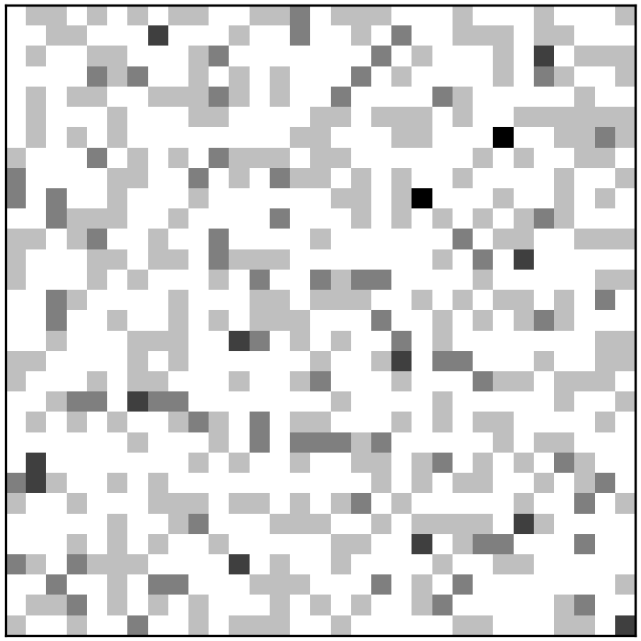
By definition: let's compute $\Pr\left(E_K(x) \oplus E_K(x \oplus \Delta x) = \Delta y\right)$ for each key

Small example: toy cipher

Differential

By definition: let's compute $\Pr\left(E_K(x) \oplus E_K(x \oplus \Delta x) = \Delta y\right)$ for each key

1)

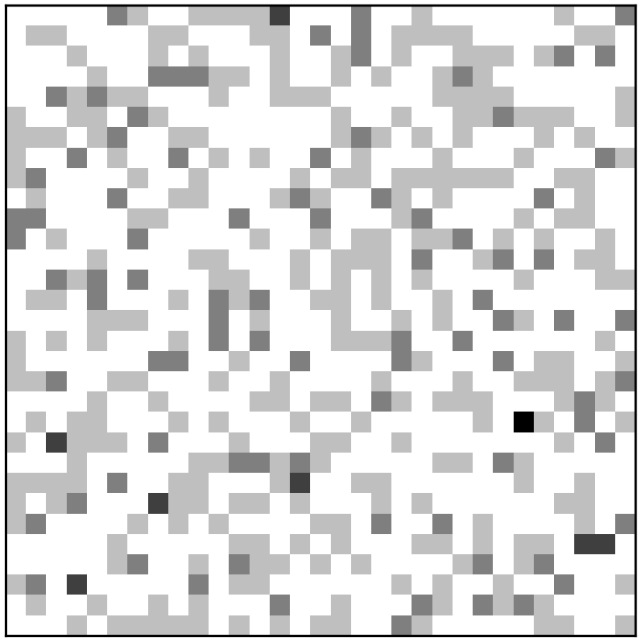


Small example: toy cipher

Differential

By definition: let's compute $\Pr\left(E_K(x) \oplus E_K(x \oplus \Delta x) = \Delta y\right)$ for each key

2)

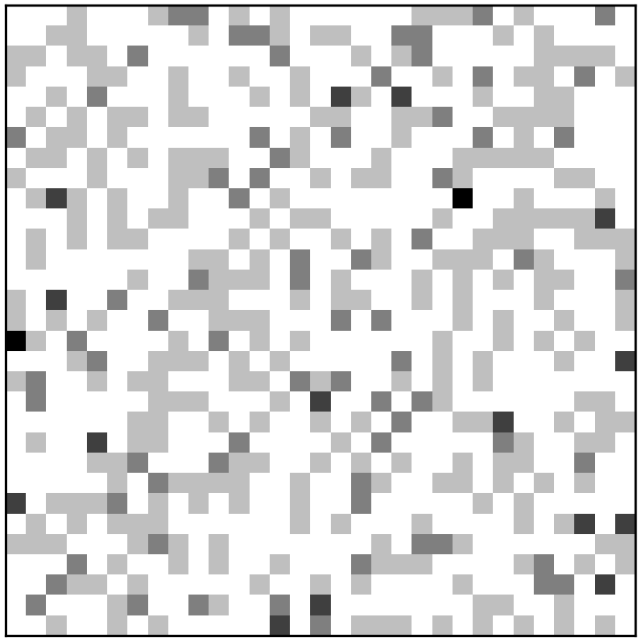


Small example: toy cipher

Differential

By definition: let's compute $\Pr\left(E_K(x) \oplus E_K(x \oplus \Delta x) = \Delta y\right)$ for each key

3)



Small example: toy cipher

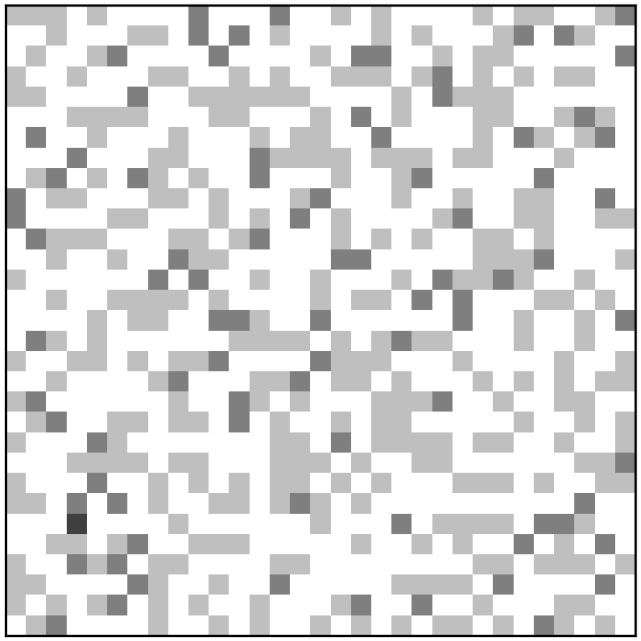
Differential

• • •

Small example: toy cipher

Differential

By definition: let's compute $\Pr\left(E_K(x) \oplus E_K(x \oplus \Delta x) = \Delta y\right)$ for each key. 2^κ)

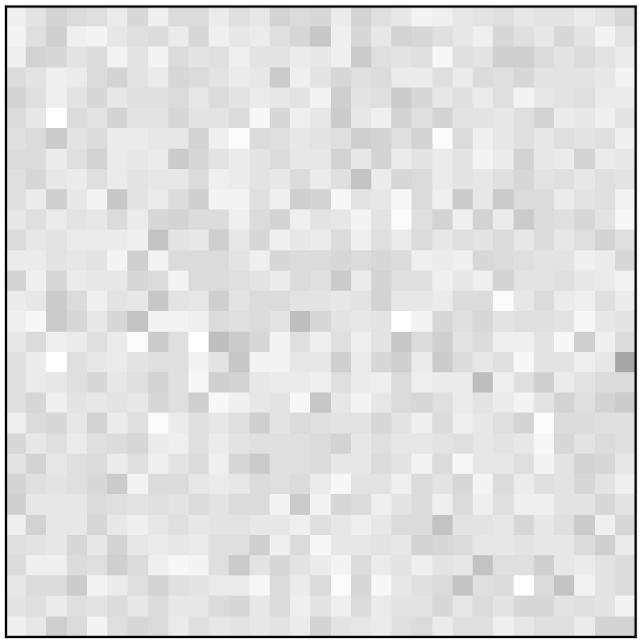


Small example: toy cipher

Differential

By definition: let's compute $\Pr\left(E_K(x) \oplus E_K(x \oplus \Delta x) = \Delta y\right)$ for each key

After averaging:



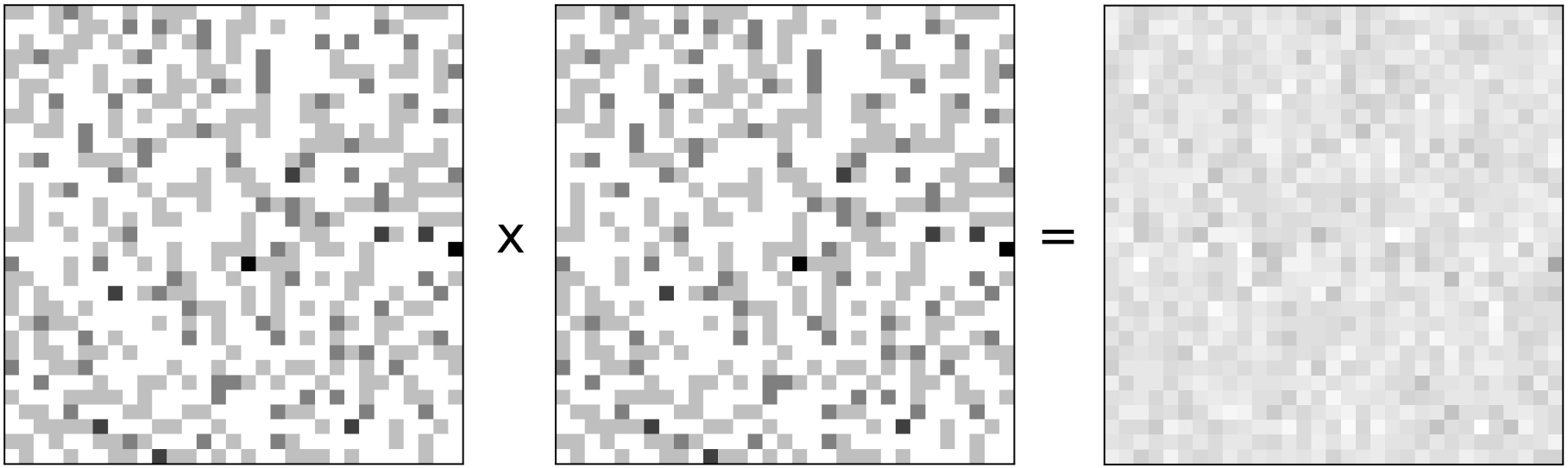
Small example: toy cipher

Differential

Another way:

we can consider $\mathrm{DDT_s}$ as a matrix of transition probabilities

$$EDP(\Delta x, \Delta y) = 2^{-\kappa} \sum_{K \in \mathbf{F}_2^{\kappa}} \Pr\left(E_K(x) \oplus E_K(x \oplus \Delta x) = \Delta y\right) = (\mathrm{DDT})^R [\Delta x][\Delta y]$$



Small example: toy cipher

Differential

$$R = 2\colon \max_{\Delta x, \Delta y \neq 0} \mathrm{EDP}\,(\Delta x, \Delta y) = 2^{-3.4\ldots},$$

$$\begin{aligned}(\Delta x, \Delta y) = (\mathtt{12}, \mathtt{1f}) = \{&\mathtt{12} \to \mathtt{5} \ \to \mathtt{1f},\\ &\mathtt{12} \to \mathtt{c} \ \to \mathtt{1f},\\ &\mathtt{12} \to \mathtt{f} \ \to \mathtt{1f},\\ &\mathtt{12} \to \mathtt{11} \to \mathtt{1f},\\ &\mathtt{12} \to \mathtt{13} \to \mathtt{1f},\\ &\mathtt{12} \to \mathtt{19} \to \mathtt{1f}\}\end{aligned}$$

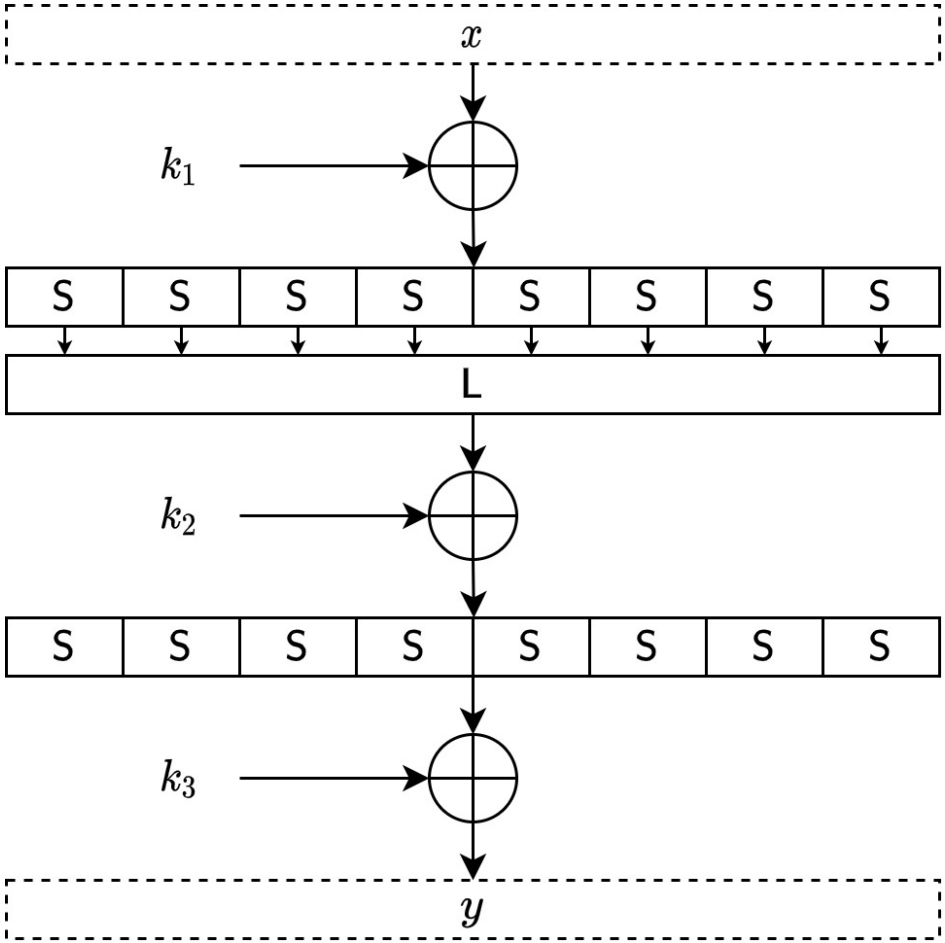
Similarly for $R = 3, 4, \ldots$

Rounds	1	2	3	4	
$\max EDP(\Delta x, \Delta y)$	2^{-2}	$2^{-3.4}$	$2^{-4.6}$	$2^{-4.8}$	non increasing, $> 2^{-5}$

Small example: comparison *EDCP*/*EDP*/*DP*

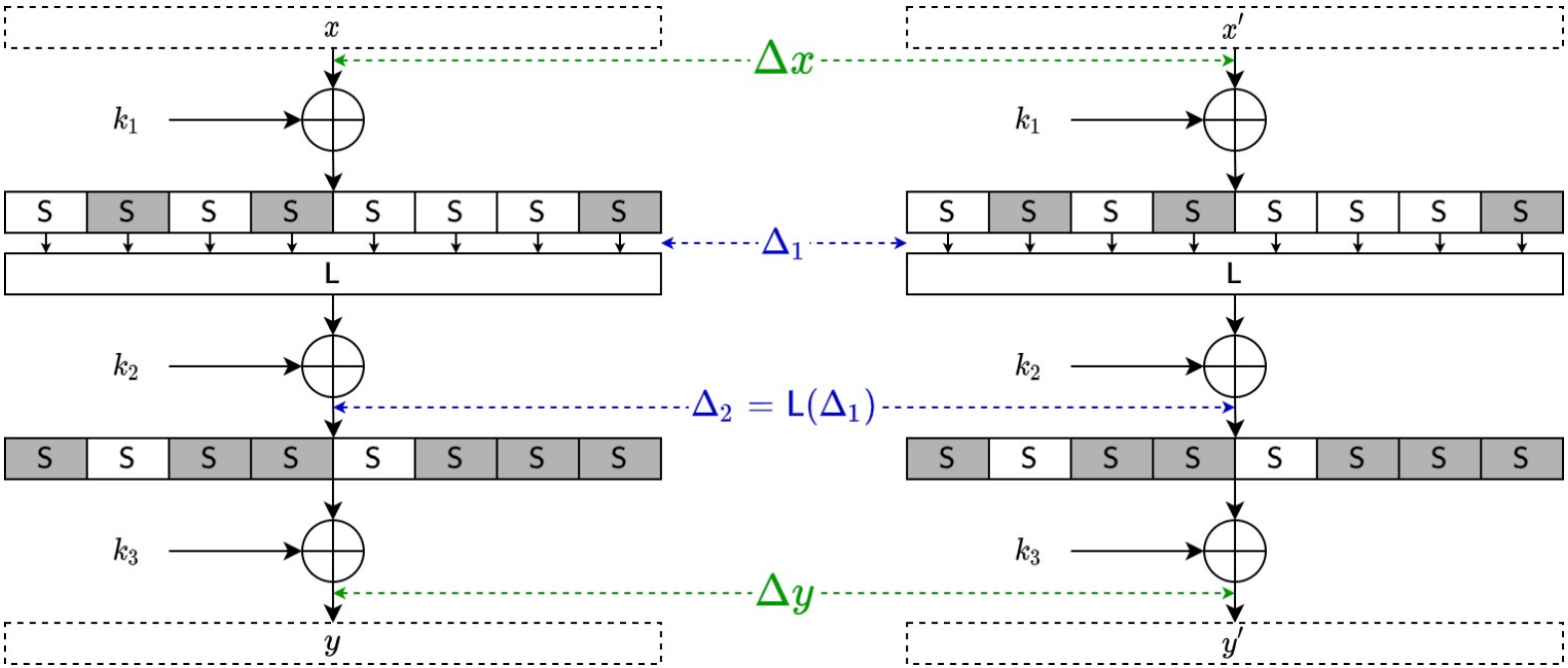
Rounds	1	2	3	4	
p^θ	2^{-2}	2^{-4}	2^{-6}	2^{-8}	decreasing
$\max EDCP(\Delta x \to \ldots \to \Delta y)$	2^{-2}	2^{-4}	$2^{-6.4}$	$2^{-9.4}$	decreasing, $\leq p^\theta$
$\max EDP(\Delta x, \Delta y)$	2^{-2}	$2^{-3.4}$	$2^{-4.6}$	$2^{-4.8}$	non increasing, $> 2^{-5}$
$\max\limits_{K \in \mathbf{F}_2^\kappa} \max\limits_{\Delta x, \Delta y \neq 0} DP(\Delta x, \Delta y)$	$\frac{8}{32}$	$\frac{10}{32}$	$\frac{14}{32}$	$\frac{16}{32}$	≤ 1

Our subject: 2-round LSX-cipher



Our subject: 2-round LSX-cipher

Difference propagation



We know:

S — difference distribution table (DDT)

L — minimal number of active Sboxes ($\mathcal{B}$)

Our goal: *MEDP* of 2R-LSX

Maximum expected differential probability

$$MEDP = \max_{\Delta x, \Delta y \neq 0} EDP(\Delta x, \Delta y)$$

Our goal: *MEDP* of 2R-LSX

Maximum expected differential probability

$$MEDP = \max_{\Delta x, \Delta y \neq 0} EDP(\Delta x, \Delta y)$$

Non-minimum weight differential

$$MEDP^{+}_{w} = \max_{\Delta x, \Delta y, \mathbf{wt}(\boldsymbol{\Delta x}) + \mathbf{wt}(\boldsymbol{\Delta y}) \geq \boldsymbol{w}} EDP(\Delta x, \Delta y),$$

where $\boldsymbol{w}$ (weight) is a minimum number of active Sboxes in the differential, $\mathcal{B} \leq w \leq 2n$.

Known results about *EDP* of 2R-LSX

Upper bounds

Theorem

For 2-round LSX-cipher

$$MEDP \leq p^{\mathcal{B}-1},$$

$$p = \max_{\Delta x, \Delta y \neq 0} \Pr\left(\mathsf{s}(x) \oplus \mathsf{s}(x \oplus \Delta x) = \Delta y\right) = \max_{\Delta x, \Delta y \neq 0} \mathrm{DDT}[\Delta x][\Delta y],$$

$\mathcal{B}$ is the minimal number of active Sboxes.

[Kang J.-S., Hong S., Lee S., Yi O., Park C., and Lim J. – *Practical and provable security against differential and linear cryptanalysis for substitution-permutation networks* – 2001]

Known results about *EDP* of 2R-LSX

Upper bounds

Let's use the DDT's rows and columns:

Theorem (FSE 2003)

For 2-round LSX-cipher

$$MEDP \leq \max\left(\max_{\Delta x \neq 0} \sum_{i=1}^{2^m-1} (\mathrm{DDT}[\Delta x][i])^{\mathcal{B}}, \max_{\Delta y \neq 0} \sum_{i=1}^{2^m-1} (\mathrm{DDT}[i][\Delta y])^{\mathcal{B}}\right)$$

[Park S., Sung S.H., Lee S., Lim J. – *Improving the Upper Bound on the Maximum Differential and the Maximum Linear Hull Probability for SPN Structures and AES* – 2003]

Known results about *EDP* of 2R-LSX

Upper bounds

Let's also use the Galois field representation of L:

Theorem

For 2-round LSX-cipher

$$MEDP \leq \max_{1 \leq t < \mathcal{B}} \max_{\mu \in \mathbf{F}_{2^m}} \max_{\lambda \in \mathbf{F}_{2^m}^*} \max_{\Delta x, \Delta y \in \mathbf{F}_{2^m}^*} \sum_{\gamma \in \mathbf{F}_{2^m}^*} \left(\mathrm{DDT}[\Delta x][\gamma]\right)^t \left(\mathrm{DDT}[\lambda \cdot \gamma \oplus \mu][\Delta y]\right)^{\mathcal{B}-t}$$

[Canteaut A., Roué J. – *On the behaviors of affine equivalent sboxes regarding differential and linear attacks* – 2015]

Known results about *EDP* of 2R-LSX

AES

[Keliher L., Sui J. – *Exact Maximum Expected Differential and Linear Probability for 2-Round Advanced Encryption Standard (AES)* – 2007]

- 2-round AES: 32-bit block size, $\mathcal{B} = 5$
- Recursive algorithm over **all** differentials
- $\mathrm{MEDP}_{\text{2-round}} = 2^{-28.272\ldots} < 2^{-32}$

Known results about *EDP* of 2R-LSX

AES

[Keliher L., Sui J. – *Exact Maximum Expected Differential and Linear Probability for 2-Round Advanced Encryption Standard (AES)* – 2007]

- 2-round AES: 32-bit block size, $\mathcal{B} = 5$
- Recursive algorithm over **all** differentials
- $\mathrm{MEDP}_{\text{2-round}} = 2^{-28.272\ldots} < 2^{-32}$

[Sano F., Ohkuma K., Shimizu H., Kawamura S. – *On the security of nested SPN cipher against the differential and linear cryptanalysis* – 2003]

- 4-round AES: nested structure, SuperSbox representation
- $\mathrm{MEDP}_{\text{4-round}} \leq (\mathrm{MEDP}_{\text{2-round}})^4 = 2^{-113\ldots} < 2^{-128}$

Known results about *EDP* of 2R-LSX

Kuznyechik

[Kiryukhin V. – *Exact maximum expected differential and linear probability for 2-round Kuznyechik* – CTCrypt'18]

- 2-round Kuznyechik: 128-bit block size, $\mathcal{B} = 17$
- Recursive algorithm over minimum weight differentials ($w = \mathcal{B}$)
- Ad-hoc lemma for upper bounding non-minimum weight differentials ($w > \mathcal{B}$)
- $MEDP = 2^{-86.66\ldots}$

Known results about *EDP* of 2R-LSX

[Canteaut A., Roué J. – *Differential Attacks Against SPN: A Thorough Analysis* – 2015]

- Differentials of weight $w = n + 2$: upper bound on the number of characteristics within a differential
- Some results about 2-round LSX with APN Sboxes
- **«MEDP can be tight for a differential of non-minimal weight»**

New approach

We propose:

Dynamic programming algorithm designed for bounding non-minimum weight differentials in 2-round LSX-ciphers

New approach

We propose:

Dynamic programming algorithm designed for bounding non-minimum weight differentials in 2-round LSX-ciphers

- Only DDT and the differential branch number $\mathcal{B}$ are used

New approach

We propose:

Dynamic programming algorithm designed for bounding non-minimum weight differentials in 2-round LSX-ciphers

- Only DDT and the differential branch number $\mathcal{B}$ are used
- The number of high probability differential trails is minimized

New approach

We propose:

Dynamic programming algorithm designed for bounding non-minimum weight differentials in 2-round LSX-ciphers

- Only DDT and the differential branch number $\mathcal{B}$ are used
- The number of high probability differential trails is minimized
- Total number of trails is NOT minimized

New approach

We propose:

Dynamic programming algorithm designed for bounding non-minimum weight differentials in 2-round LSX-ciphers

- Only DDT and the differential branch number $\mathcal{B}$ are used
- The number of high probability differential trails is minimized
- Total number of trails is NOT minimized
- The main goal is «heavy» LSX-ciphers

Preliminary

Trail $\leftrightarrow$ codeword

- The code $\mathcal{C}_{\mathsf{L}} = \left\{ (\mathbf{c}, \mathsf{L}(\mathbf{c})) \,,\ \mathbf{c} \in \mathbf{F}_{2^8}^n \right\}$ of length $2n$
- The differential branch number $\mathcal{B}$ is the minimum distance of the code $\mathcal{C}_{\mathsf{L}}$
- $\Omega = (\Delta x, \Delta_1, \Delta_2, \Delta y)$ in 2-round differential $(\Delta x, \Delta y) \leftrightarrow$ a codeword (Δ_1, Δ_2) in $\mathcal{C}_{\mathcal{L}}$

Strategy

For an arbitrary (hypothetical) differential $(\Delta x, \Delta y)$:

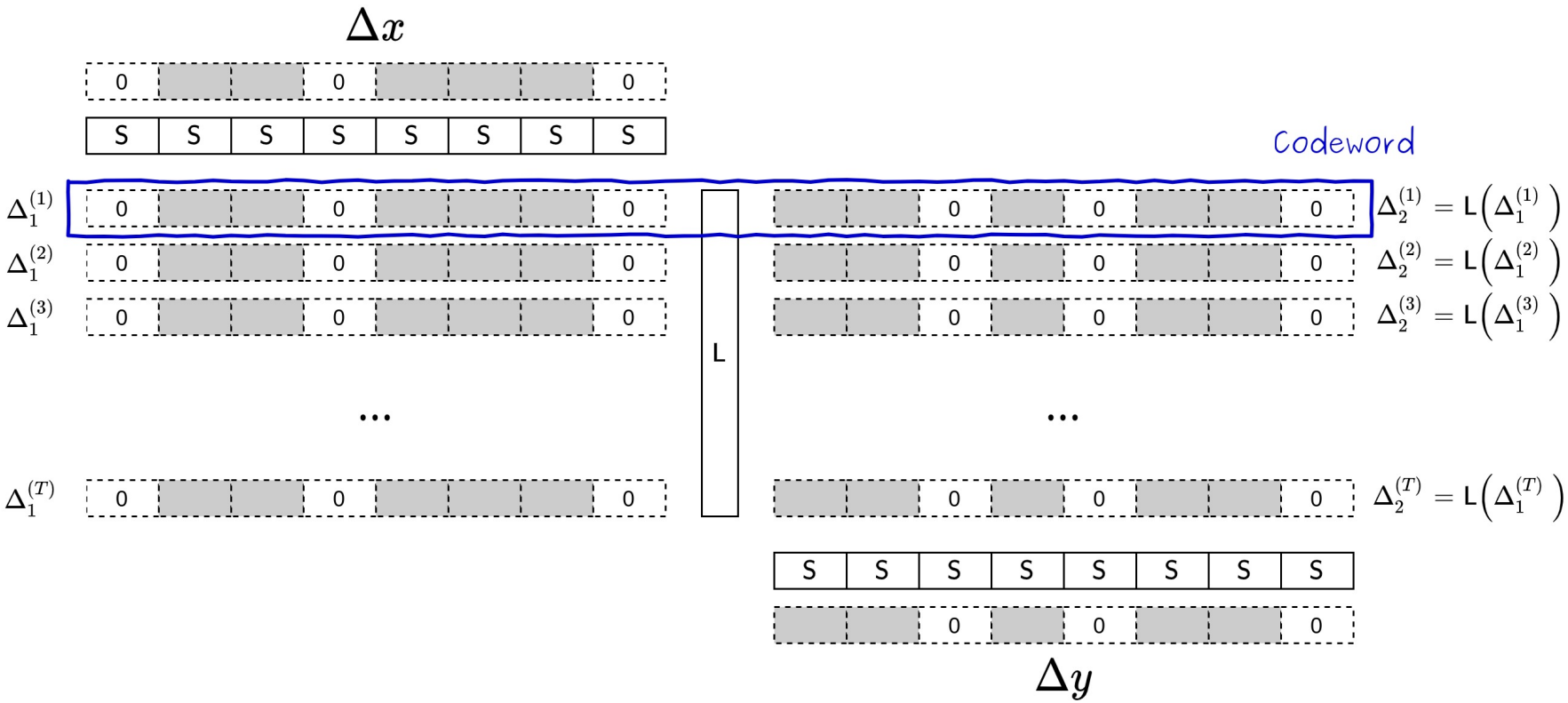
- consider a set of all possible trails = codewords
- derive a constraints («maximum cost») for the entire set
- divide the set into several subsets
- compute the constraints («cost») and upper bound («value») for each possible subset

Dynamic programming

Let's select subsets:

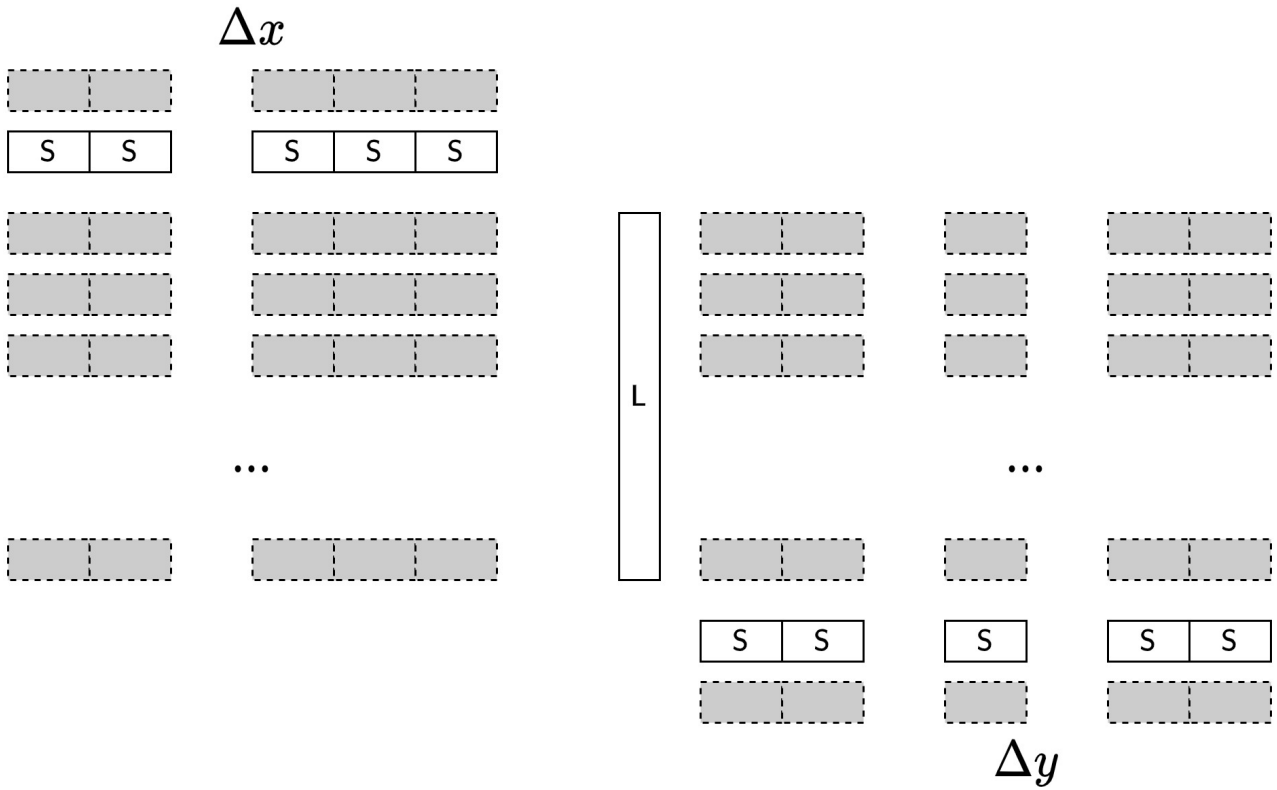
- the upper bound («total value») is maximum
- selection satisfies all constraints («total cost» does not exceed «maximum cost»)

Codewords


Δx
0 0 0
S S S S S S S S
Codeword
$\Delta_1^{(1)}$ 0 0 0
$\Delta_1^{(2)}$ 0 0 0
$\Delta_1^{(3)}$ 0 0 0
...
$\Delta_1^{(T)}$ 0 0 0
L
0 0 0 $\Delta_2^{(1)} = \mathsf{L}\left(\Delta_1^{(1)}\right)$
0 0 0 $\Delta_2^{(2)} = \mathsf{L}\left(\Delta_1^{(2)}\right)$
0 0 0 $\Delta_2^{(3)} = \mathsf{L}\left(\Delta_1^{(3)}\right)$
...
0 0 0 $\Delta_2^{(T)} = \mathsf{L}\left(\Delta_1^{(T)}\right)$
S S S S S S S S
0 0 0
Δy

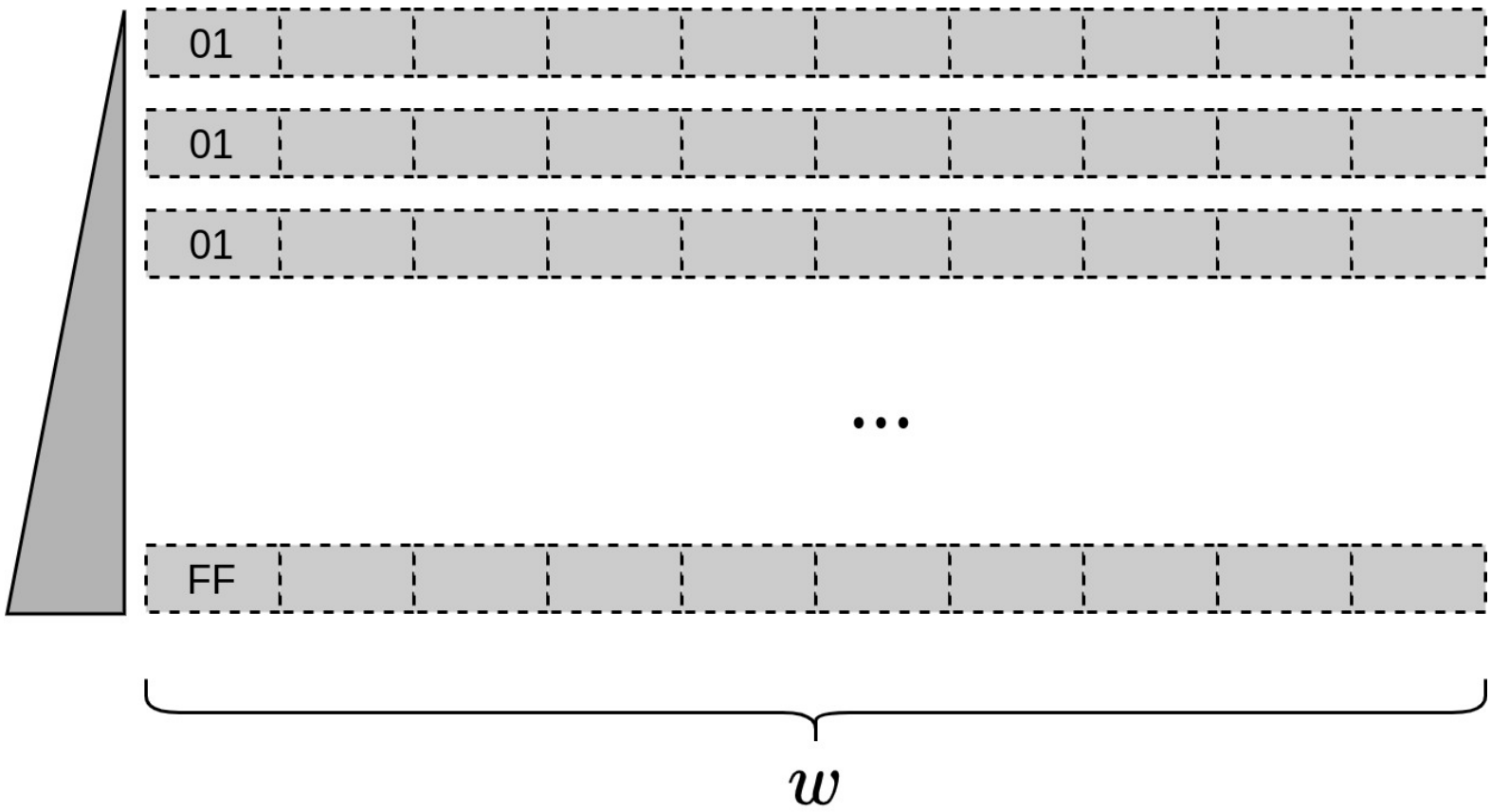
Codewords

Remove the zero coordinates



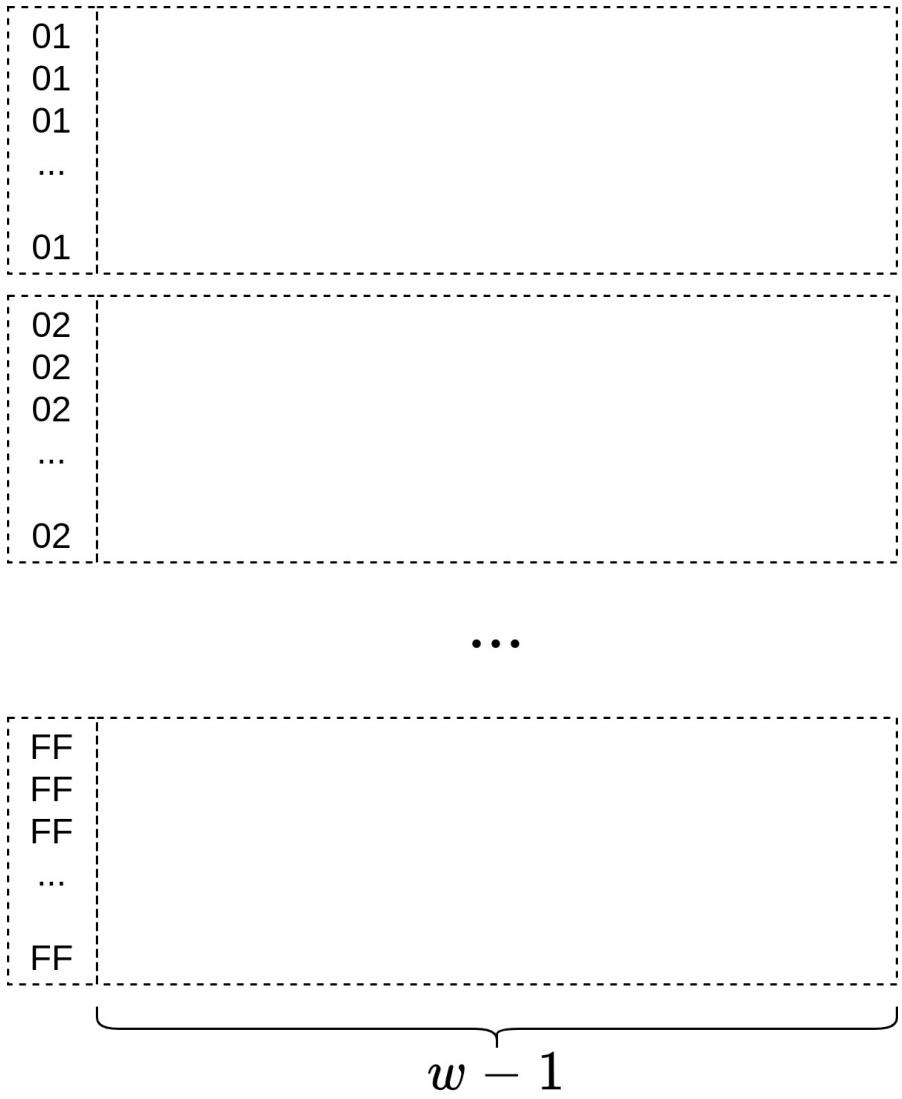
Codewords

Sorting by the first coordinate



Codewords

- Consider the subsets of codewords
- Each element b of the table corresponds to $\mathrm{DDT}[\Delta x_i][b]$ or $\mathrm{DDT}[b][\Delta y_i]$



DDT simplification

We «replace» each row/column with the «maximum» row/column

Example

Kuznyechik: $\frac{\mathbf{8}}{\mathbf{256}}, \frac{\mathbf{8}}{\mathbf{256}}, \frac{6}{256}, \ldots, \frac{6}{256}, \frac{4}{256}, \ldots, \frac{4}{256}, \frac{2}{256}, \frac{2}{256}, \ldots, \frac{2}{256}, 0, \ldots, 0$

$\boldsymbol{p} = {}^{8}\!/_{256}$, $\boldsymbol{\nu} = 2$

AES: $\frac{\mathbf{4}}{\mathbf{256}}, \frac{2}{256}, \frac{2}{256}, \ldots, \frac{2}{256}, 0, \ldots, 0$ — no change

$\boldsymbol{p} = {}^{4}\!/_{256}$, $\boldsymbol{\nu} = 1$

Constraints

Basic idea

Many «rows» with many $\boldsymbol{p}$ **can't** exist.

This would contradict the properties of the code $\mathcal{C}_{\mathrm{L}}$.

Lemma

Let $w = \mathcal{B} + 1$ *and* $\boldsymbol{\omega_q}$ *is the number of rows containing exactly* $\boldsymbol{q}$ *elements* $\boldsymbol{p}$. *Then*

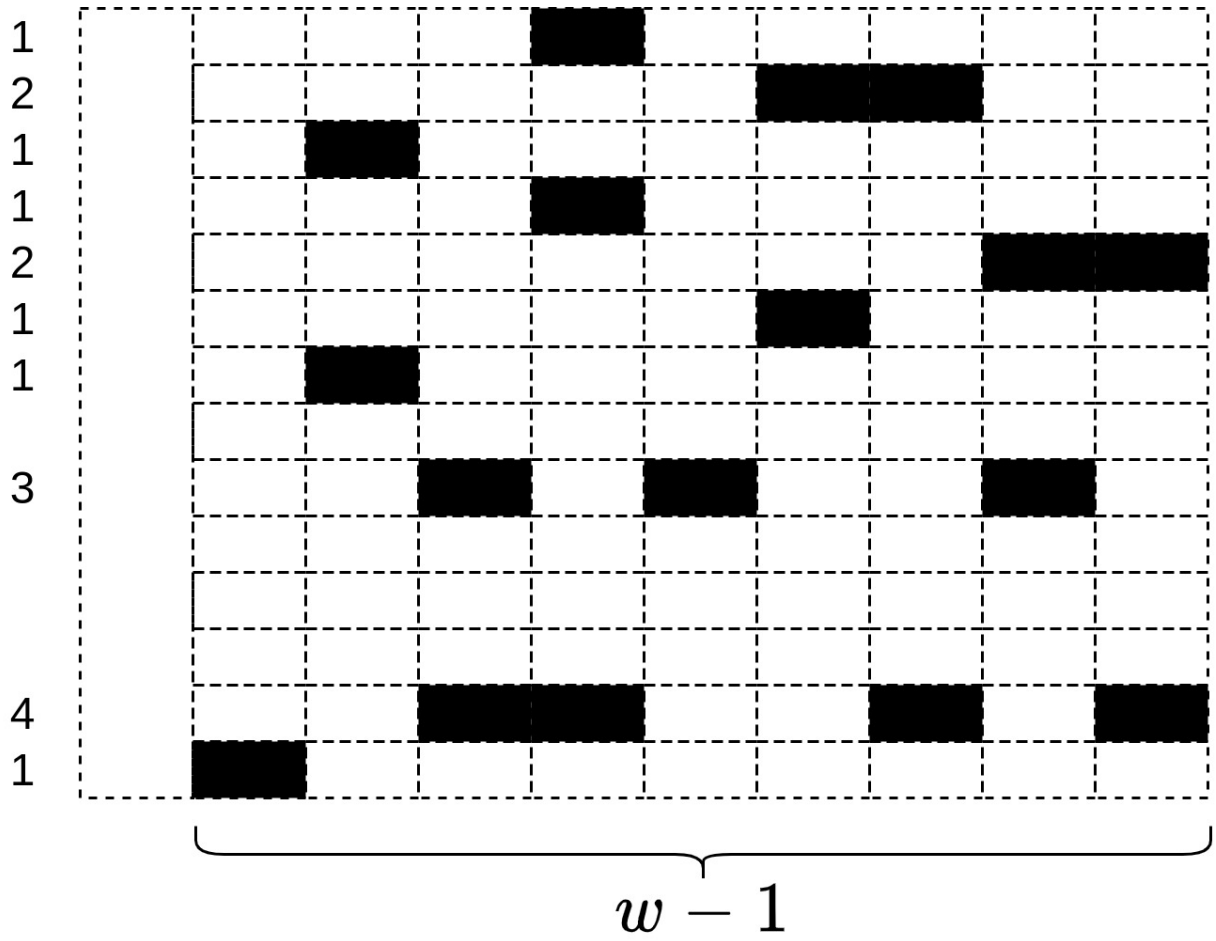
$$\sum_{q=2}^{w-1} \omega_q \cdot \binom{q}{2} \leq \binom{w-1}{2} \cdot \nu^2.$$

By analogy for $w > \mathcal{B} + 1$

Constraints can also be extended in other ways...

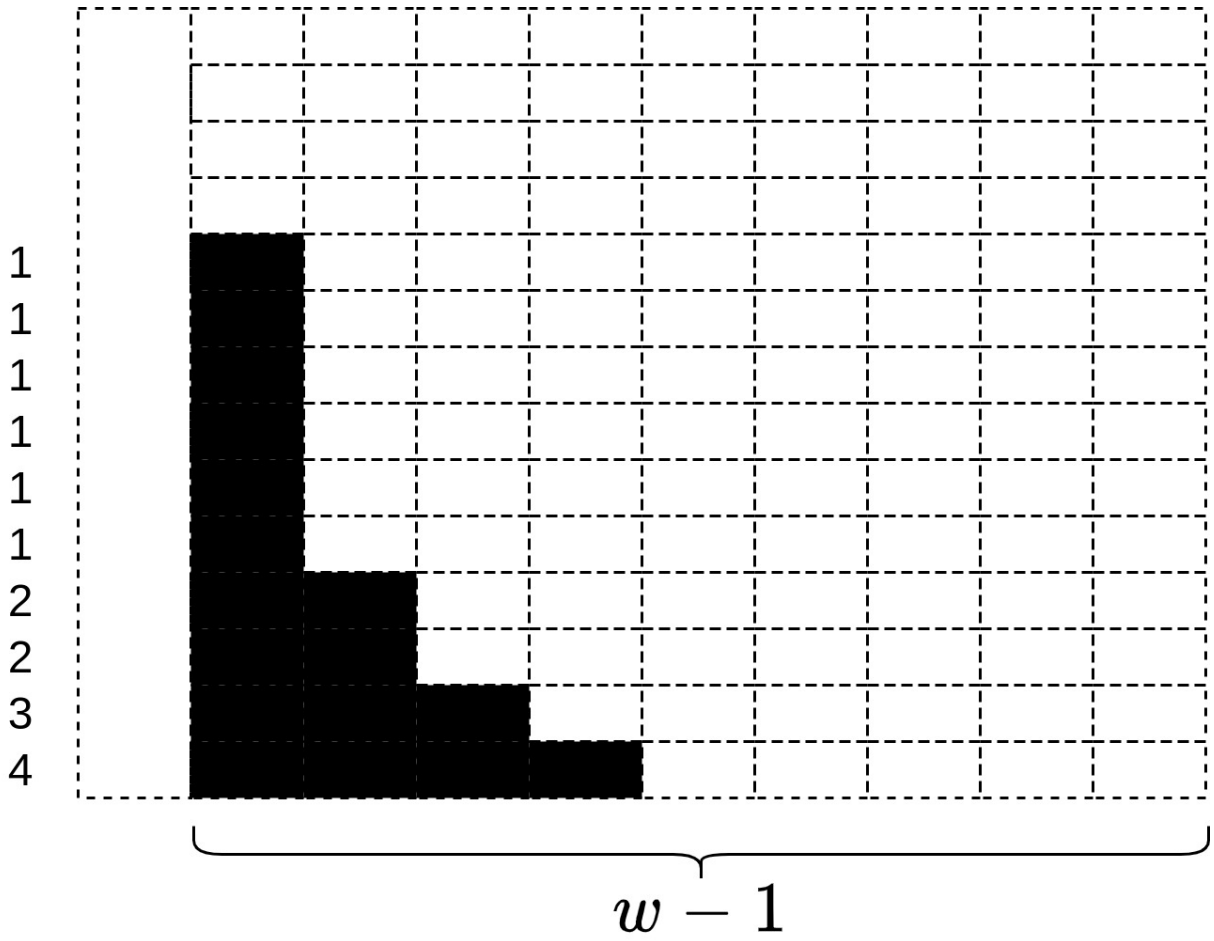
Upper bound on subset

Let we know the «distribution» $\omega_1, \omega_2 \ldots$ of $\boldsymbol{p}$ among codewords («rows»).



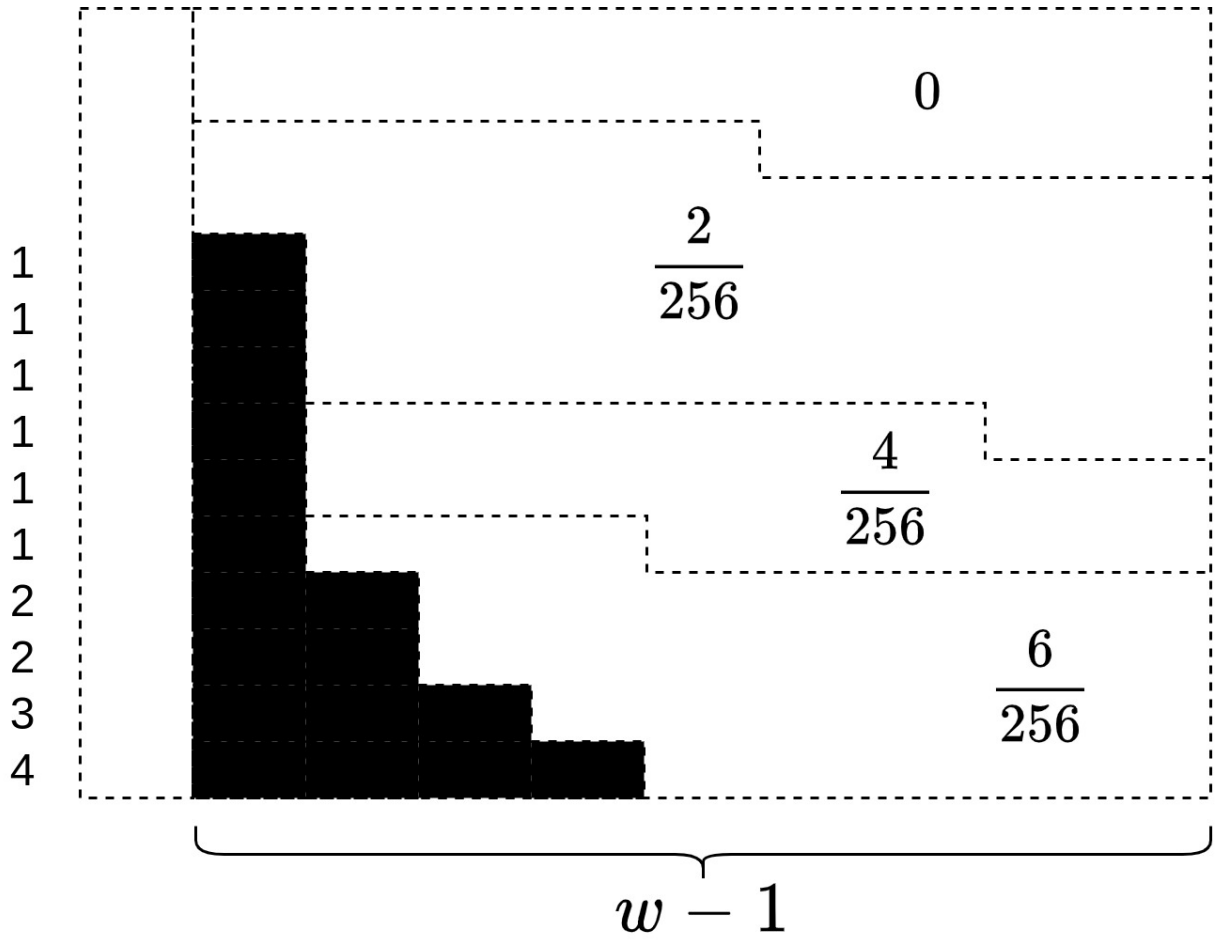
Upper bound on subset

Let we know the «distribution» $\omega_1, \omega_2 \ldots$ of $\boldsymbol{p}$ among codewords («rows»).



Upper bound on subset

Upper bound



Optimization problem

1. Consider each possible set W_i of ω_1, ω_2...

Optimization problem

1. Consider each possible set W_i of ω_1, ω_2...
2. Compute «cost» of W_i: $C_i = \sum_{q=2}^{w-1} \omega_q \cdot \binom{q}{2}$

Optimization problem

1. Consider each possible set W_i of ω_1, ω_2...
2. Compute «cost» of W_i: $C_i = \sum_{q=2}^{w-1} \omega_q \cdot \binom{q}{2}$
3. Derive upper bound on W_i («value» V_i)

Optimization problem

1. Consider each possible set W_i of ω_1, ω_2...
2. Compute «cost» of W_i: $C_i = \sum_{q=2}^{w-1} \omega_q \cdot \binom{q}{2}$
3. Derive upper bound on W_i («value» V_i)
4. Choose $I = \{i_1, i_2, \ldots\}$

$$\mathrm{MEDP}^{+}_{\mathcal{B}+1} \leq \max_{I} \sum_{j \in \mathbf{F}_2^m} \boldsymbol{\rho}[j] \cdot V_{i_j}, \text{ under condition } \sum_{i \in I} C_i \leq \binom{w-1}{2} \cdot \nu^2,$$

where $\boldsymbol{\rho}$ is row/column of the DDT

How to increase weight?

- The algorithm starts with $w = \mathcal{B} + 1$
- Each solution for $w = \mathcal{B} + 1$ has own «cost» for the case $w = \mathcal{B} + 2$
- Maximize the «cost» in case of $\mathcal{B} + 1$ under each «cost» for the case $\mathcal{B} + 2$
- Solve optimization problem for $\mathcal{B} + 2$
- etc.
- The result (upper bound) does **not increase** when w changes to $w + 1$

Main applicability

Non-trivial upper bounds for all non-minimum weight 2-round differentials $\Rightarrow$

Main applicability

Non-trivial upper bounds for all non-minimum weight 2-round differentials $\Rightarrow$

1. Easy way to compute the exact value of *MEDP* for 2-round

Main applicability

Non-trivial upper bounds for all non-minimum weight 2-round differentials $\Rightarrow$

1. Easy way to compute the exact value of *MEDP* for 2-round
2. New upper bounds on *MEDP* for **more** rounds
   - [Keliher L. – *Linear Cryptanalysis of Substitution-Permutation Networks* – 2003]
   - [Keliher L. – *Refined Analysis of Bounds Related to Linear and Differential Cryptanalysis for the AES* – 2004]

New results: Kuznyechik

2-round Kuznyechik

- $\mathrm{MEDP}^{+}_{\mathcal{B}+1} = \max\limits_{\Delta x, \Delta y, \mathrm{wt}(\Delta x)+\mathrm{wt}(\Delta y) \geq 18} EDP(\Delta x, \Delta y) \leq 2^{-87.54\ldots}$
- $\mathrm{MEDP}^{+}_{\mathcal{B}+2} = \max\limits_{\Delta x, \Delta y, \mathrm{wt}(\Delta x)+\mathrm{wt}(\Delta y) \geq 19} EDP(\Delta x, \Delta y) \leq 2^{-88.34\ldots}$

A small but non-trivial decrease relative to $MEDP = 2^{-86.66\ldots}$.

New results: Kuznyechik

$$MEDP_{\text{R-round}} \leq \ldots \leq MEDP_{\text{3-round}} \leq MEDP_{\text{2-round}} = 2^{-86.66\ldots}$$

New result:

$$MEDP_{\text{R-round}} \leq \ldots \leq MEDP_{\text{3-round}} \leq 2^{-88.34\ldots} \approx \mathrm{MEDP}^{+}_{\text{2-round},\mathcal{B}+2}$$

$\Rightarrow$ A small but non-trivial decrease of upper bound on $MEDP_{\text{3-round}}$ relative to $MEDP_{\text{2-round}}$

Khazad

- LSX-cipher
- 64-bit block
- 128-bit key
- S and L are involutions, i.e. $\mathsf{S} = \mathsf{S}^{-1}$, $\mathsf{L} = \mathsf{L}^{-1}$
- $p = \frac{8}{256}$
- MDS-matrix, $\mathcal{B} = 9$

New results: Khazad

2-round Khazad

- 8 best differentials $(\Delta x, \Delta y)$ and 8 «involution» differentials $(\Delta y, \Delta x)$
- $\mathrm{MEDP} = \mathrm{MEDP}_{\mathcal{B}} = 2^{-45} + 2^{-60} = 2^{-49.99\ldots}$
- $\mathrm{MEDP}^{+}_{\mathcal{B}+1} \leq 2^{-45.02\ldots}$
- $\mathrm{MEDP}^{+}_{\mathcal{B}+2} \leq 2^{-45.09\ldots}$

New results: Khazad

2-round Khazad

Example of the best 2-round differential

Δx	1208f0000000000f		$\log_2 \mathrm{EDCP}(\Omega_i)$
Ω_1	1248f0000000000f	0000b548fbeb4800	-45
Ω_2	c8070a0000000023	0000130753a60700	-60
Δy		0000bf0818910800	

Conclusion

- New approach for bounding non-minimum weight differentials in 2-round LSX-ciphers

Conclusion

- New approach for bounding non-minimum weight differentials in 2-round LSX-ciphers
- Small improvement of upper bound on R-round $MEDP$ of Kuznyechik

Conclusion

- New approach for bounding non-minimum weight differentials in 2-round LSX-ciphers
- Small improvement of upper bound on R-round $MEDP$ of Kuznyechik
- All the best differentials for 2-round Khazad

Conclusion

- New approach for bounding non-minimum weight differentials in 2-round LSX-ciphers
- Small improvement of upper bound on R-round $MEDP$ of Kuznyechik
- All the best differentials for 2-round Khazad
- Similar results for linear cryptanalysis (ELP, $MELP$)

Conclusion

- New approach for bounding non-minimum weight differentials in 2-round LSX-ciphers
- Small improvement of upper bound on R-round $MEDP$ of Kuznyechik
- All the best differentials for 2-round Khazad
- Similar results for linear cryptanalysis (ELP, $MELP$)
- New results for Streebog, Whirlpool and other «heavy» LSX

Conclusion

- New approach for bounding non-minimum weight differentials in 2-round LSX-ciphers
- Small improvement of upper bound on R-round $MEDP$ of Kuznyechik
- All the best differentials for 2-round Khazad
- Similar results for linear cryptanalysis (ELP, $MELP$)
- New results for Streebog, Whirlpool and other «heavy» LSX
- Source codes https://gitlab.com/v.kir/diff2rLSX

Thank you for attention!

Questions?