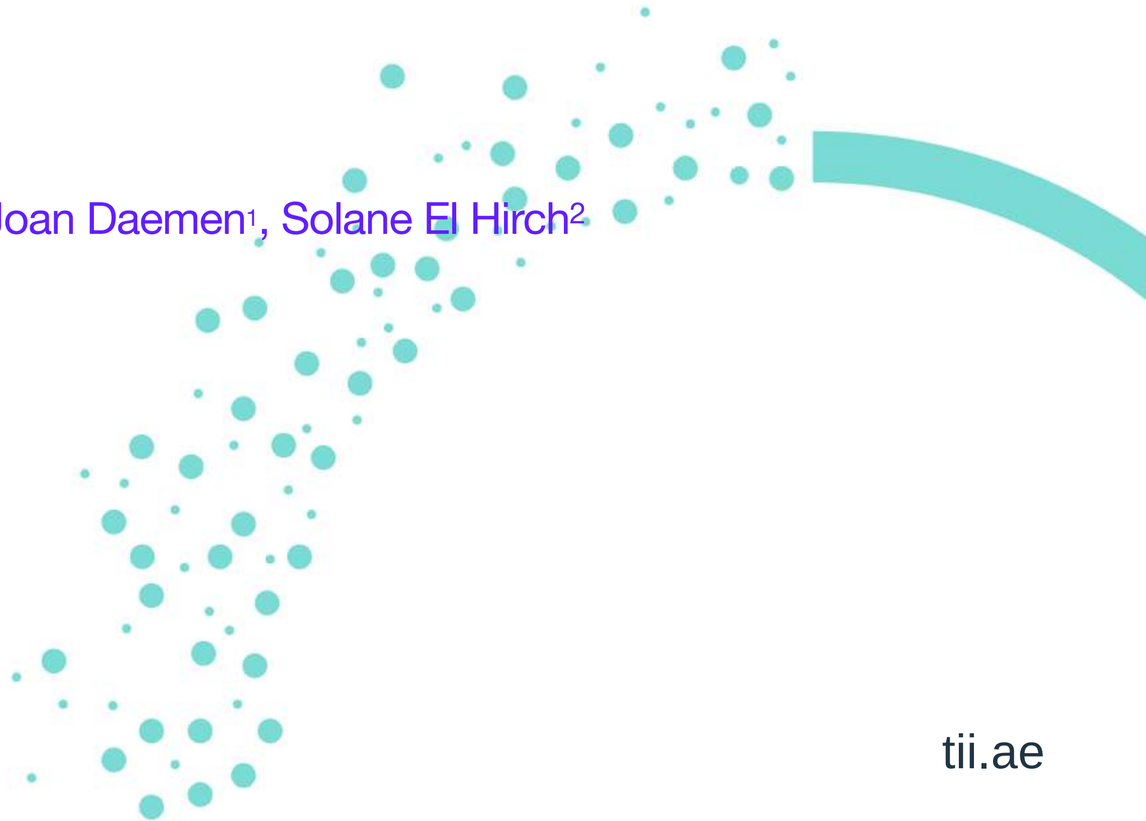
Multi-GPU Acceleration of High-order Avalanche Tests for Symmetric Ciphers

Emanuelle Bellini[1], Juan Grados[1], Mohamed Rachidi[1], Nitin Satpute[1], Joan Daemen[1], Solane El Hirch[2]

1 Technology Innovation Institute, Abu Dhabi, UAE
2 Radboud University, Radboud, Netherlands



Cryptography
Research Center

tii.ae

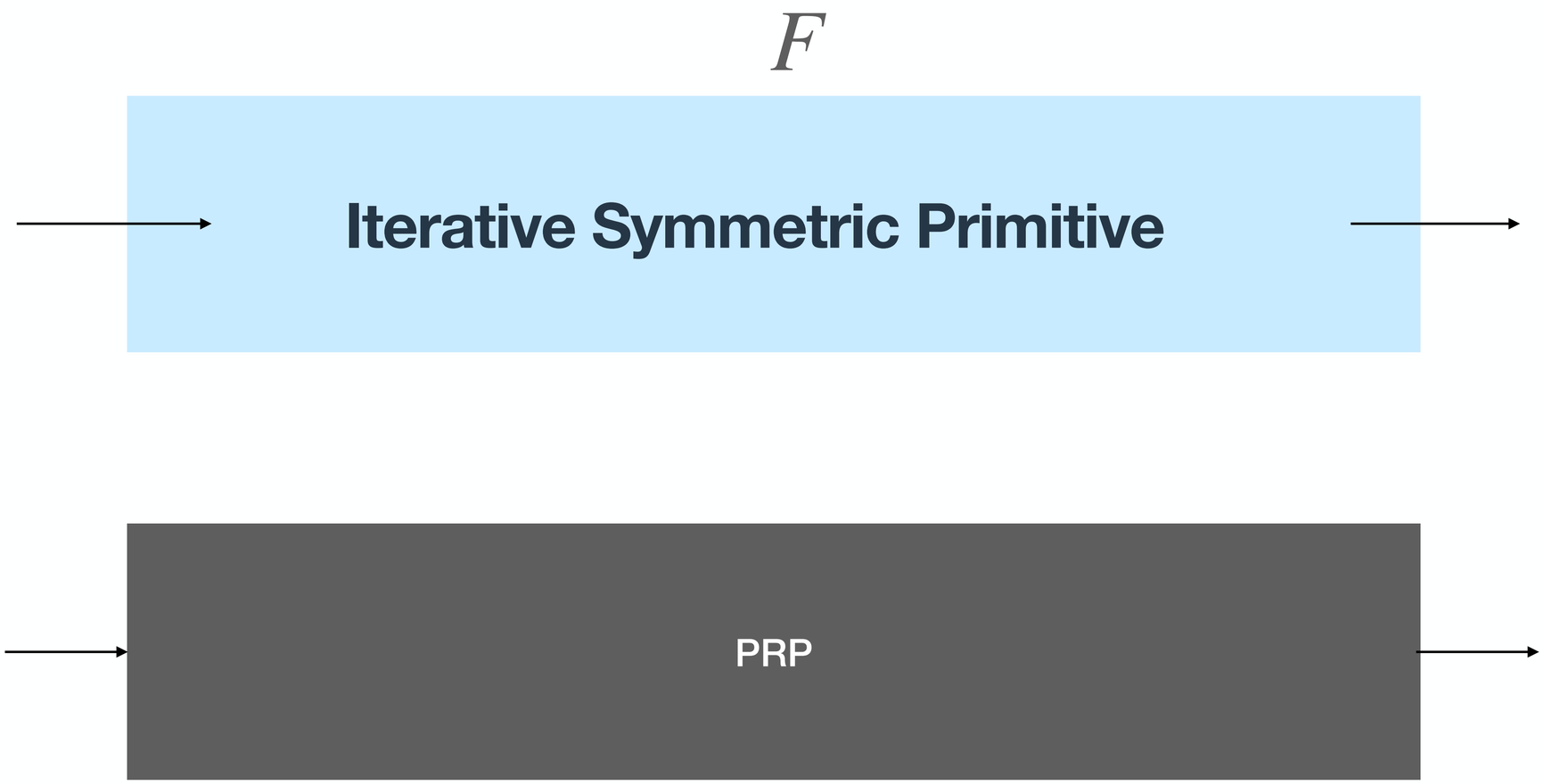
Outline

- First-Order avalanche tests
- High-Order avalanche tests
- GPU-implementation
- Results on ASCON
- Conclusion

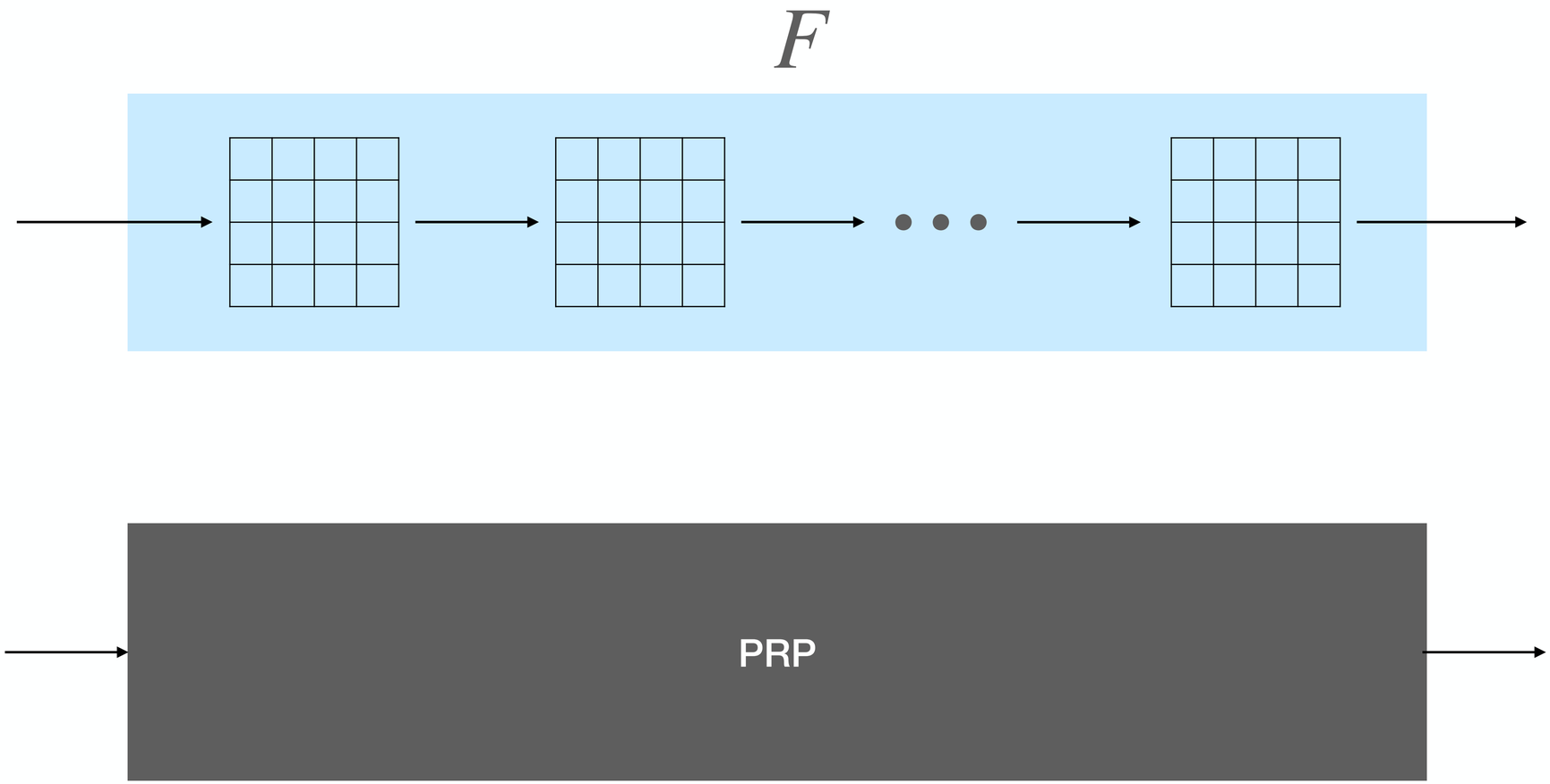
Iterative Symmetric Primitive vs Pseudo-Random Permutation (PRP)

Iterative Symmetric Primitive vs Pseudo-Random Permutation (PRP)

Iterative Symmetric Primitive vs Pseudo-Random Permutation (PRP)



Iterative Symmetric Primitive vs Pseudo-Random Permutation (PRP)



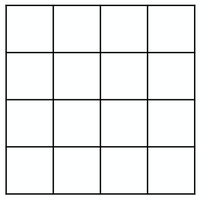
Motivation

Truncated Differential Distinguisher of a 3-round Iterative symmetric cipher

Motivation

Truncated Differential Distinguisher of a 3-round Iterative symmetric cipher

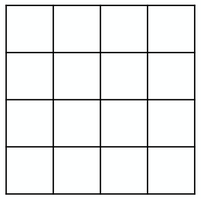
$$x \xleftarrow{\$} \mathbb{F}_2^{16}$$



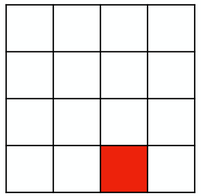
Motivation

Truncated Differential Distinguisher of a 3-round Iterative symmetric cipher

$$x \xleftarrow{\$} \mathbb{F}_2^{16}$$



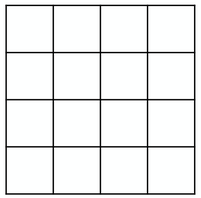
$$\Delta_{in}$$



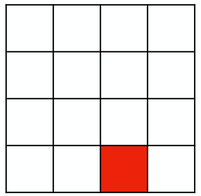
Motivation

Truncated Differential Distinguisher of a 3-round Iterative symmetric cipher

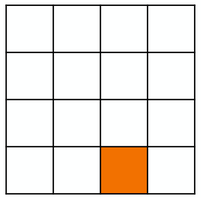
$x \xleftarrow{\$} \mathbb{F}_2^{16}$



$\oplus$

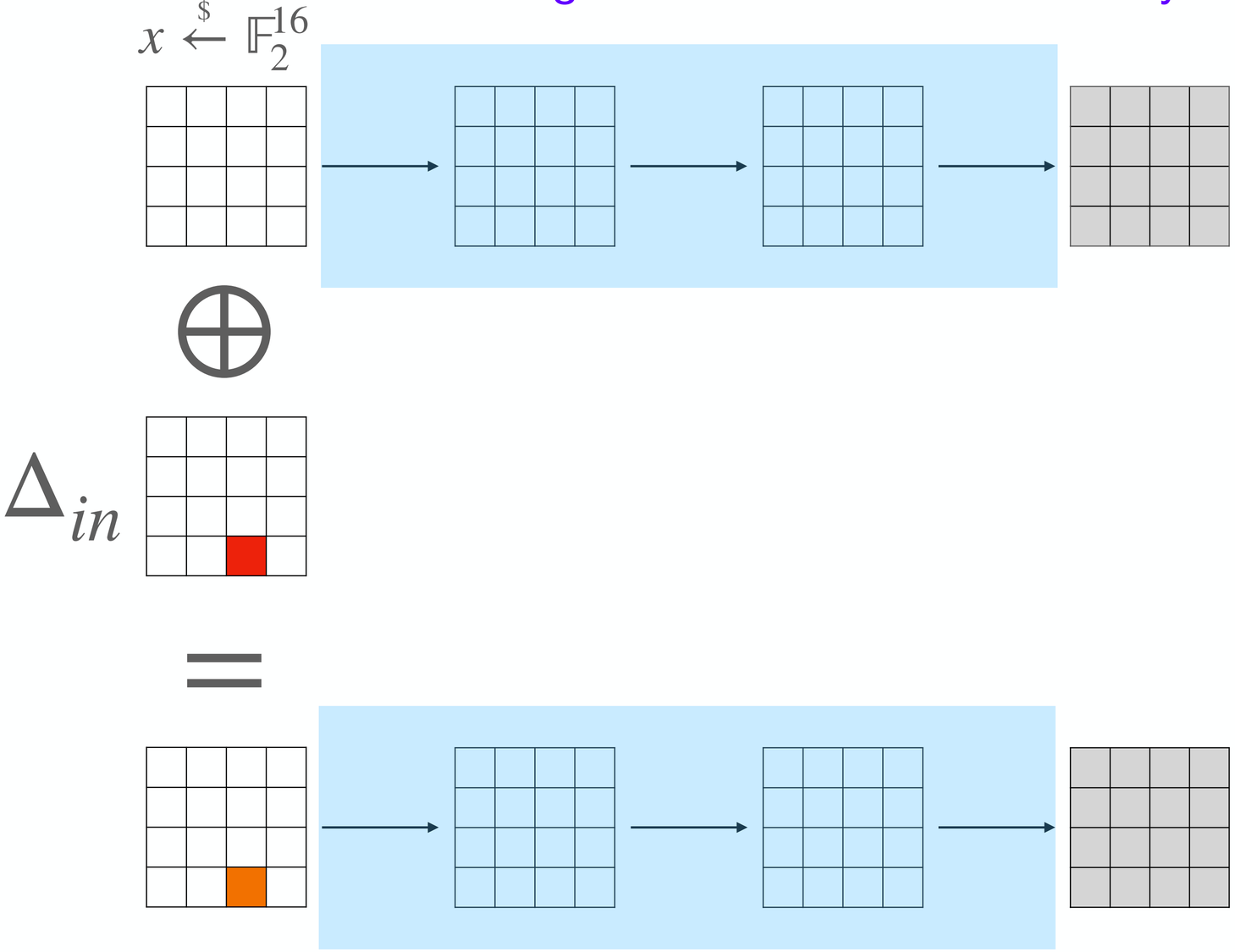
Δ_{in} 

$=$



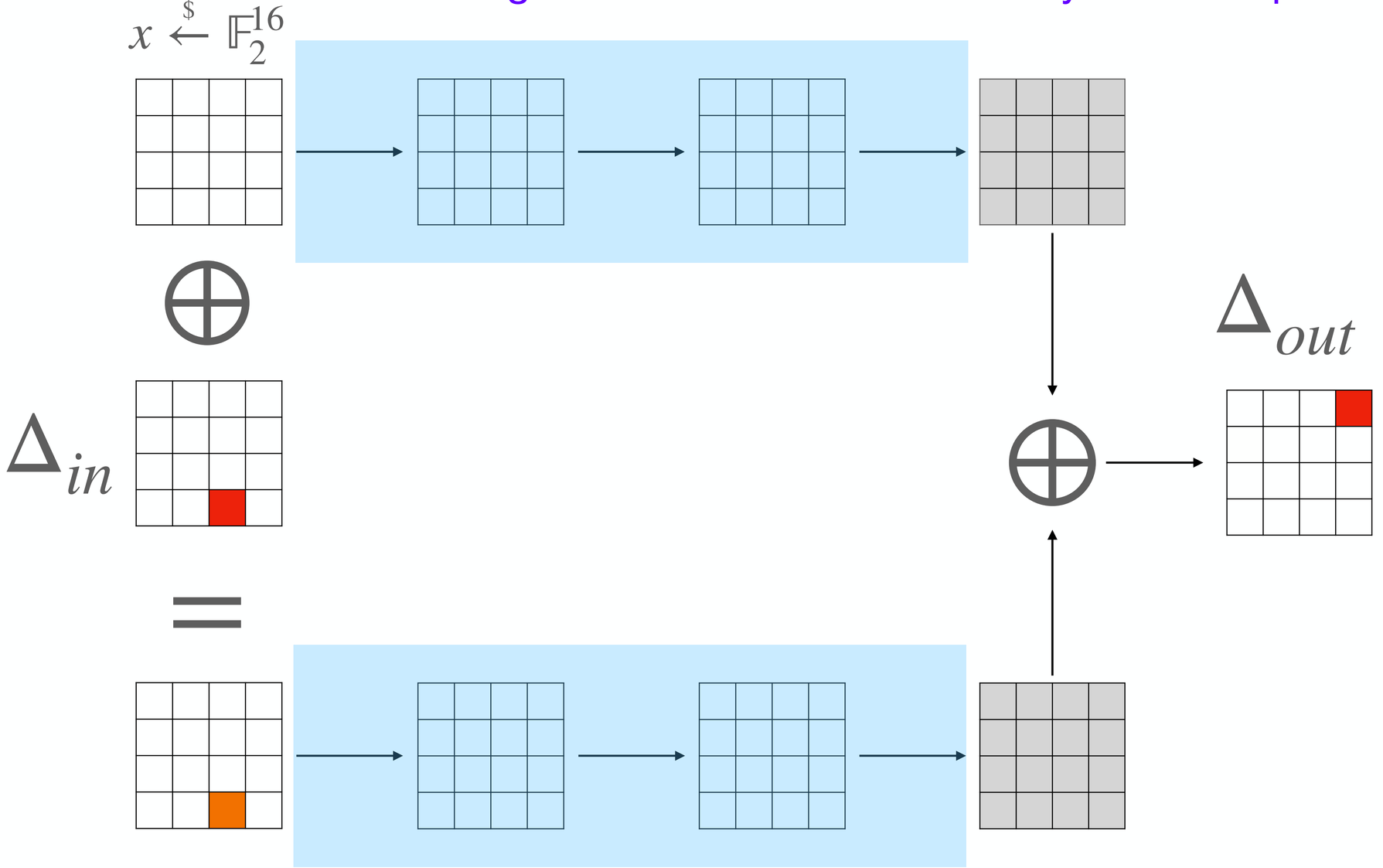
Motivation

Truncated Differential Distinguisher of a 3-round Iterative symmetric cipher



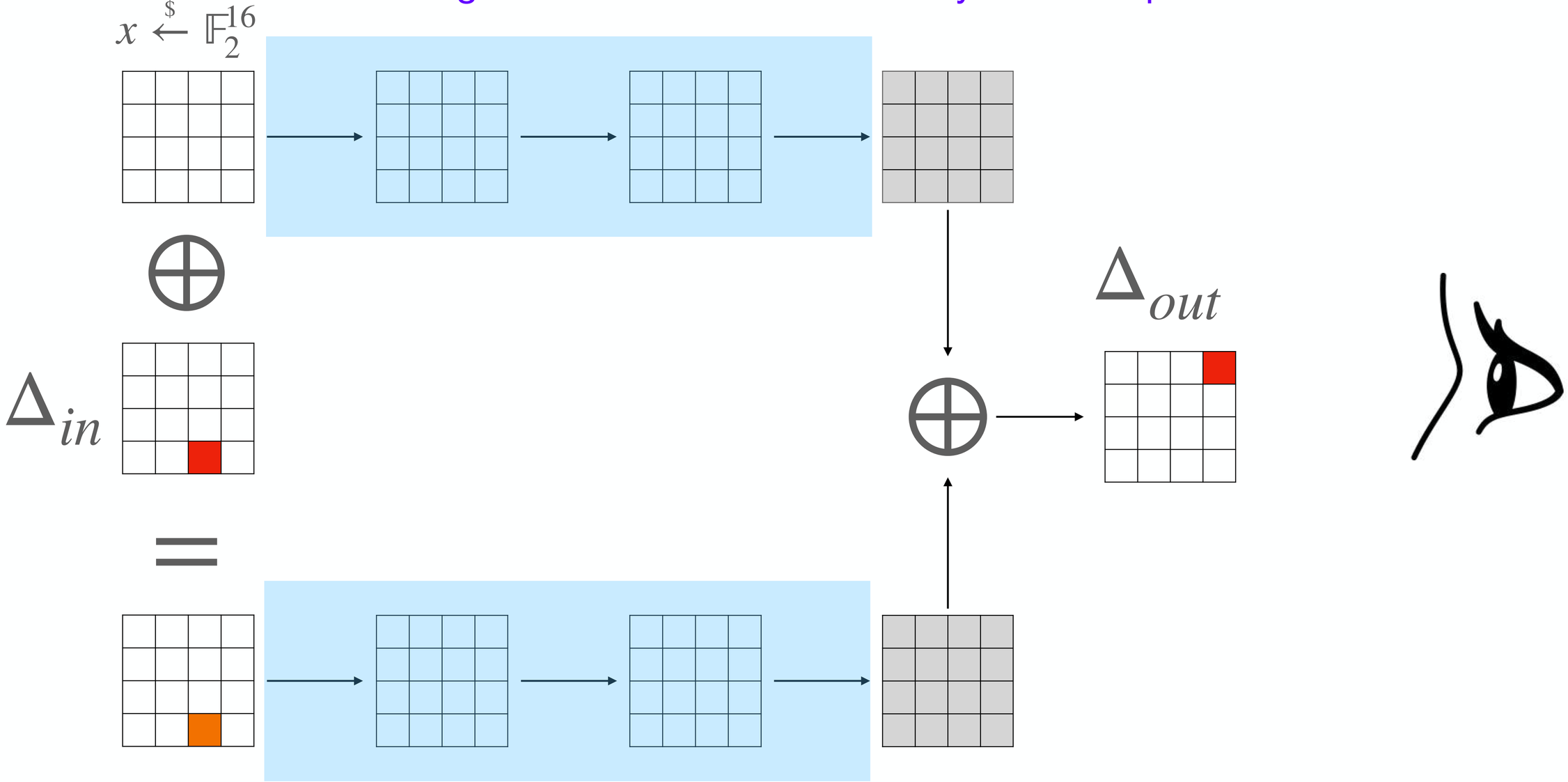
Motivation

Truncated Differential Distinguisher of a 3-round Iterative symmetric cipher



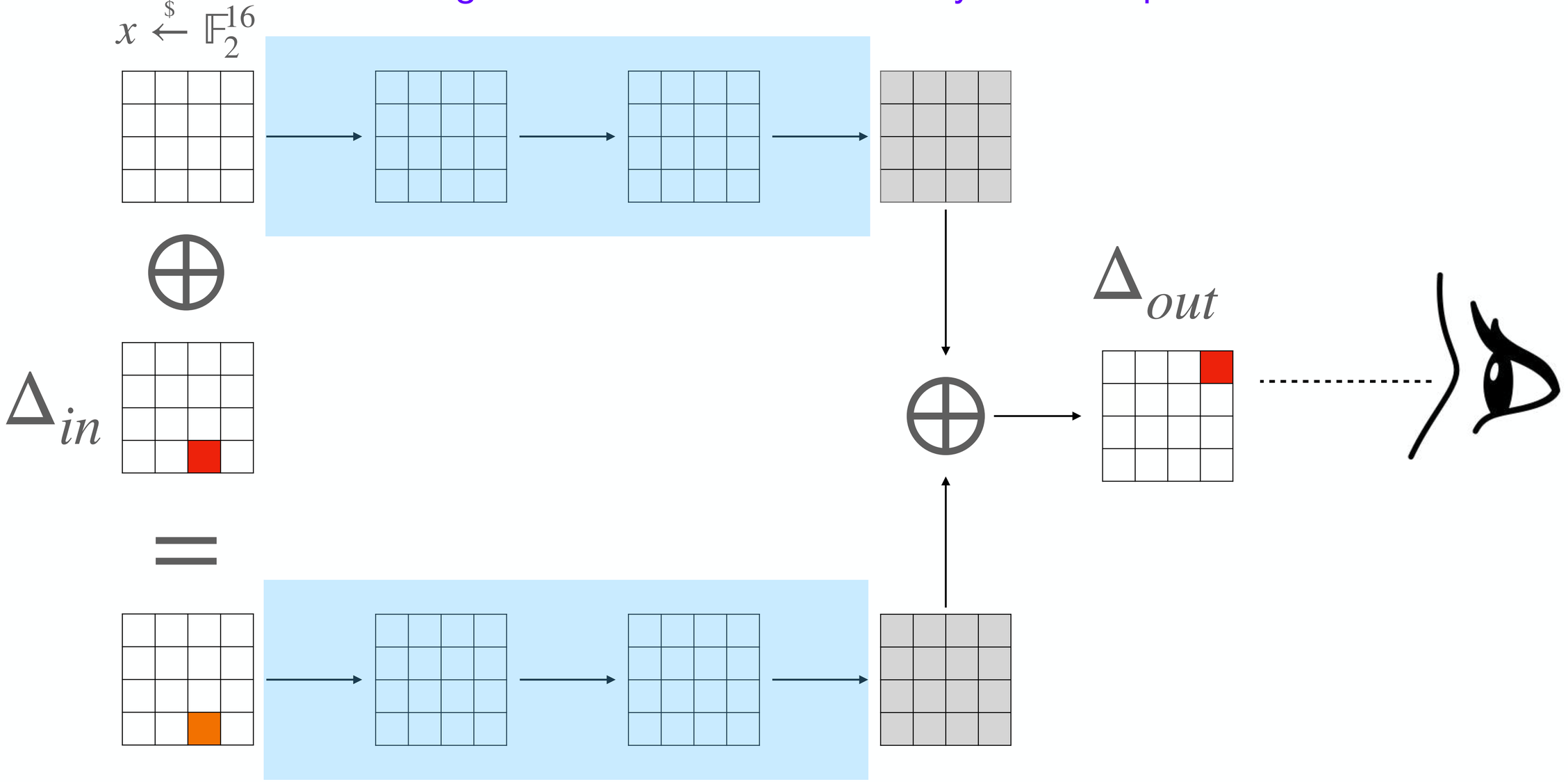
Motivation

Truncated Differential Distinguisher of a 3-round Iterative symmetric cipher



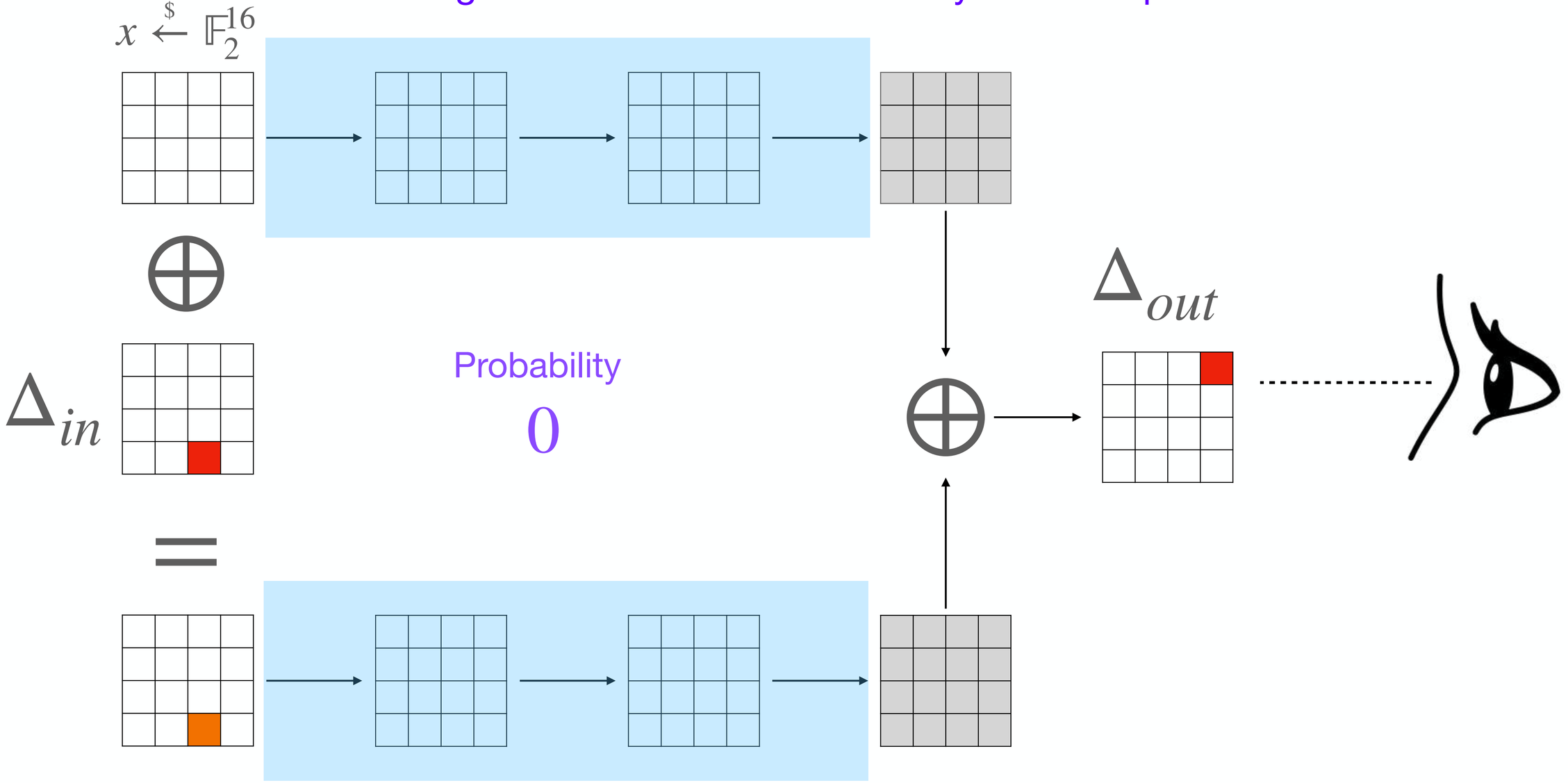
Motivation

Truncated Differential Distinguisher of a 3-round Iterative symmetric cipher



Motivation

Truncated Differential Distinguisher of a 3-round Iterative symmetric cipher



Avalanche Dependence [Webster85]

Metric

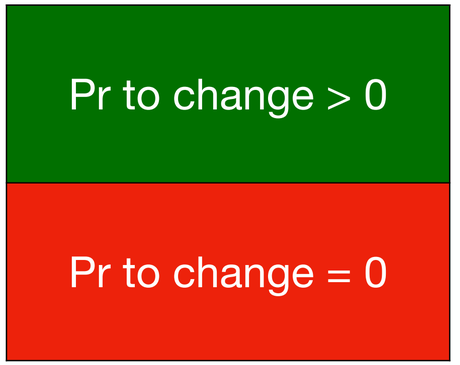
Number of output bits that may flip, defined as

$$D_{av}(F, \Delta) = b - \sum_{i} \delta(p_i)$$

with $\delta(x)$ equal to 1 if $x = 0$ and 0 otherwise.

Metric

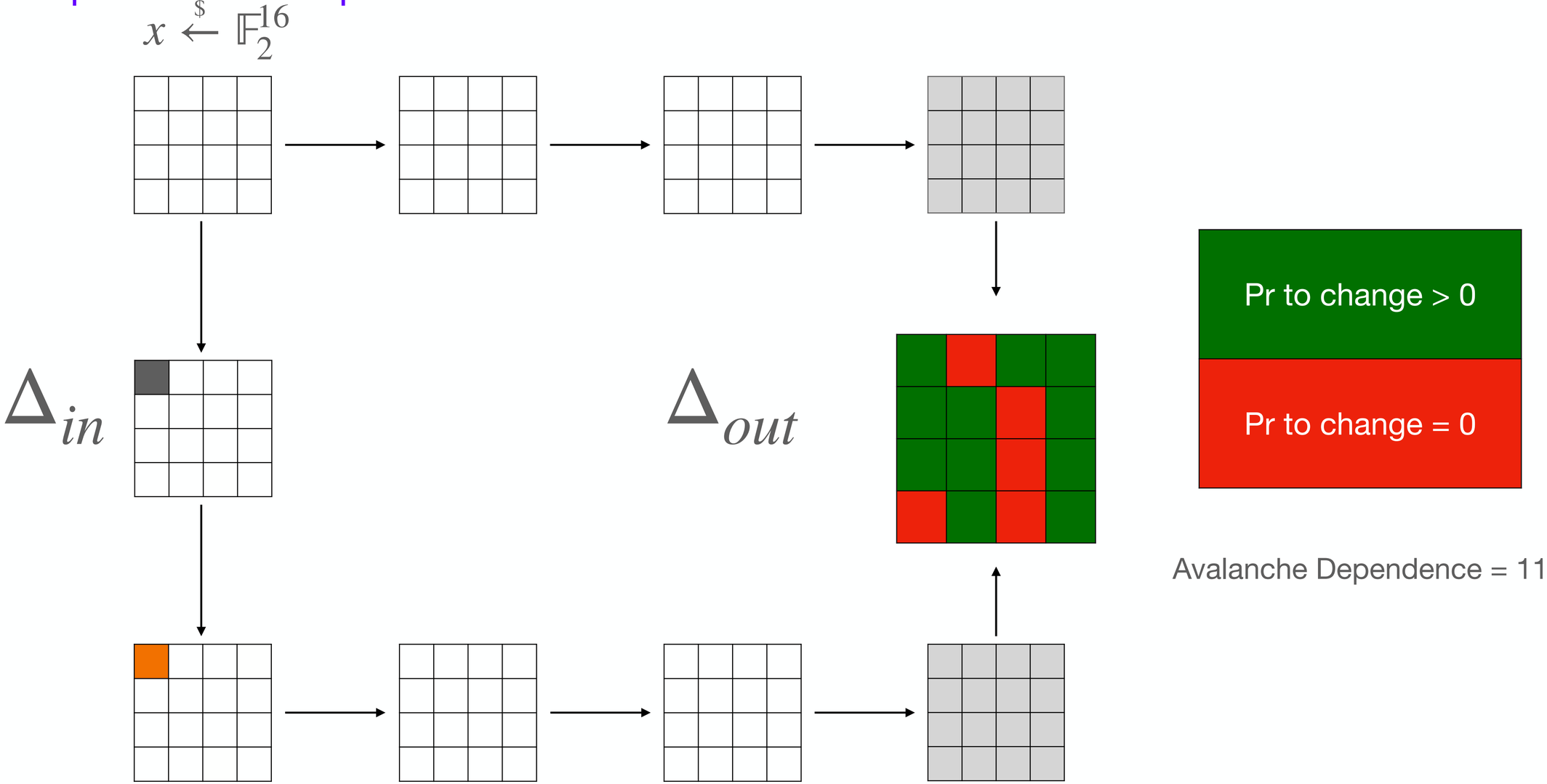
Avalanche Dependence for Input Bit 0



Avalanche Dependence = 11

Metric

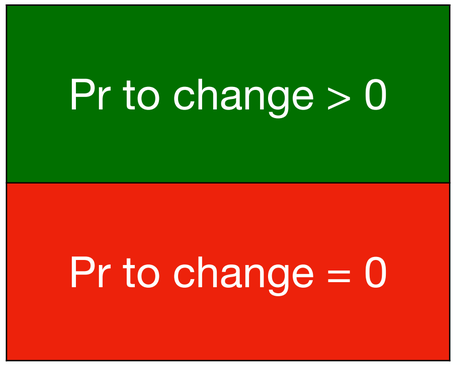
Avalanche Dependence for Input Bit 0



Avalanche Dependence = 11

Metric

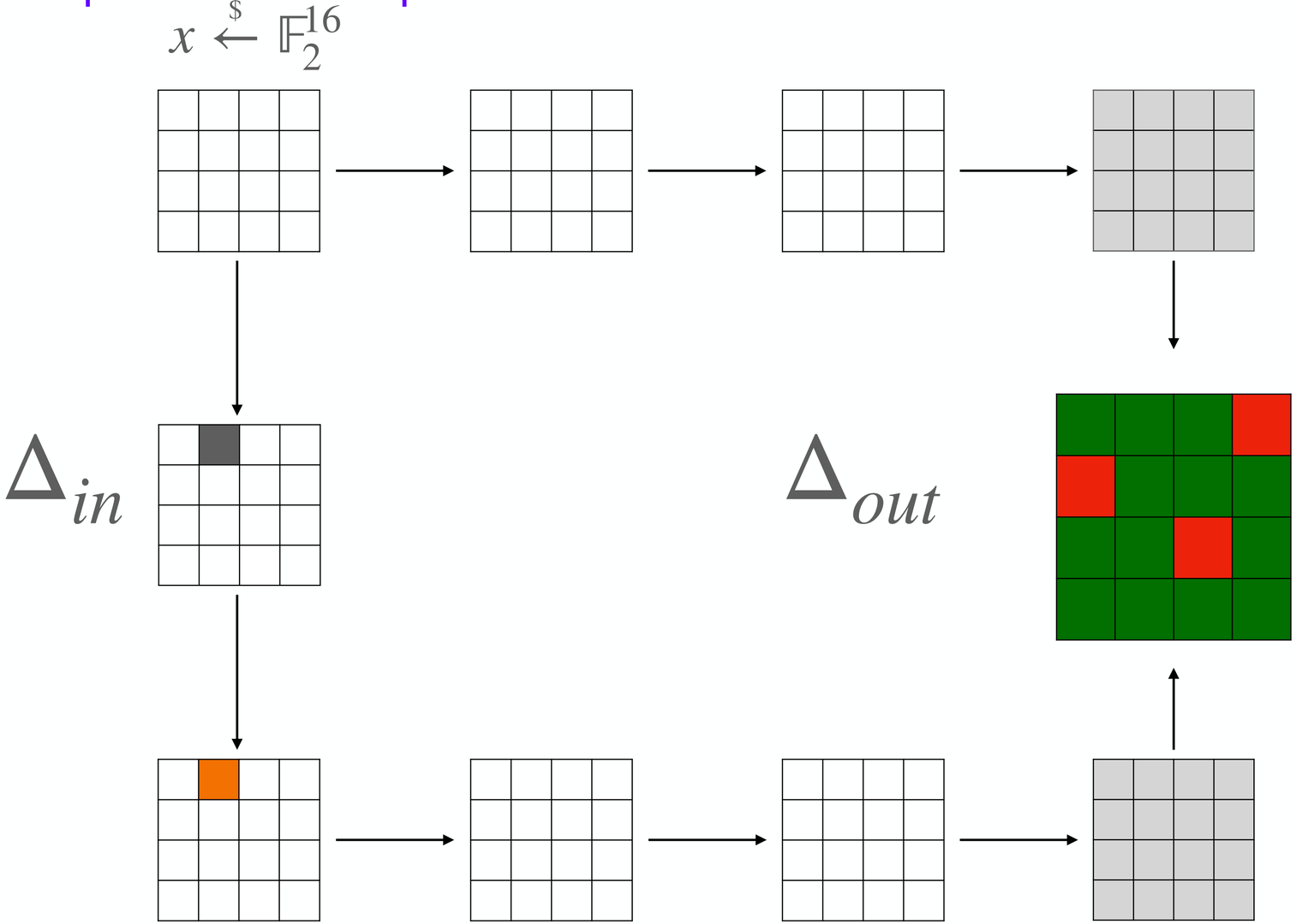
Avalanche Dependence for Input Bit 1



Avalanche Dependence = 13

Metric

Avalanche Dependence for Input Bit 1



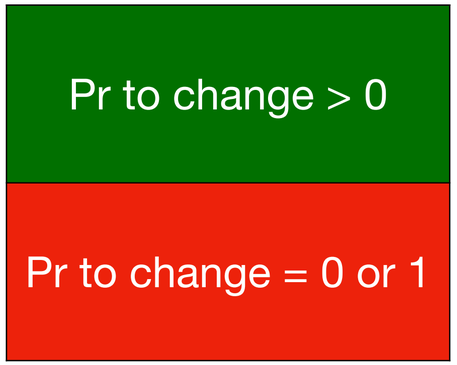
Pr to change > 0

Pr to change = 0

Avalanche Dependence = 13

Metric

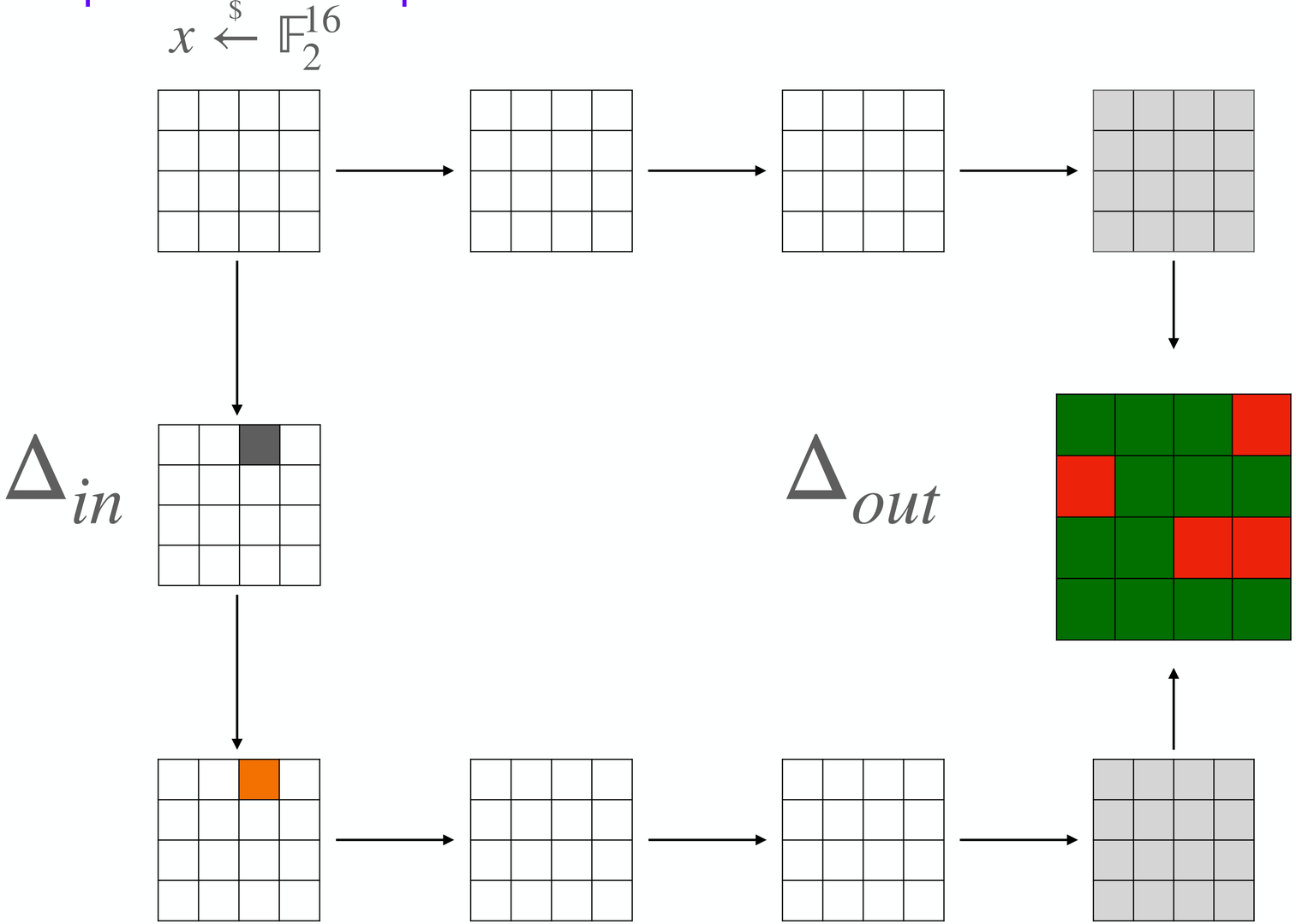
Avalanche Dependence for Input Bit 2



Avalanche Dependence = 12

Metric

Avalanche Dependence for Input Bit 2



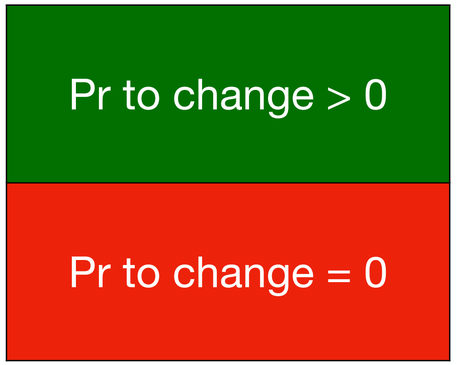
Pr to change > 0

Pr to change = 0 or 1

Avalanche Dependence = 12

Metric

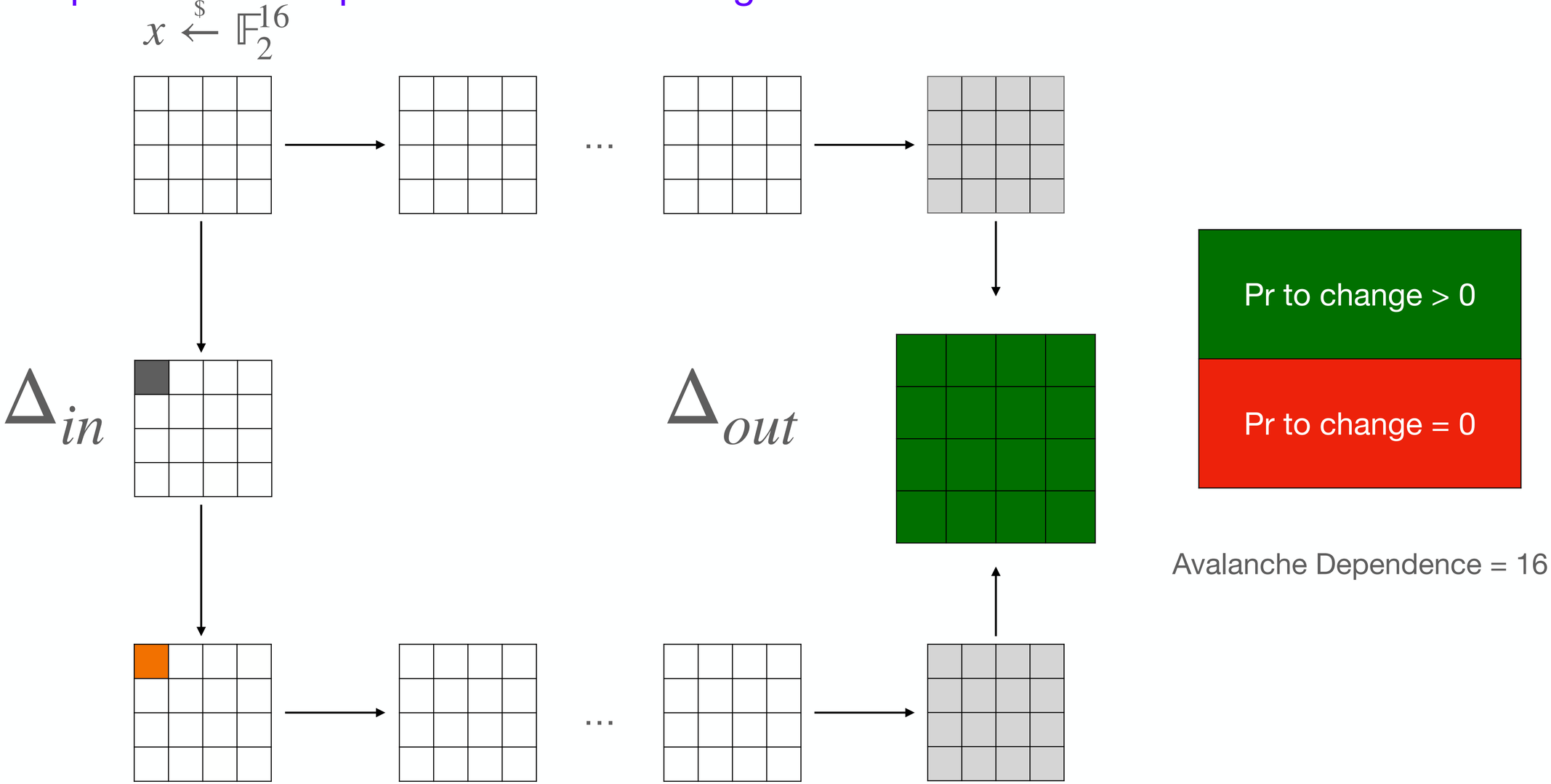
Avalanche Dependence for Input Bit 0 after increasing the number of rounds



Avalanche Dependence = 16

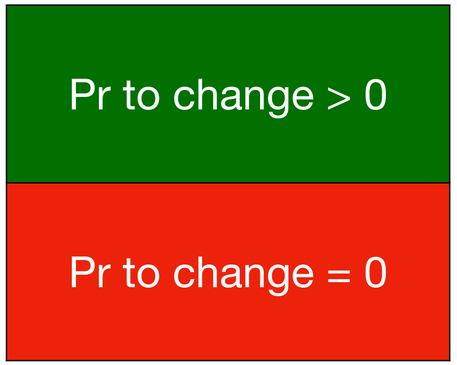
Metric

Avalanche Dependence for Input Bit 0 after increasing the number of rounds



Metric

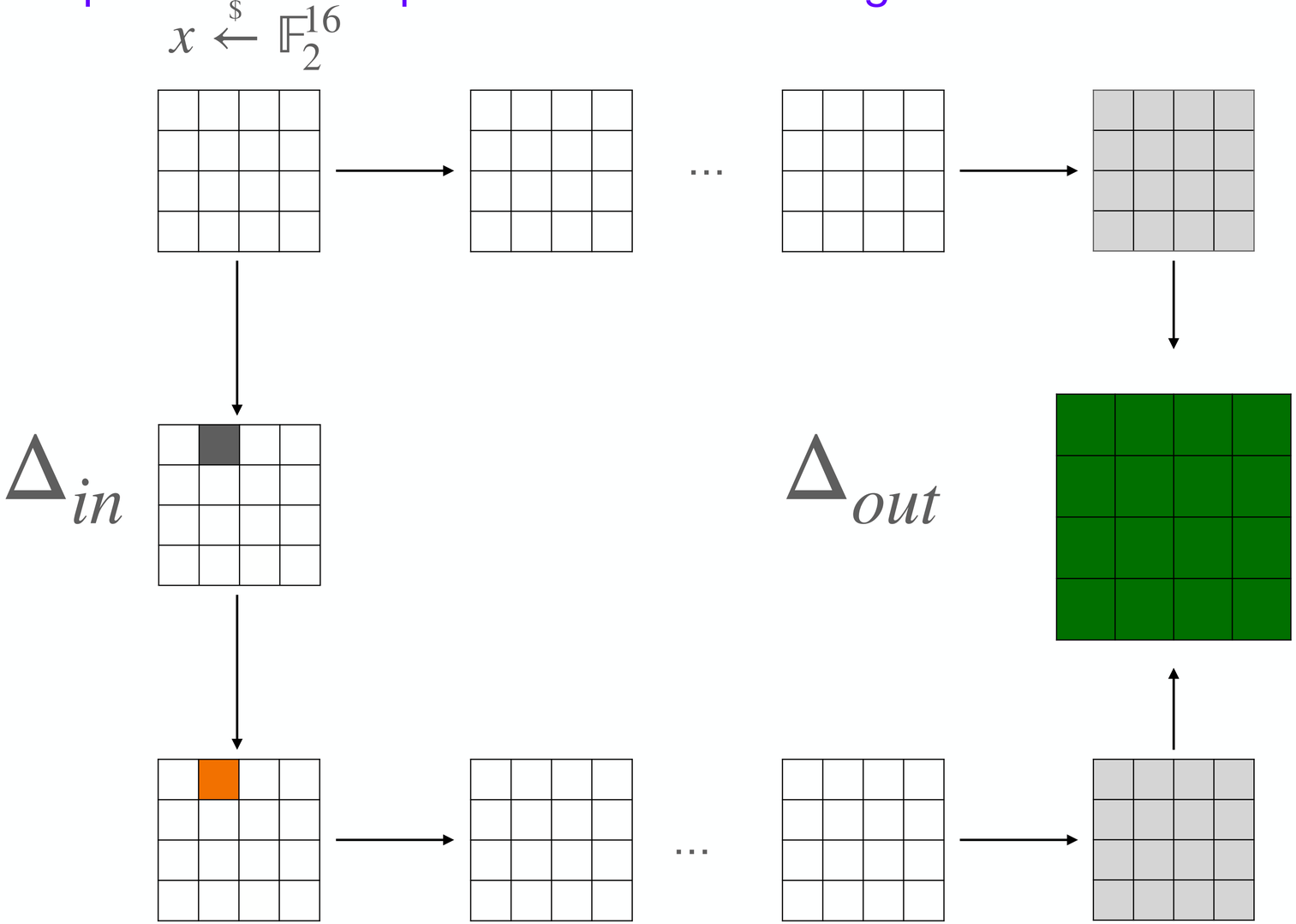
Avalanche Dependence for Input Bit 1 after increasing the number of rounds



Avalanche Dependence = 16

Metric

Avalanche Dependence for Input Bit 1 after increasing the number of rounds



Pr to change > 0

Pr to change = 0

Avalanche Dependence = 16

Full Diffusion Criterion

Full-Diffusion

If the number of output bits that may flip for every input single bit difference is the size of the cipher

The full-diffusion criterion is satisfied if $D_{av}(F, \Delta) = b$ for all Δ with Hamming weight 1.

Other two metrics [Webster85]

Other two metrics [Webster85]

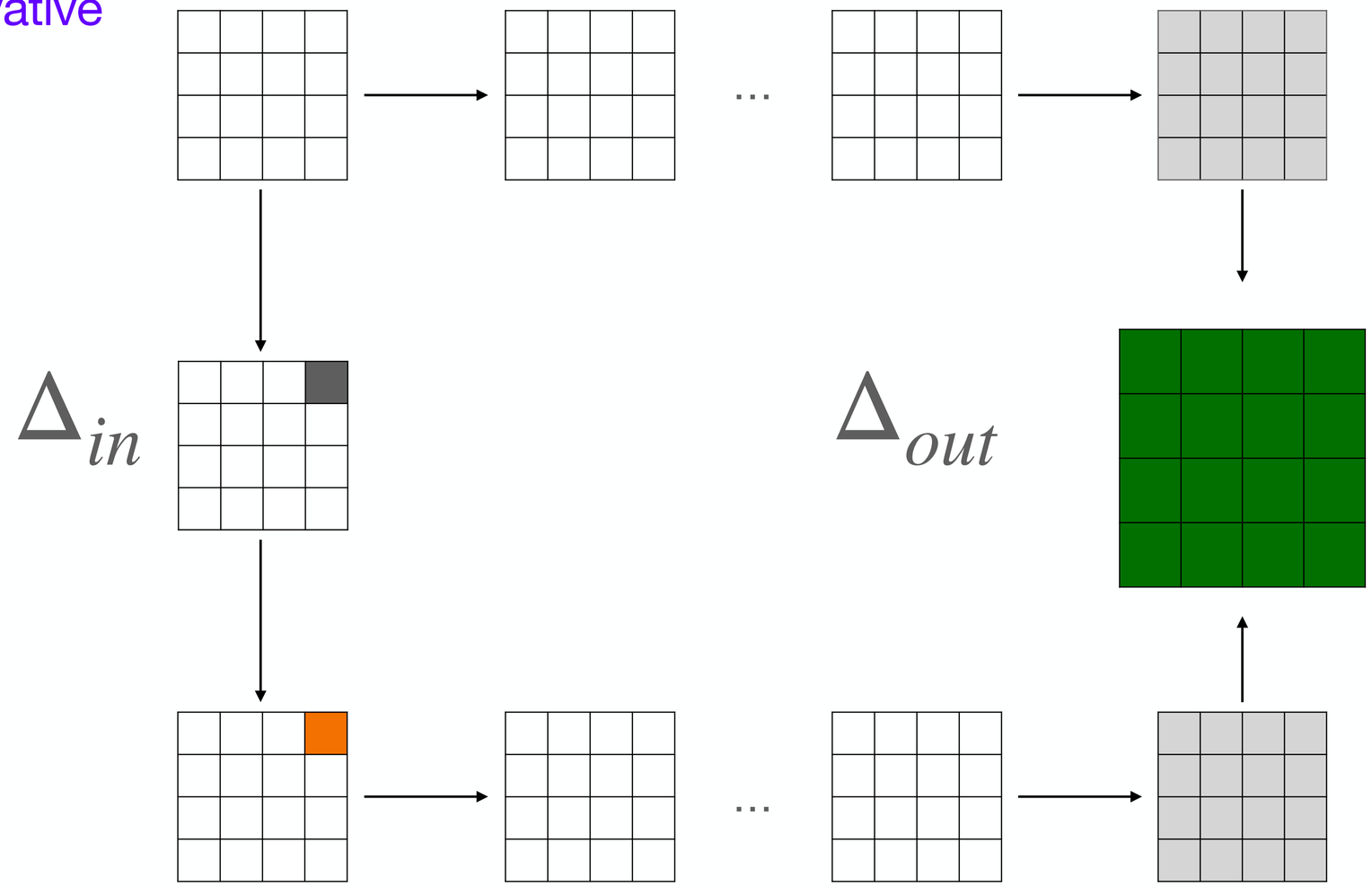
- Avalanche Weight -> Avalanche Criterion

Other two metrics [Webster85]

- Avalanche Weight -> Avalanche Criterion
- Avalanche Entropy -> Strict Avalanche Criterion

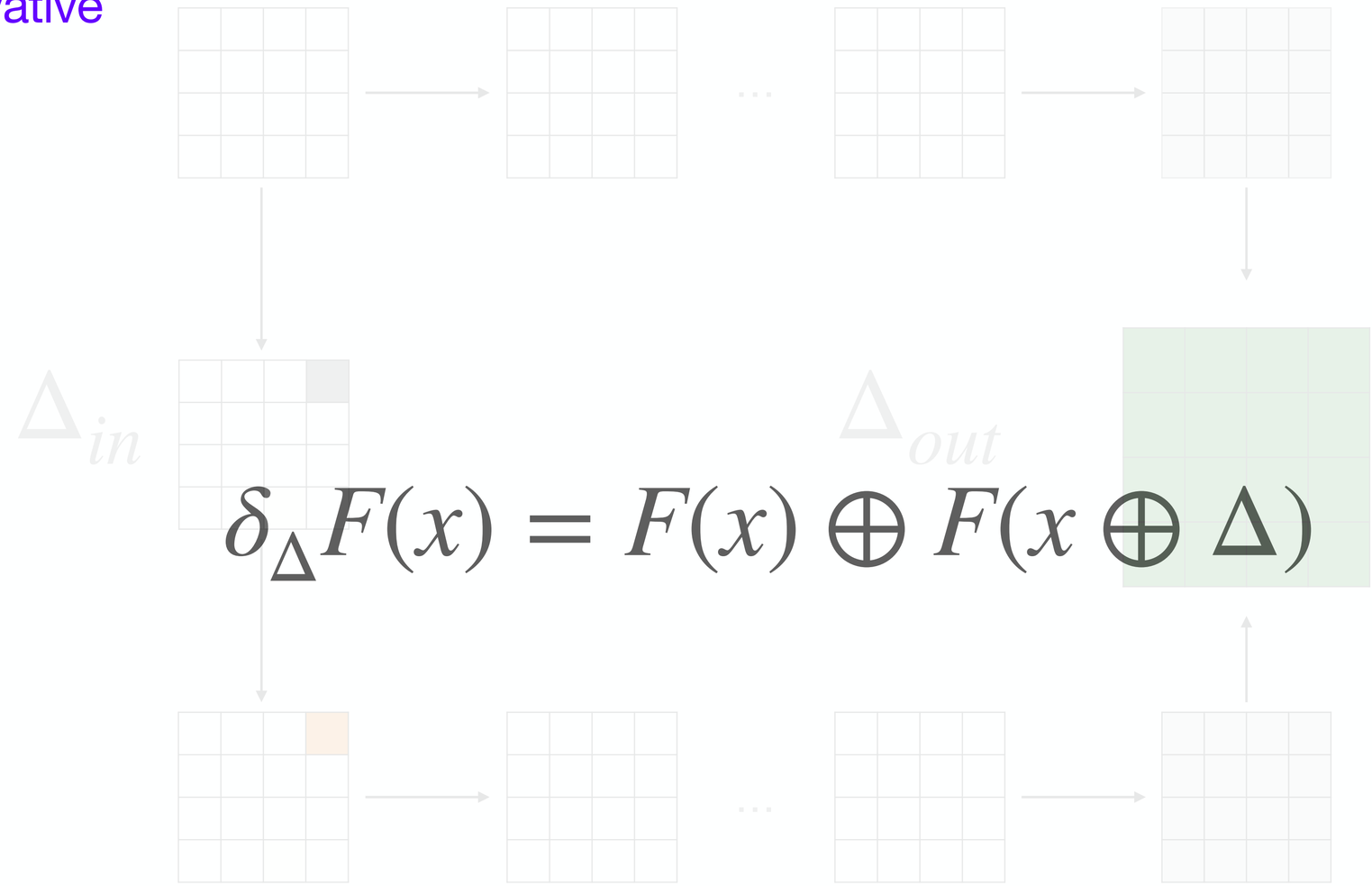
The process to compute the metrics involve first-order derivative

First-order derivative



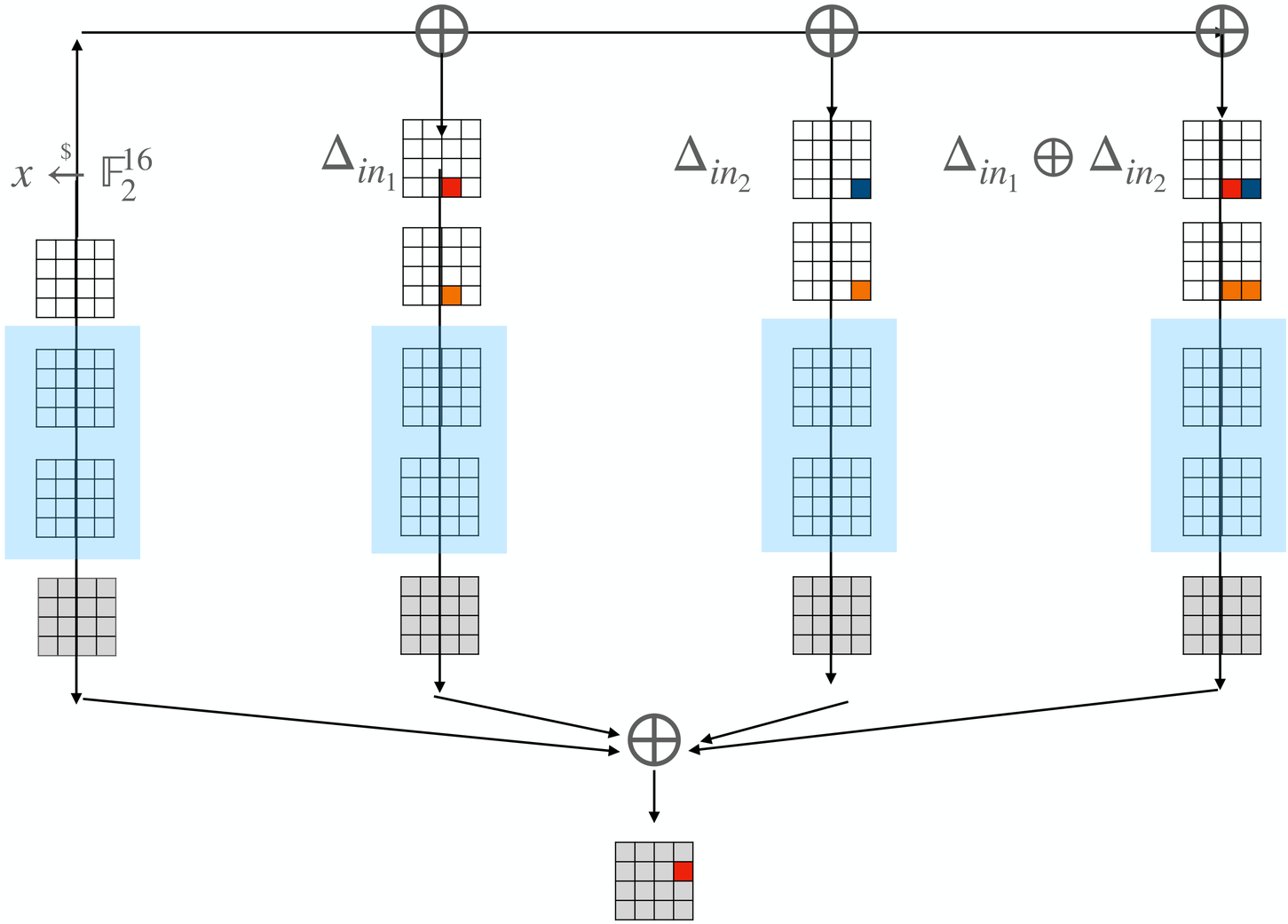
The process to compute the metrics involve first-order derivative

First-order derivative



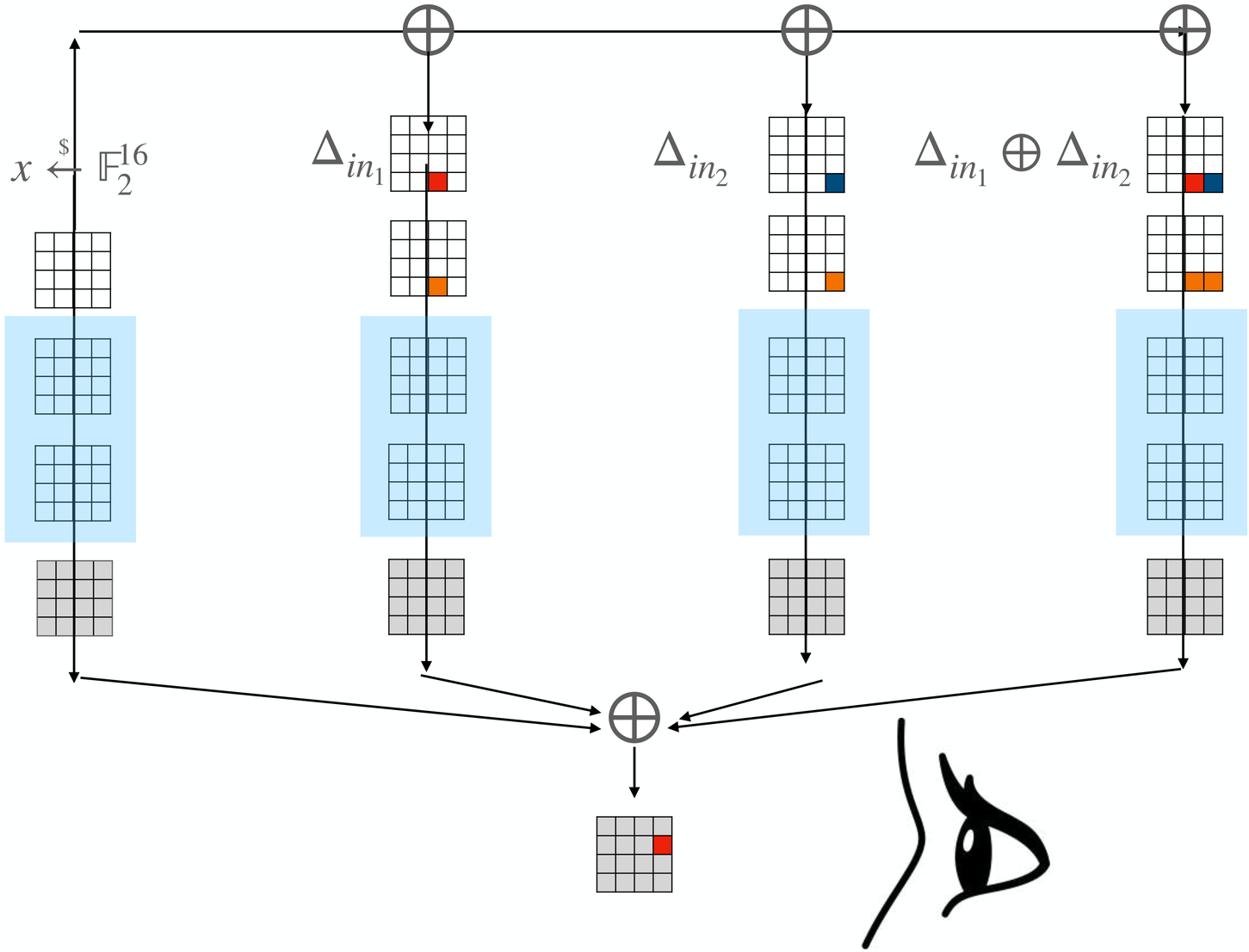
High-order differential distinguisher

Truncated Differential Distinguisher of a 2-round symmetric primitive



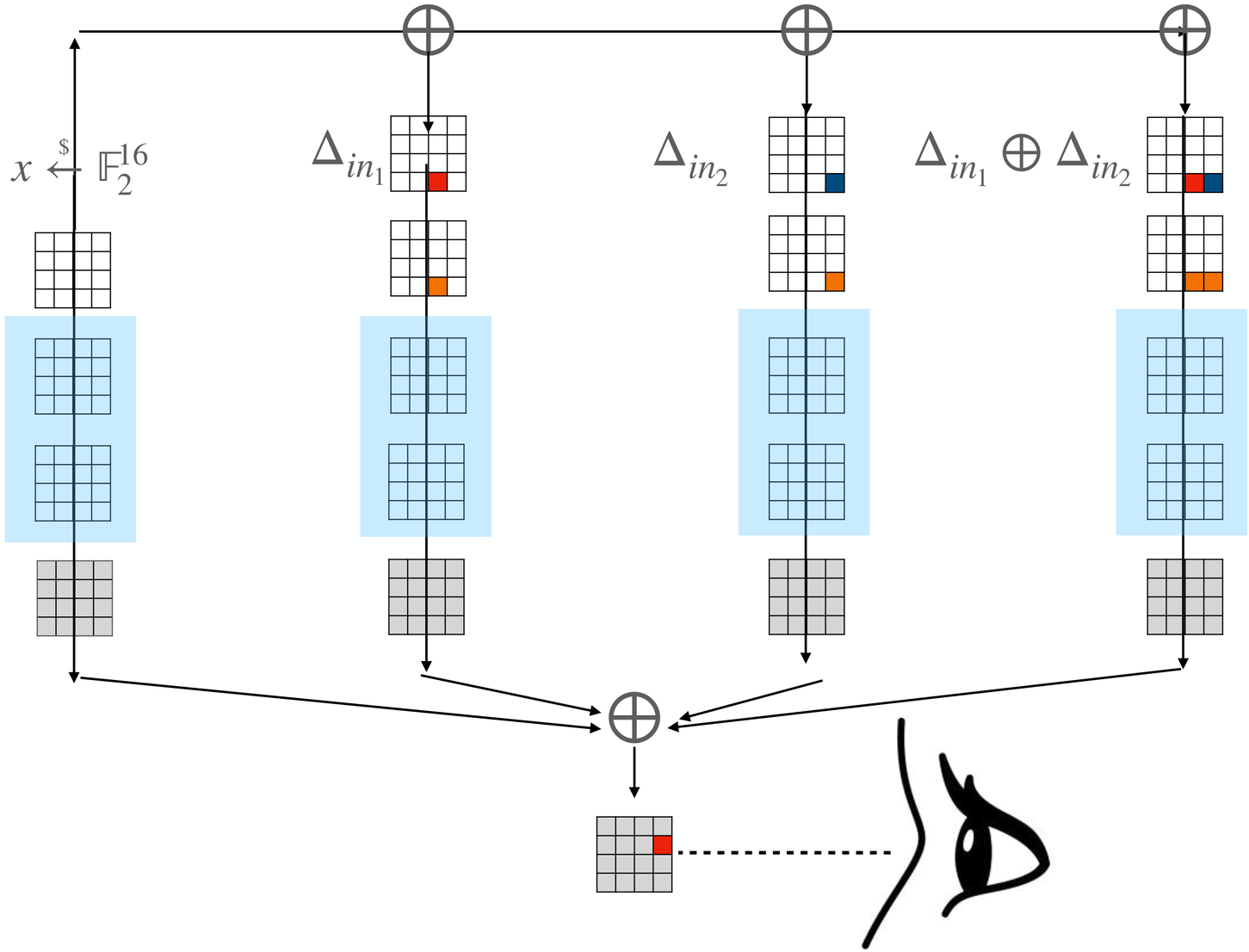
High-order differential distinguisher

Truncated Differential Distinguisher of a 2-round symmetric primitive



High-order differential distinguisher

Truncated Differential Distinguisher of a 2-round symmetric primitive

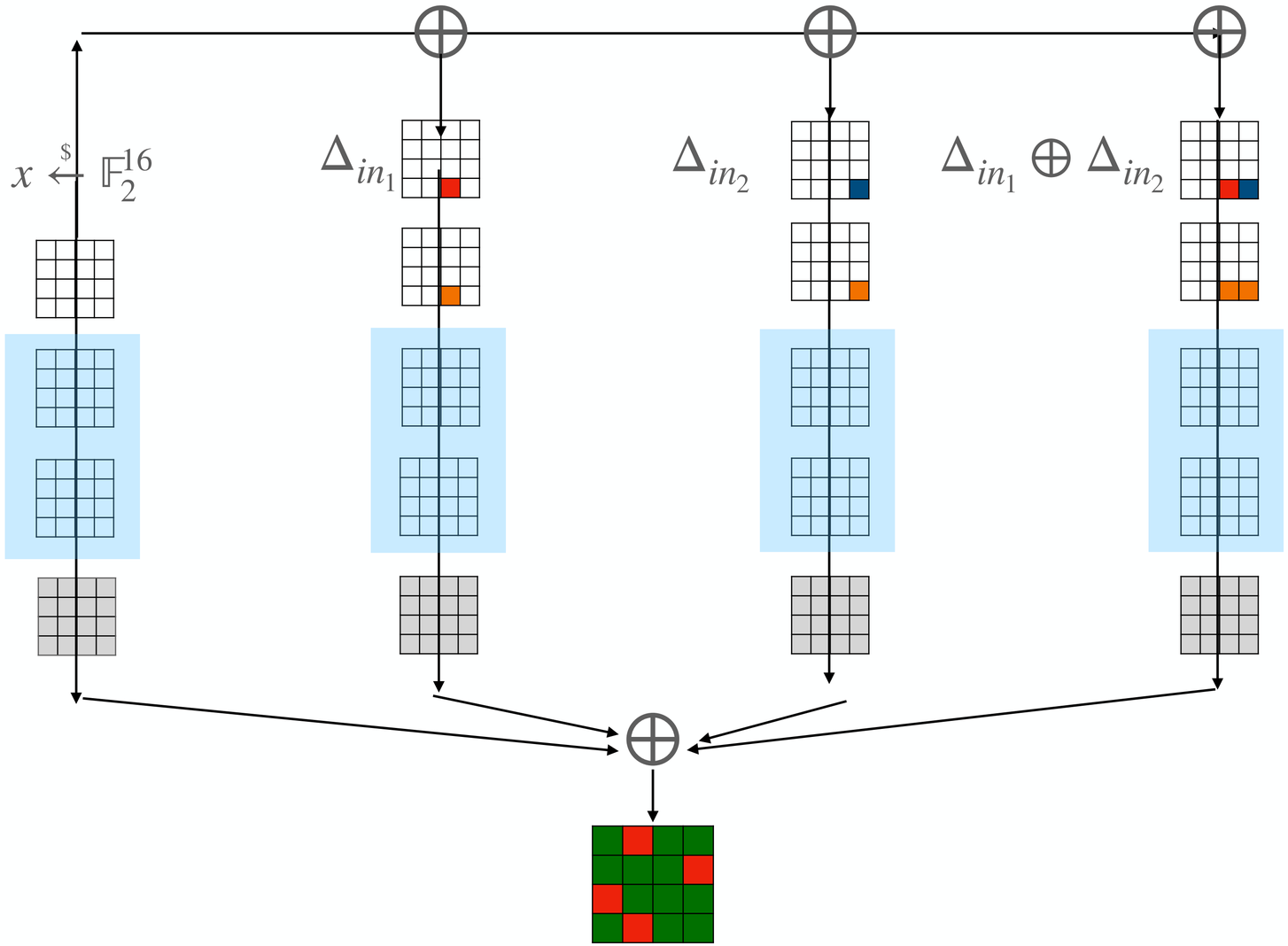


High-order differential distinguisher

Truncated Differential Distinguisher of a 2-round symmetric primitive

$$\delta^{(2)}_{\Delta_{in_1},\Delta_{in_2}} F(x) = F(x) \oplus F\left(x \oplus \Delta_{in_1}\right) \oplus F\left(x \oplus \Delta_{in_2}\right) \oplus F\left(x \oplus \Delta_{in_1} \oplus \Delta_{in_2}\right)$$

Avalanche Dependence for the second order case

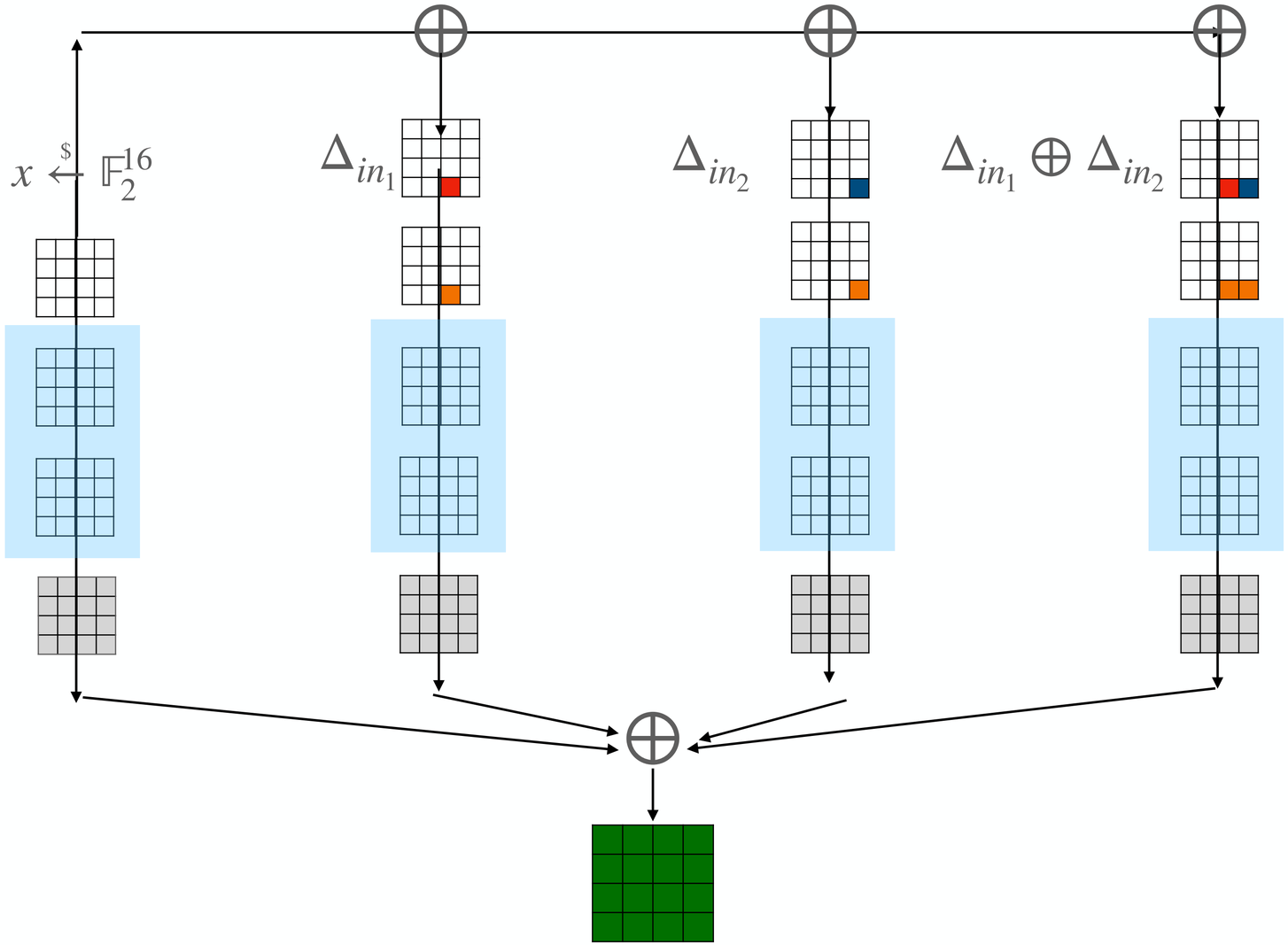


Pr to change > 0

Pr to change = 0 or 1

Avalanche Dependence = 12

Full-diffusion criterion for the second order case

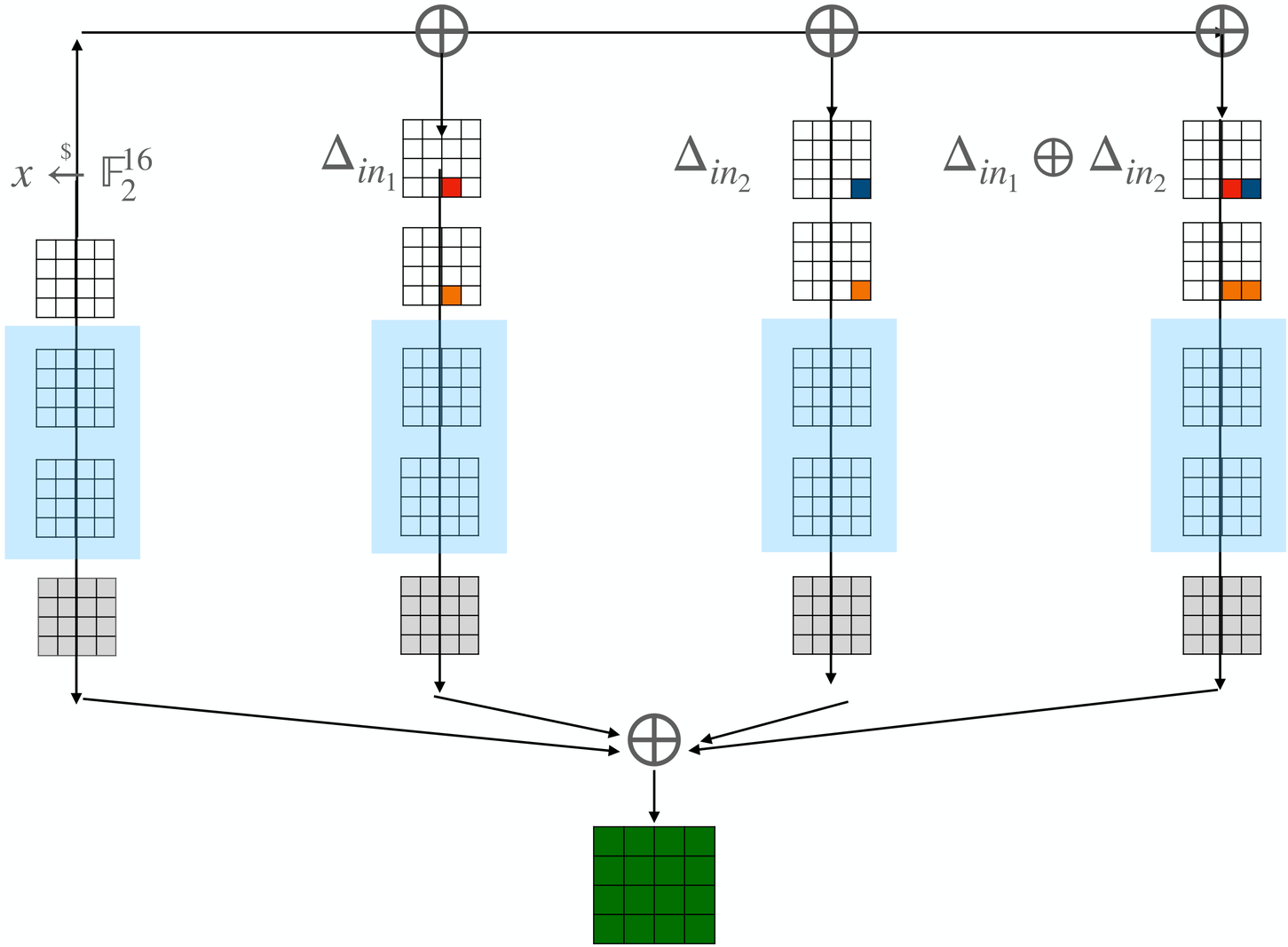


Pr to change > 0

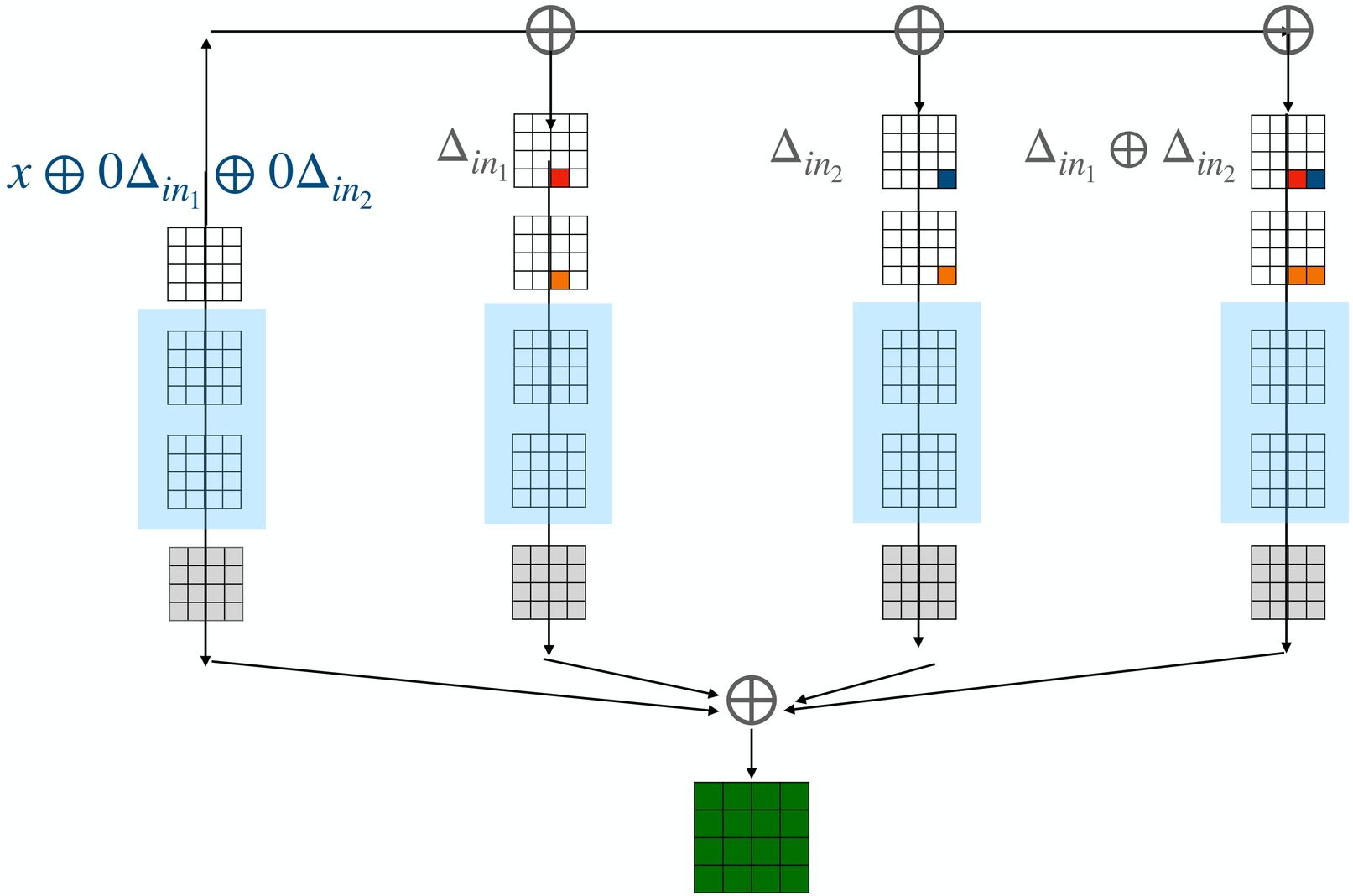
Pr to change = 0 or 1

Avalanche Dependence = 16

Full-diffusion criterion for the second order case



Full-diffusion criterion for the second order case



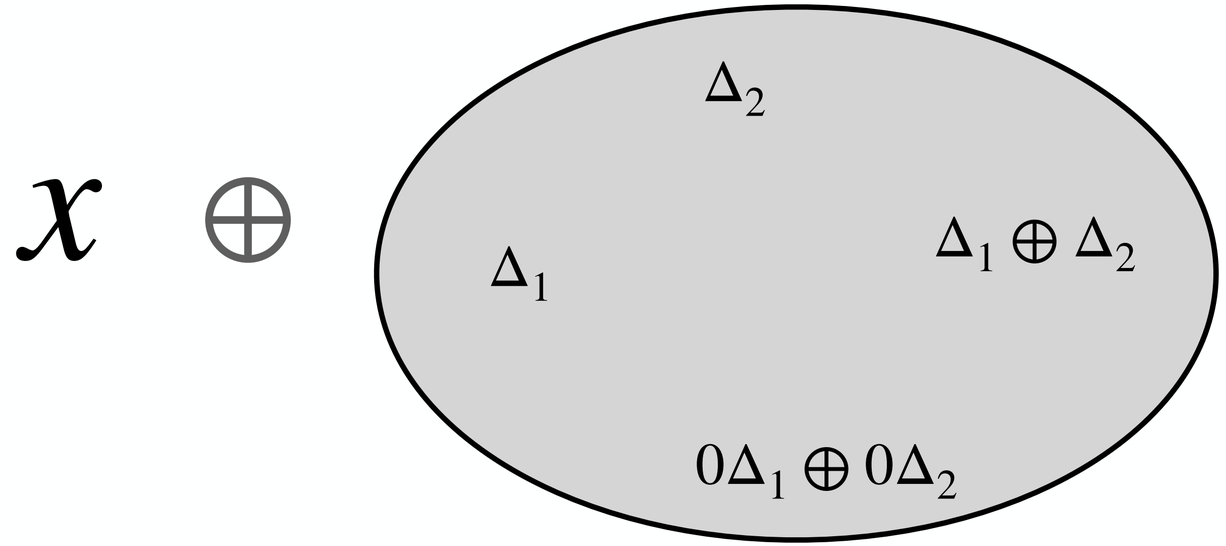
Pr to change > 0

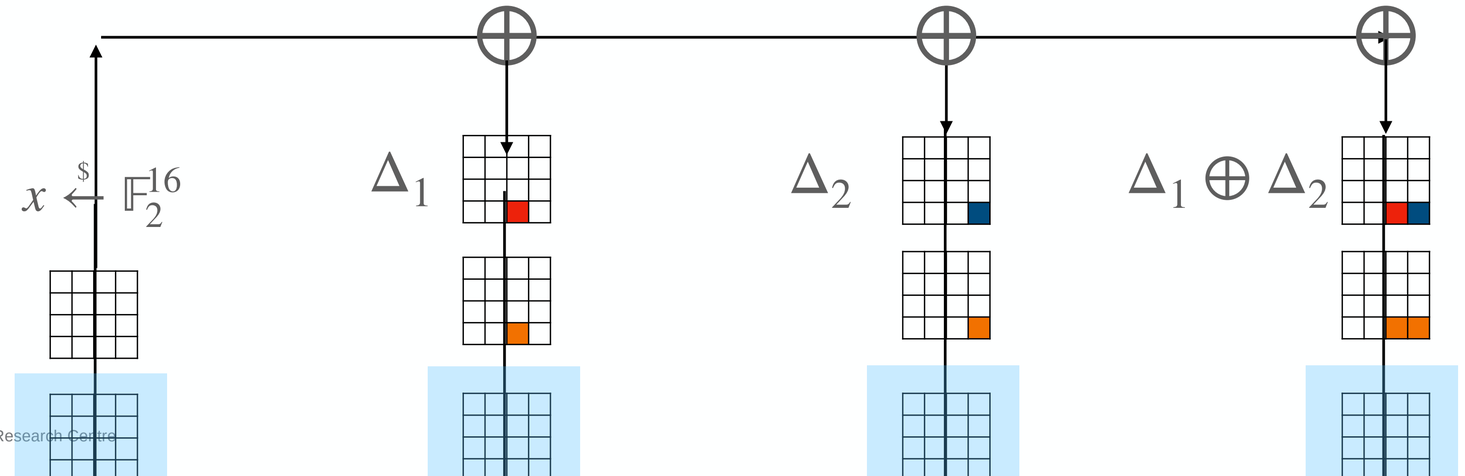
Pr to change = 0 or 1

Avalanche Dependence = 16

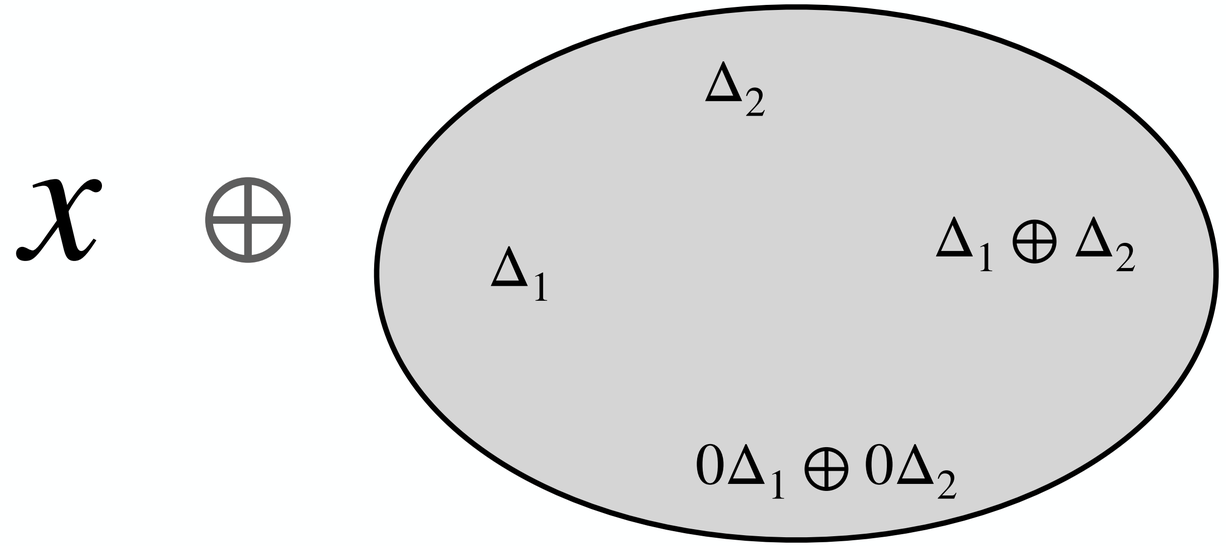
Full-diffusion criterion for the second order case

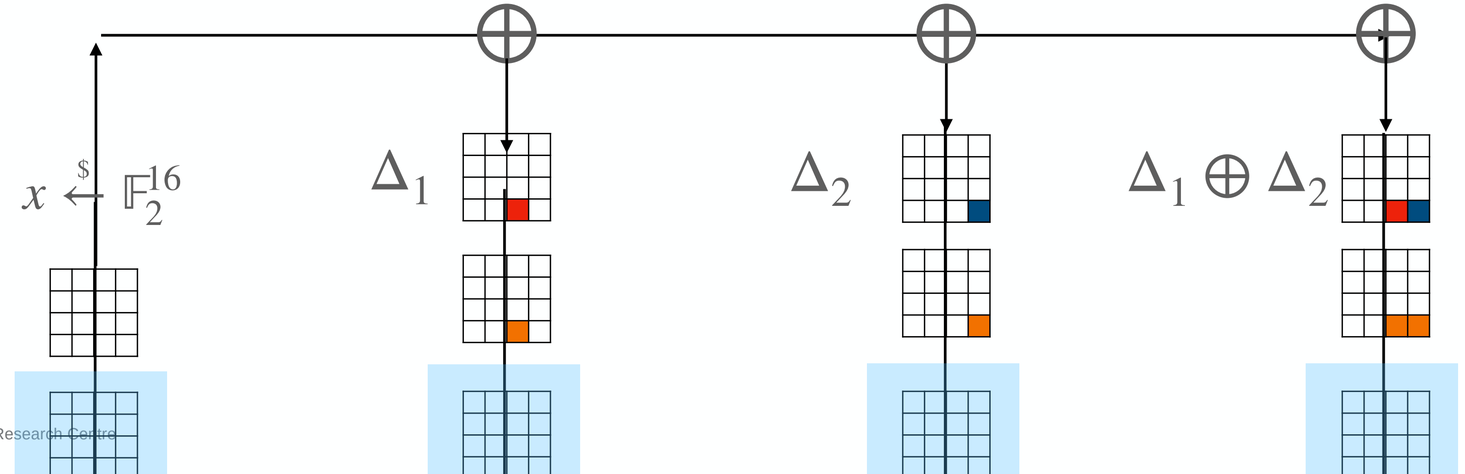
Avalanche metric for the second order case



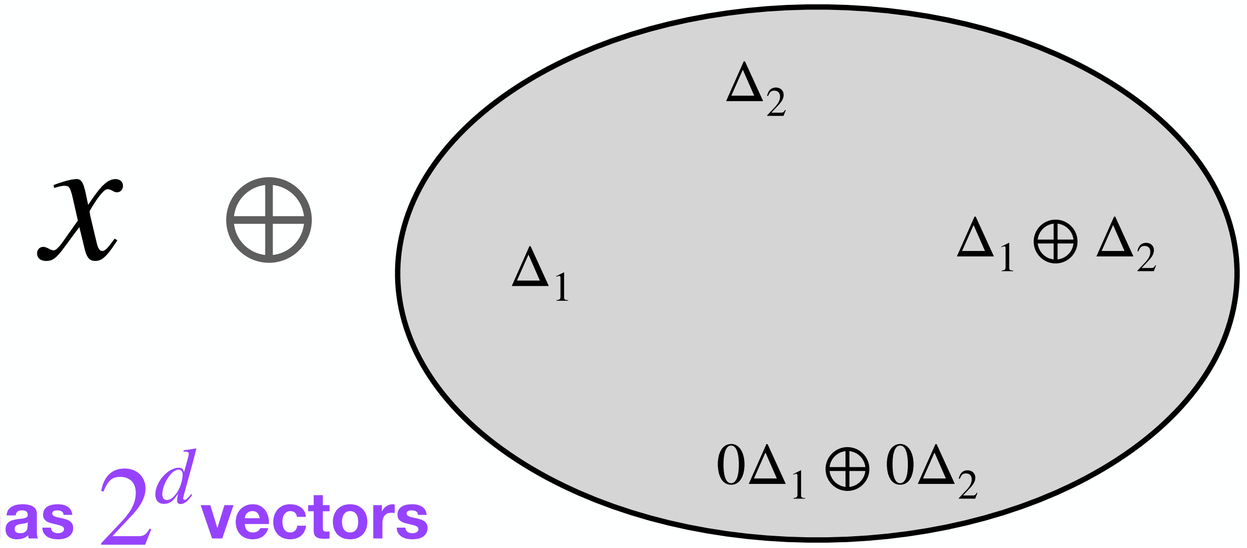


Avalanche metric for the second order case



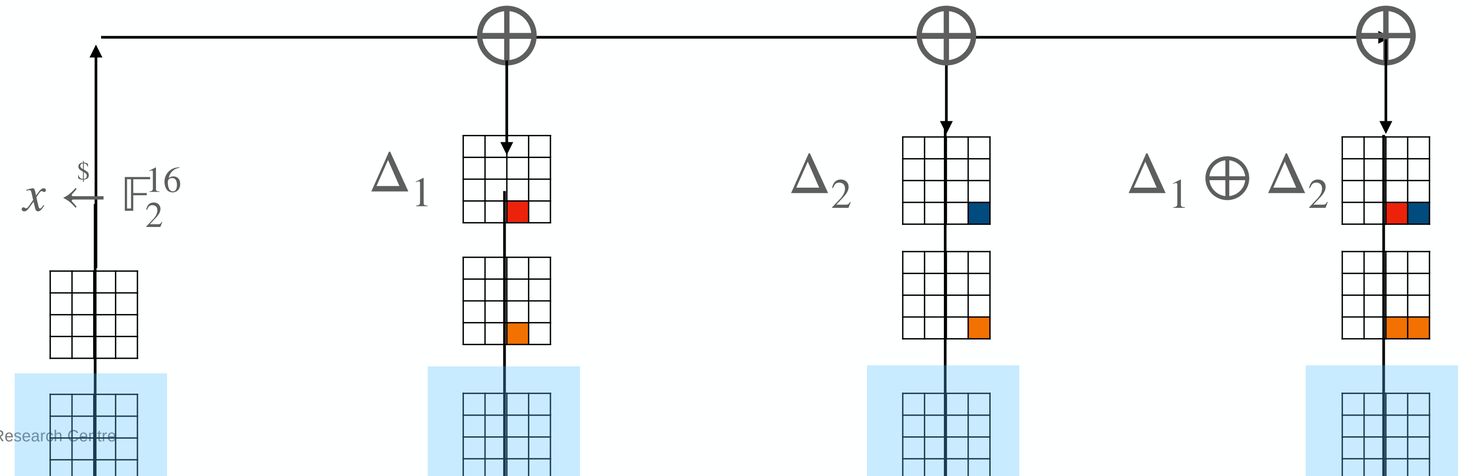


Avalanche metric for the second order case



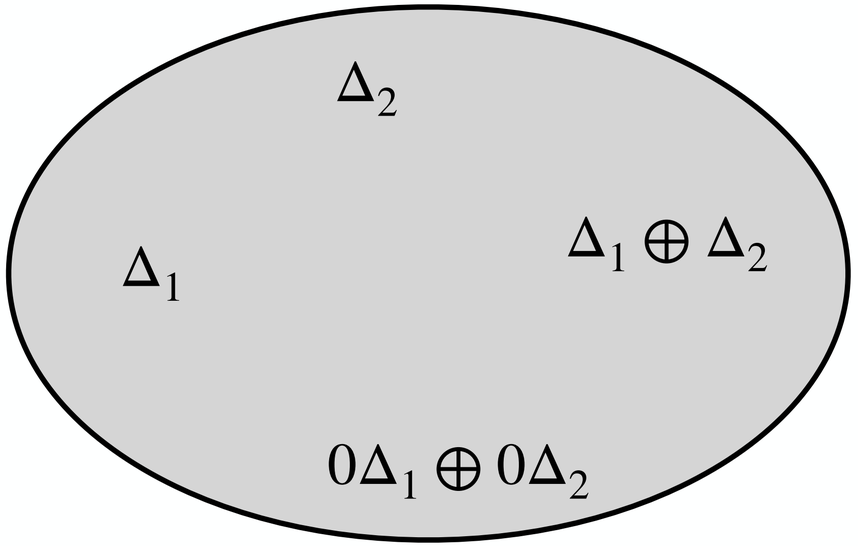
Each vector space has 2^d vectors

$span\{\Delta_1, \Delta_2\}$



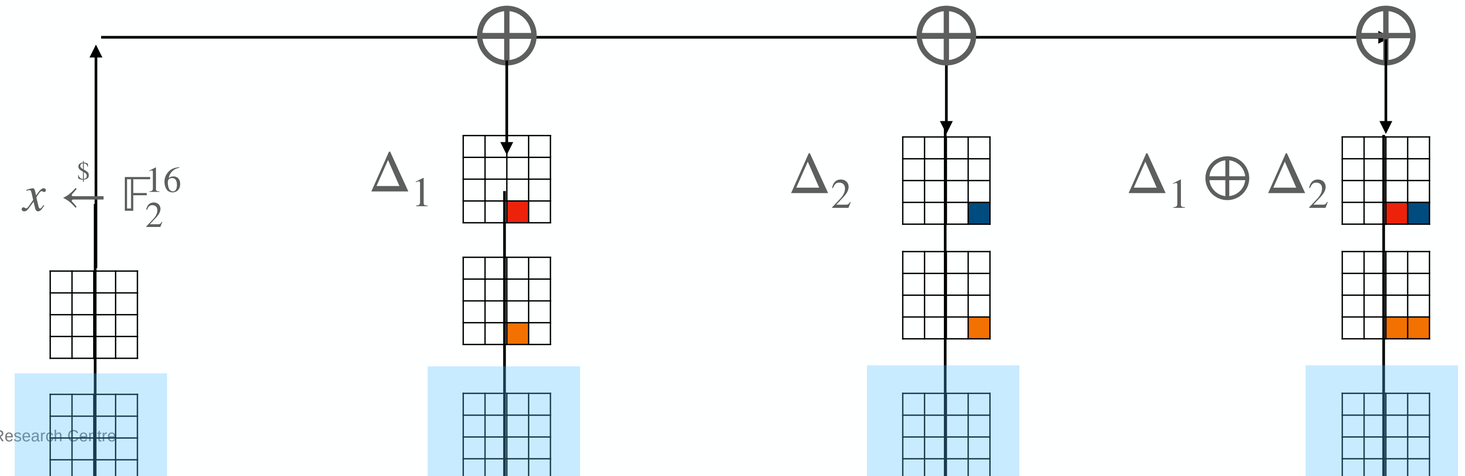
Avalanche metric for the second order case

$x \oplus$

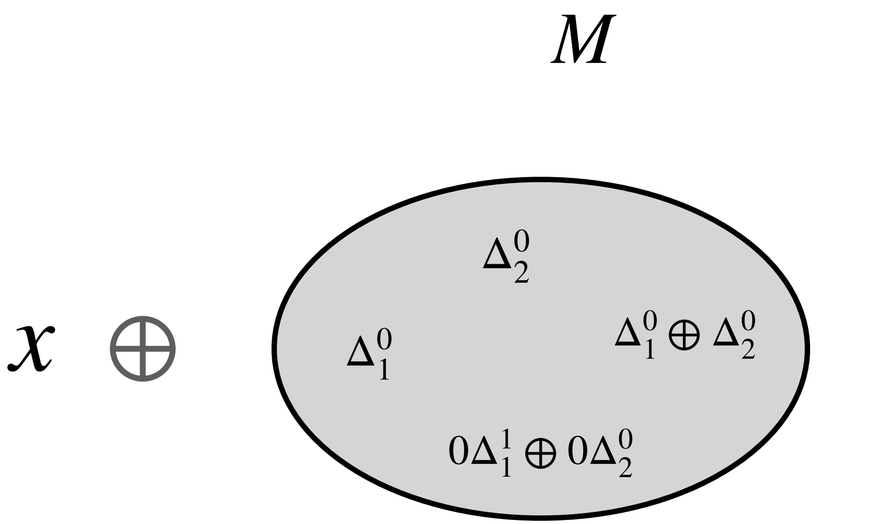


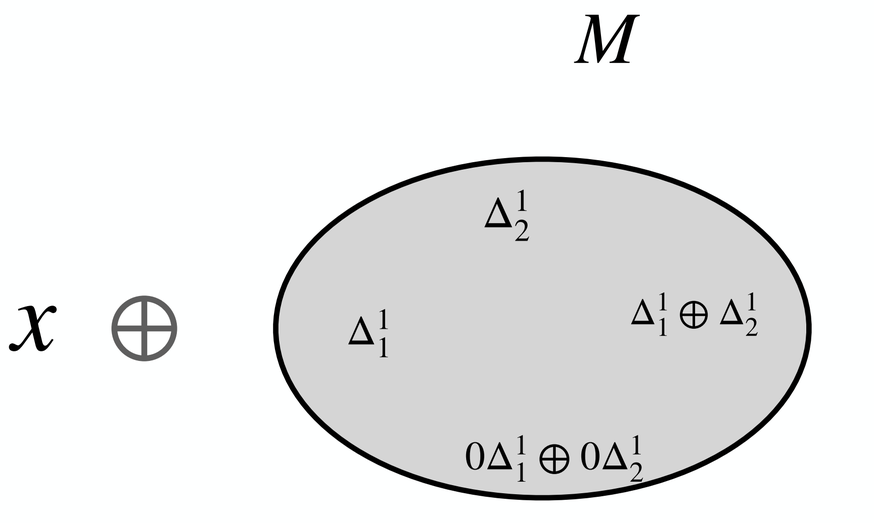
Repeat the process M times to obtain value of the metric

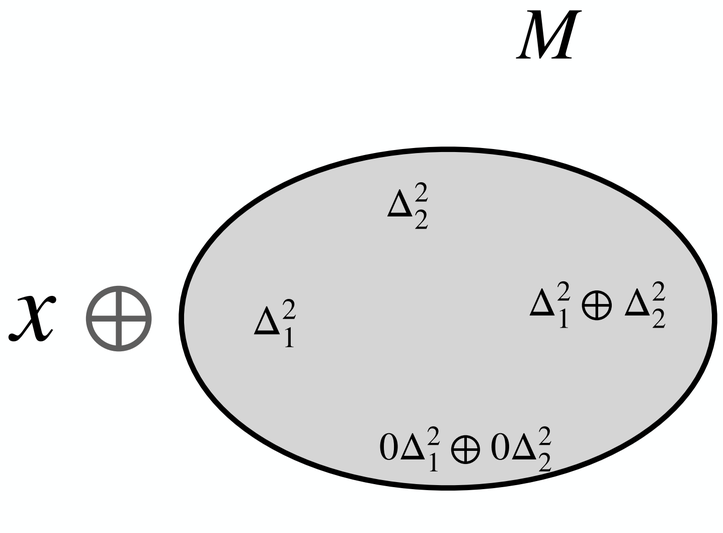
$span\{\Delta_1, \Delta_2\}$

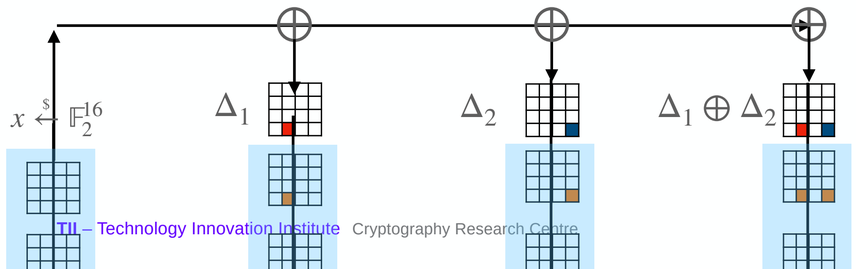


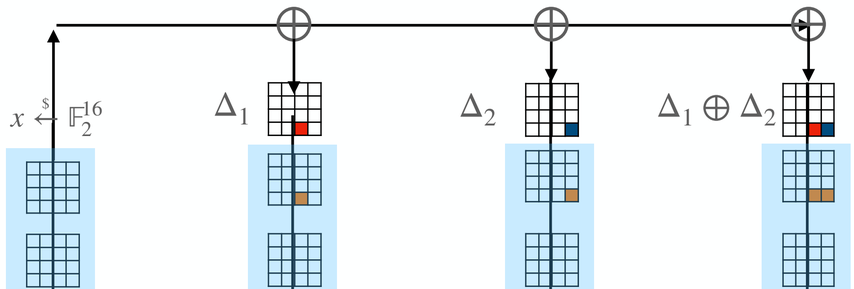
Full-diffusion criterion for the second order case

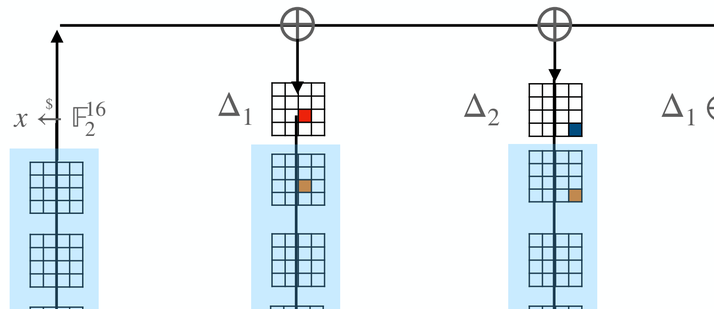




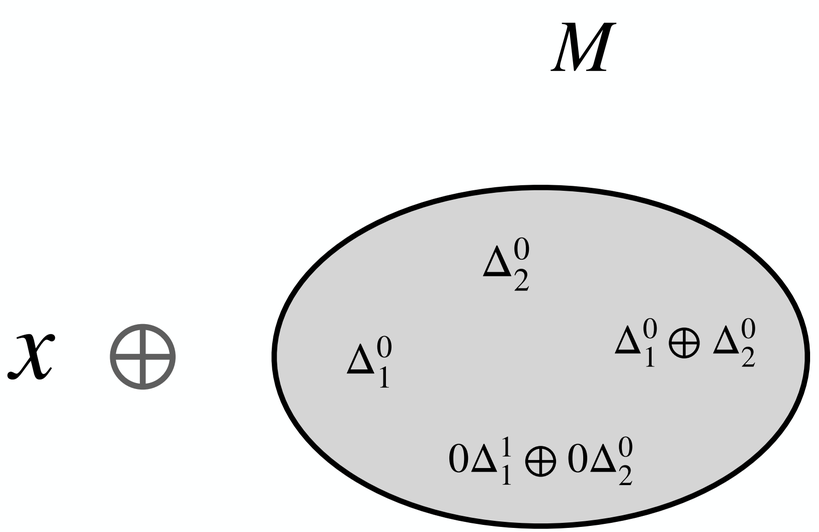




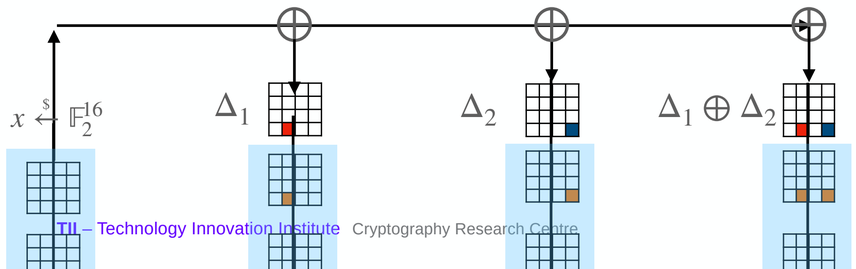


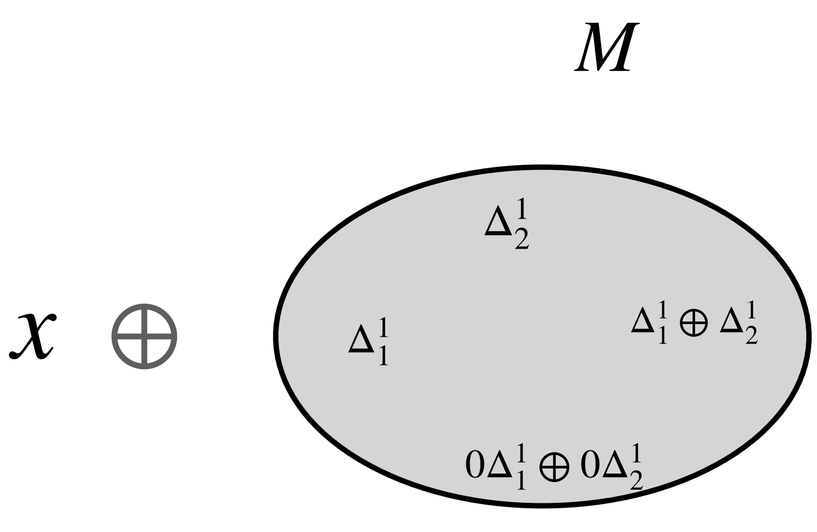


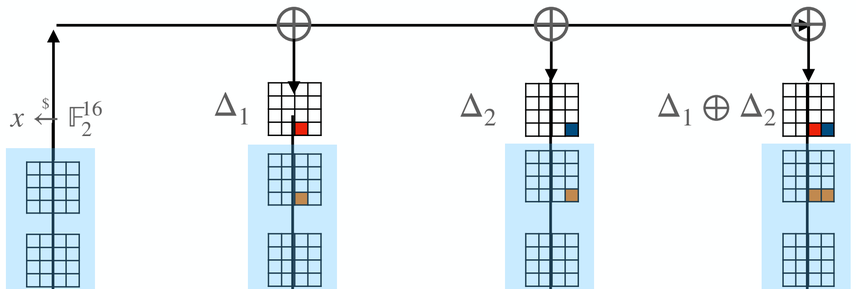
Full-diffusion criterion for the second order case

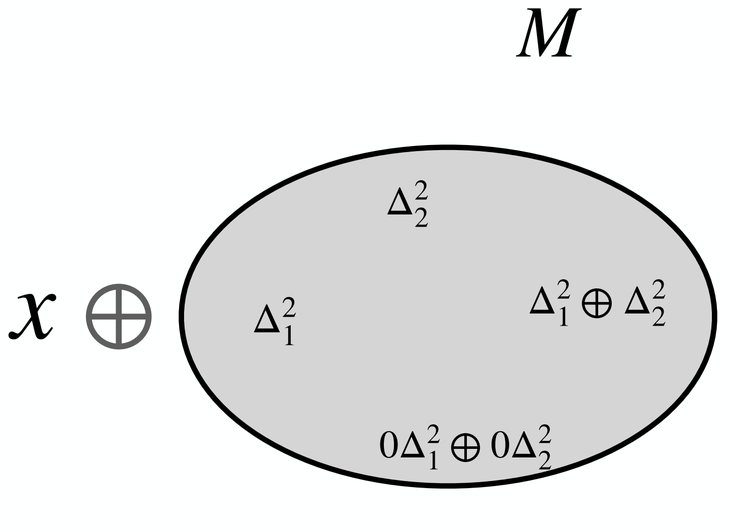


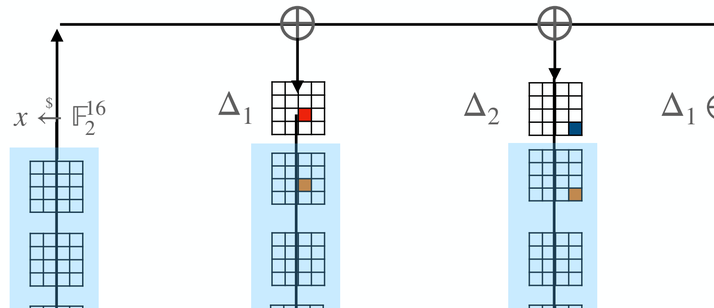
$span\{\Delta_1^0, \Delta_2^0\}$



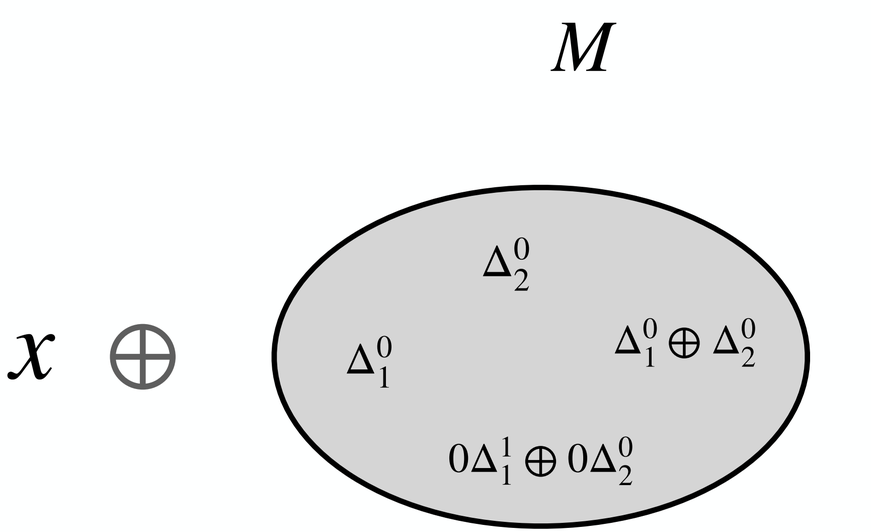




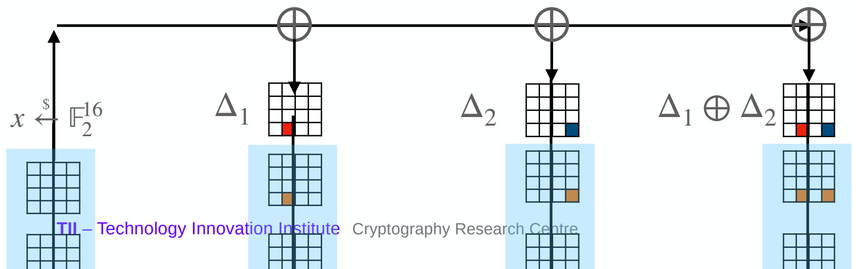


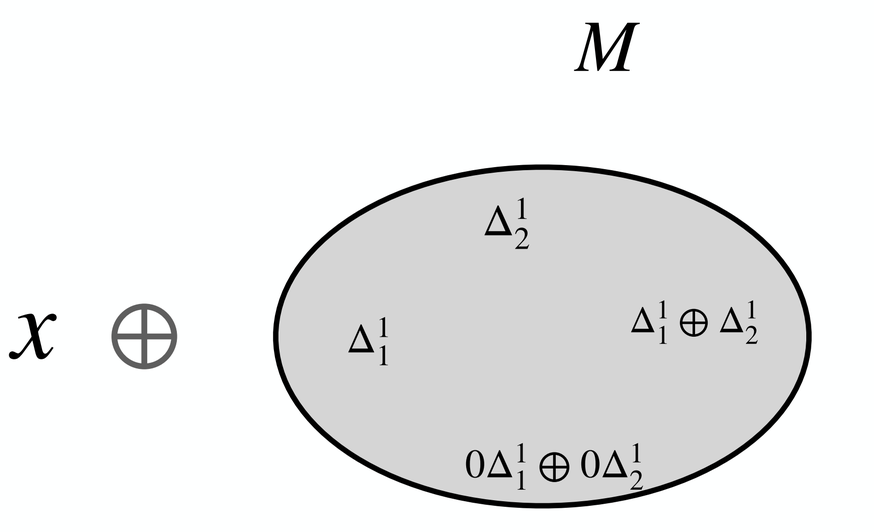


Full-diffusion criterion for the second order case

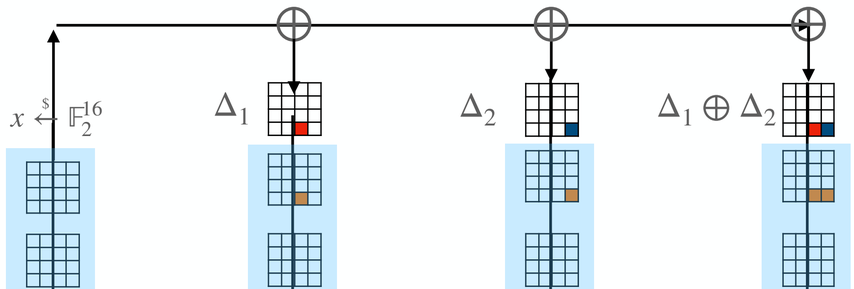


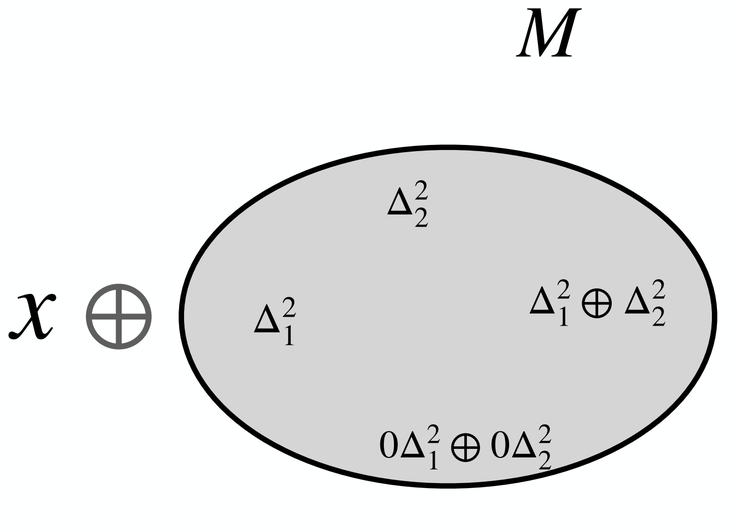
$span\{\Delta_1^0, \Delta_2^0\}$

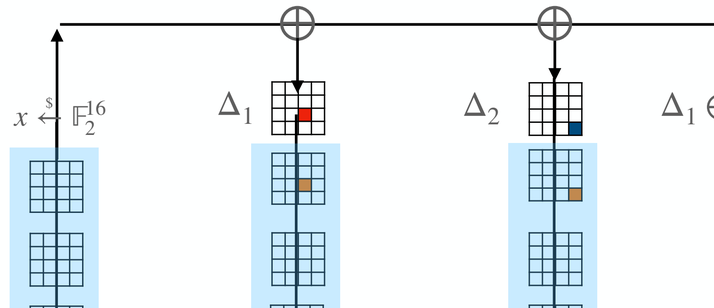




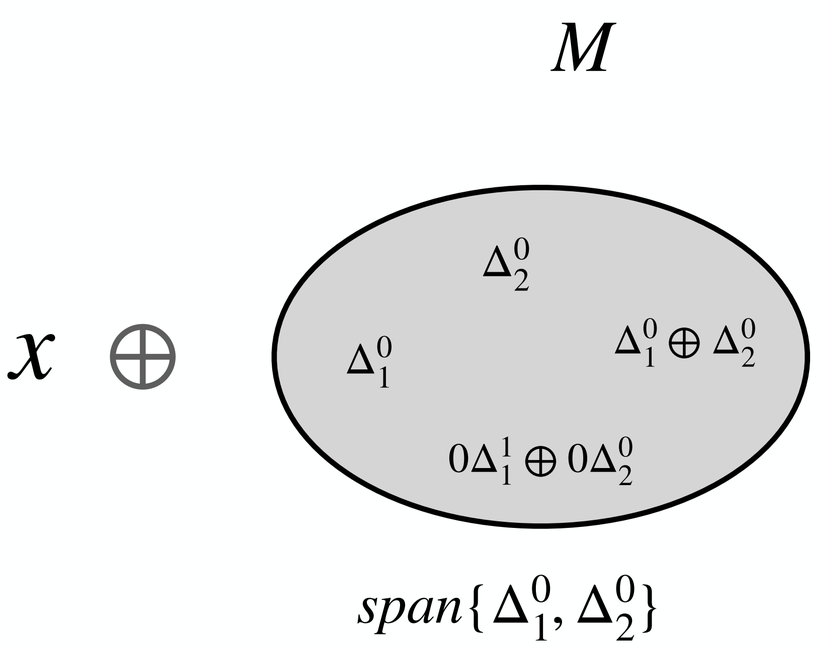
$span\{\Delta_1^1, \Delta_2^1\}$

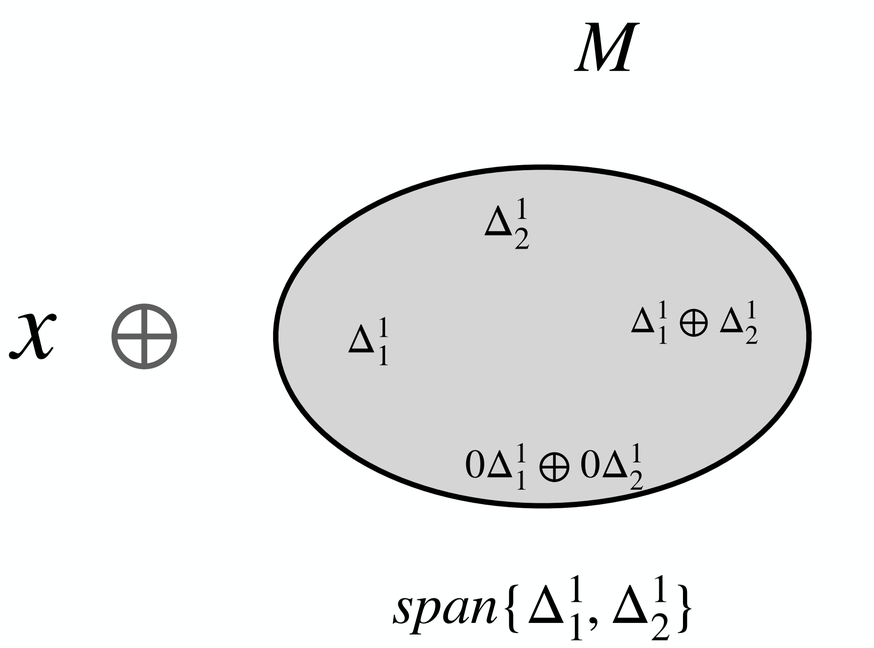


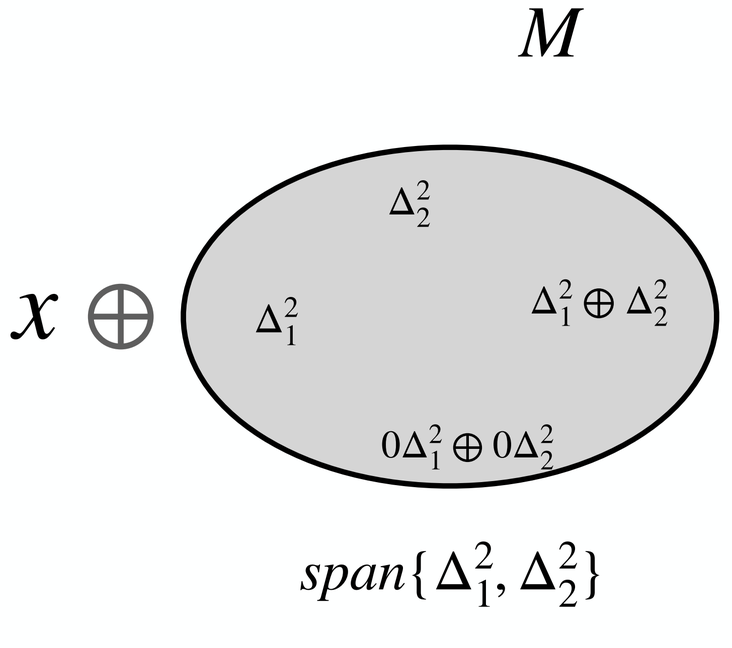


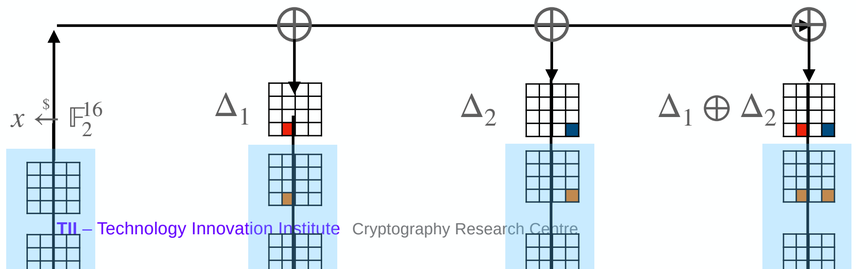


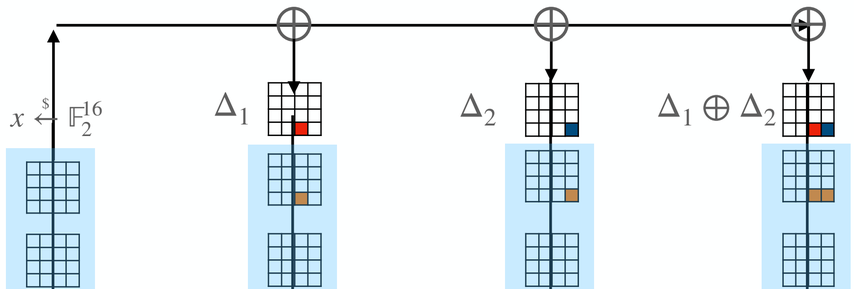
Full-diffusion criterion for the second order case

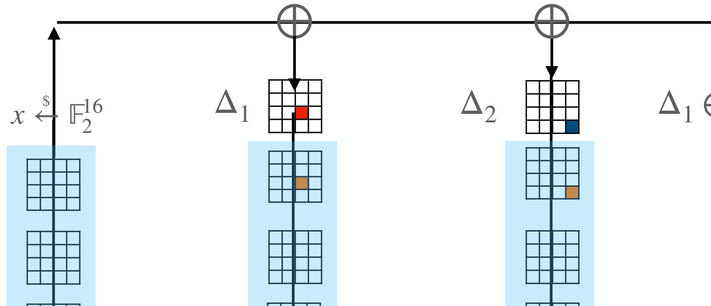






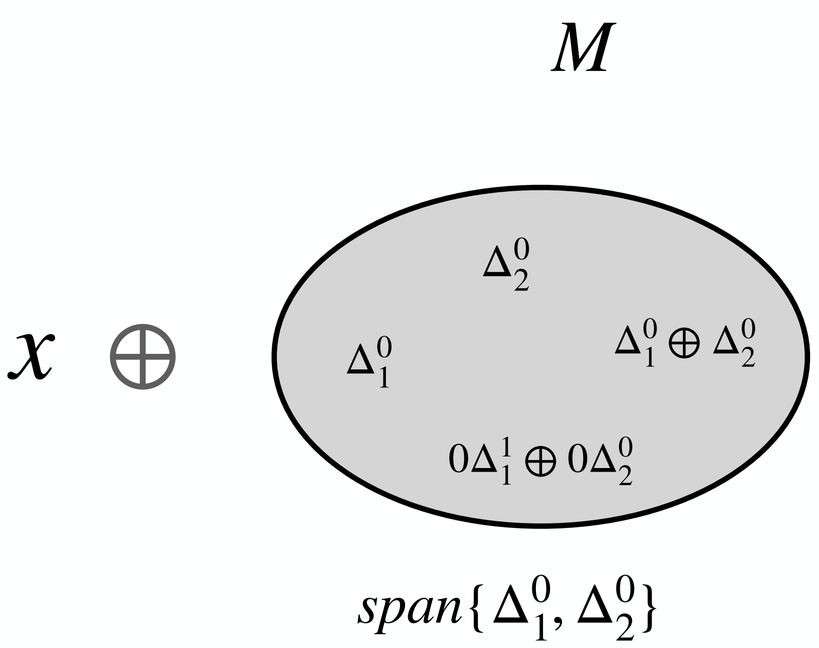




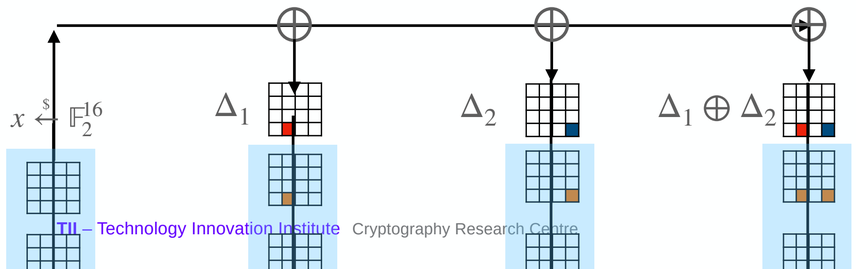


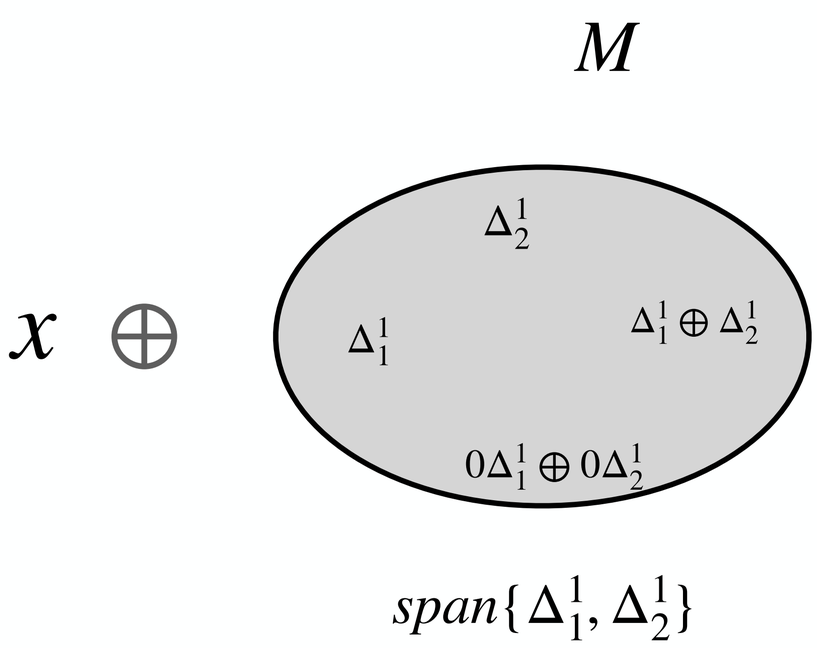
Full-diffusion criterion for the second order case

We need $\binom{n}{d}$ vector spaces

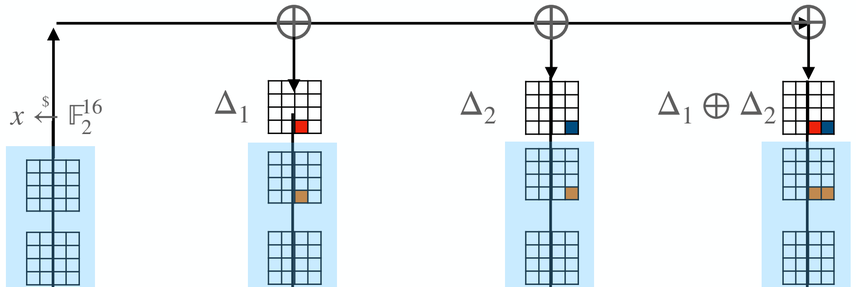


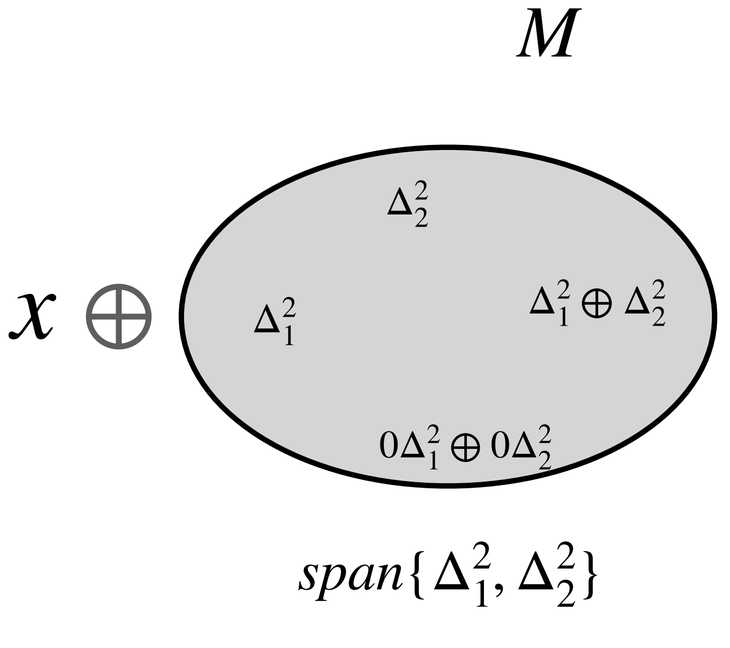
$span\{\Delta_1^0, \Delta_2^0\}$



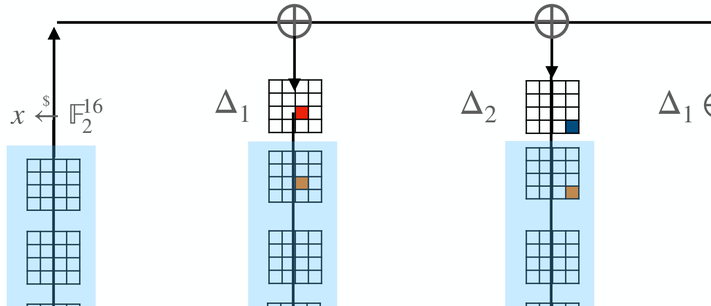


$span\{\Delta_1^1, \Delta_2^1\}$



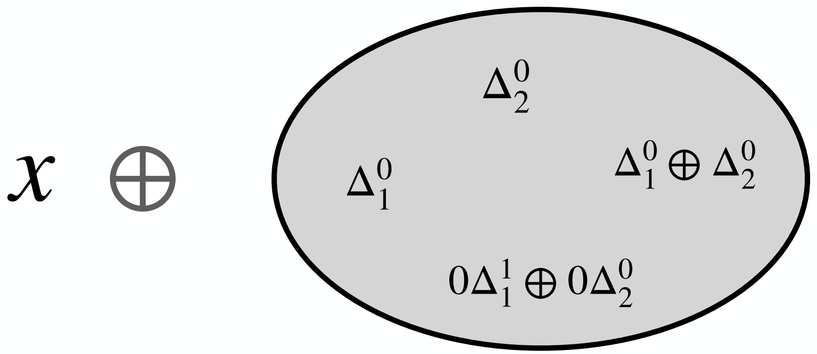


$span\{\Delta_1^2, \Delta_2^2\}$

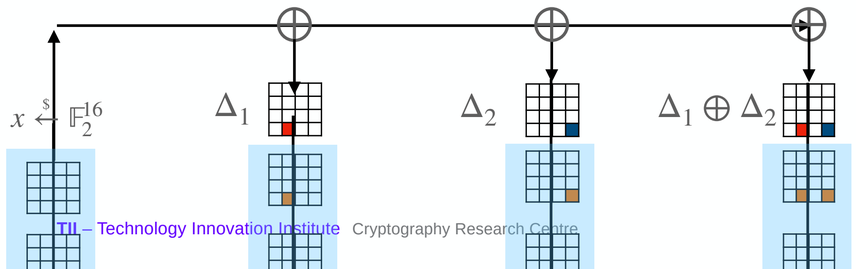


Full-diffusion criterion for the second order case

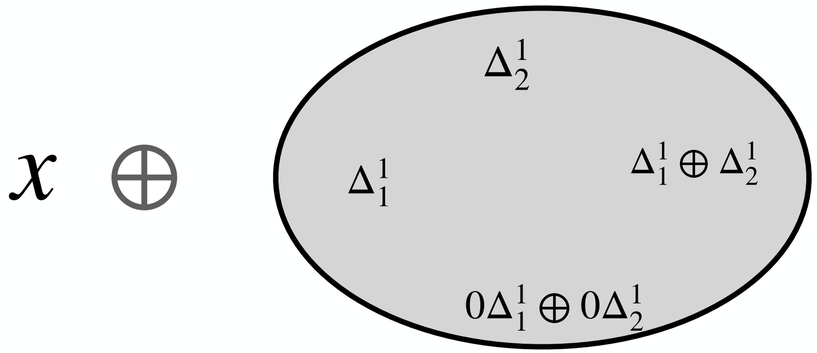
Repeat the process M times to obtain value of the metric



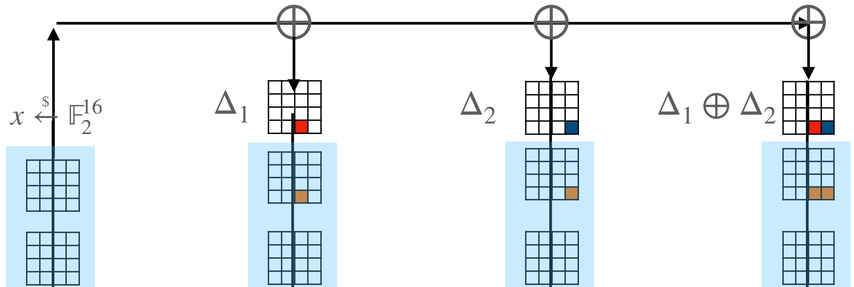
$$span\{\Delta_1^0, \Delta_2^0\}$$



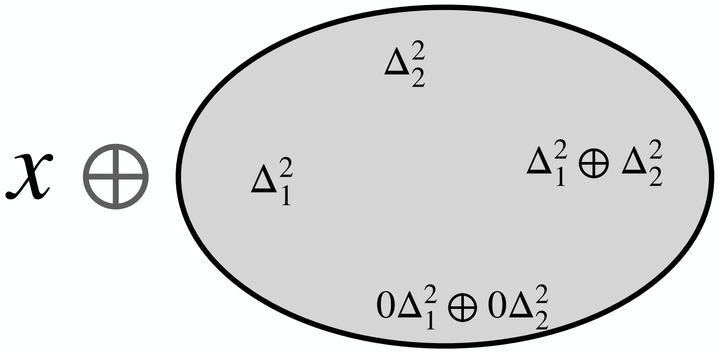
Repeat the process M times to obtain value of the metric



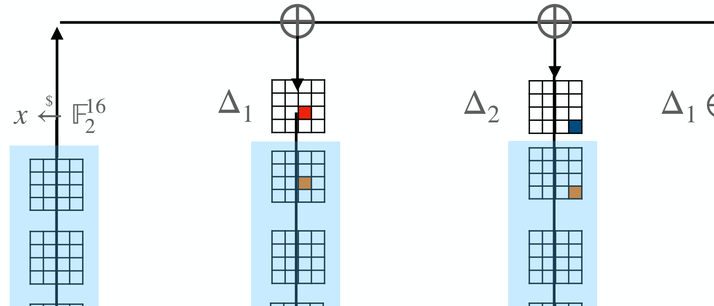
$$span\{\Delta_1^1, \Delta_2^1\}$$



Repeat the process M times to obtain value of the metric



$$span\{\Delta_1^2, \Delta_2^2\}$$



Ascon Case 4-order metrics

Computation of High-order metrics

- To compute the four-order derivate -> $span\{\Delta_1, \Delta_2, \Delta_3, \Delta_4\}$
- How many vector spaces we have? $\binom{320}{4} \approx 2^{29} \approx 428$ millions
- How many vectors in each vector space? 2^4

Number of cipher evaluations

$$\binom{320}{4} \quad 2^4 \quad M$$

Challenges

- Determine the workload,
- Parallelize the computation of the metrics associated to each vector space
- Distribute the main workload among threads, thread blocks, and the GPUs to compute the criteria,
- Avoid CPU bottleneck,
- Manage inter-block GPU synchronization,
- Avoid inter-GPU communication during the processing of avalanche tests.

GPU model parallelism

GPU model parallelism

- Data-level parallelism

GPU model parallelism

- Data-level parallelism
  - The number of cipher evaluations is $\binom{n}{d} \times M \times 2^d$

GPU model parallelism

- Data-level parallelism
  - The number of cipher evaluations is $\binom{n}{d} \times M \times 2^d$

GPU model parallelism

- Data-level parallelism
  - The number of cipher evaluations is $\binom{n}{d} \times M \times 2^d$
- Model-level parallelism

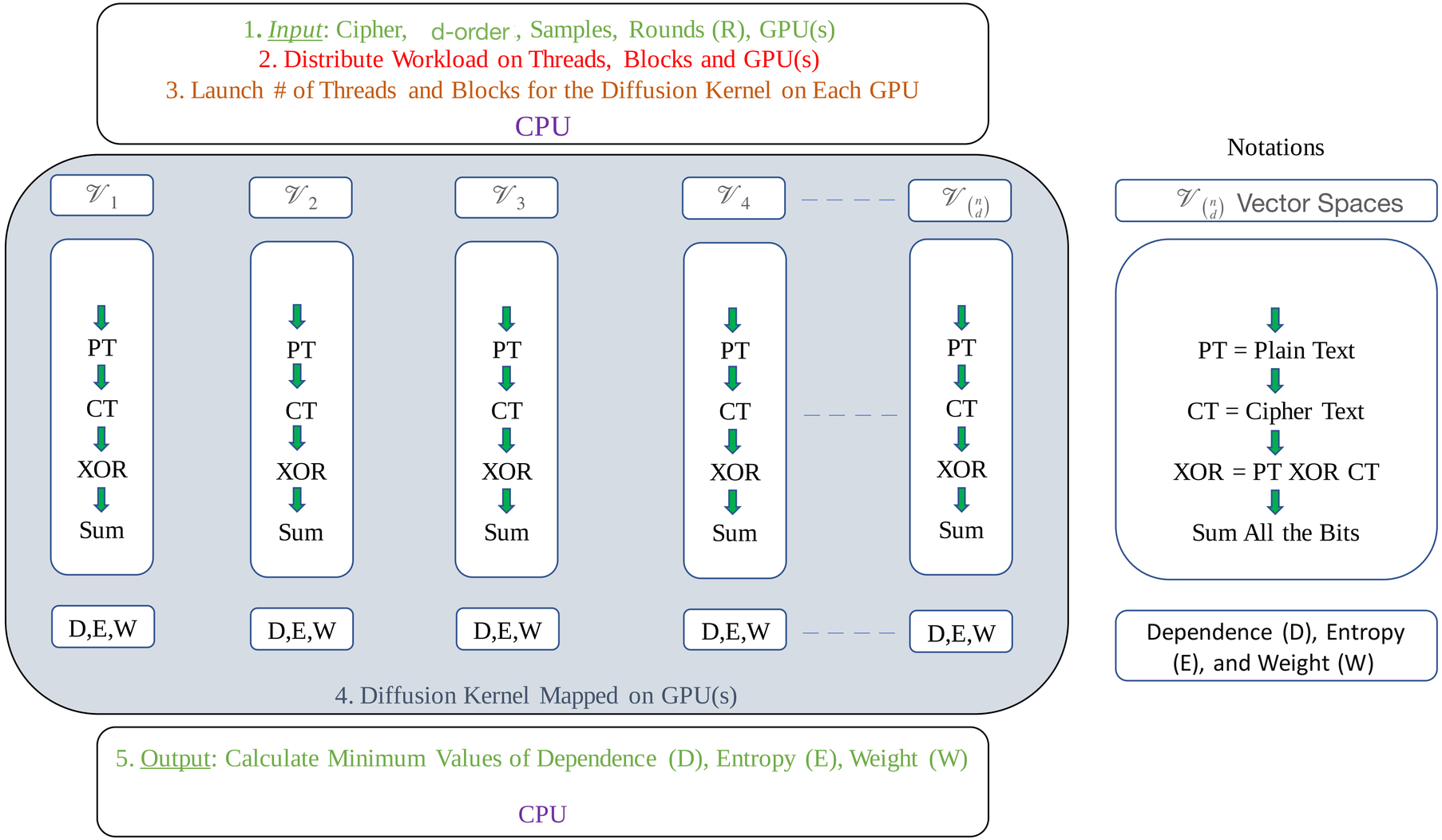
GPU model parallelism

- Data-level parallelism
  - The number of cipher evaluations is $\binom{n}{d} \times M \times 2^d$
- Model-level parallelism
  - The number of cipher evaluations is $\binom{n}{d} \times M \times 2^d$

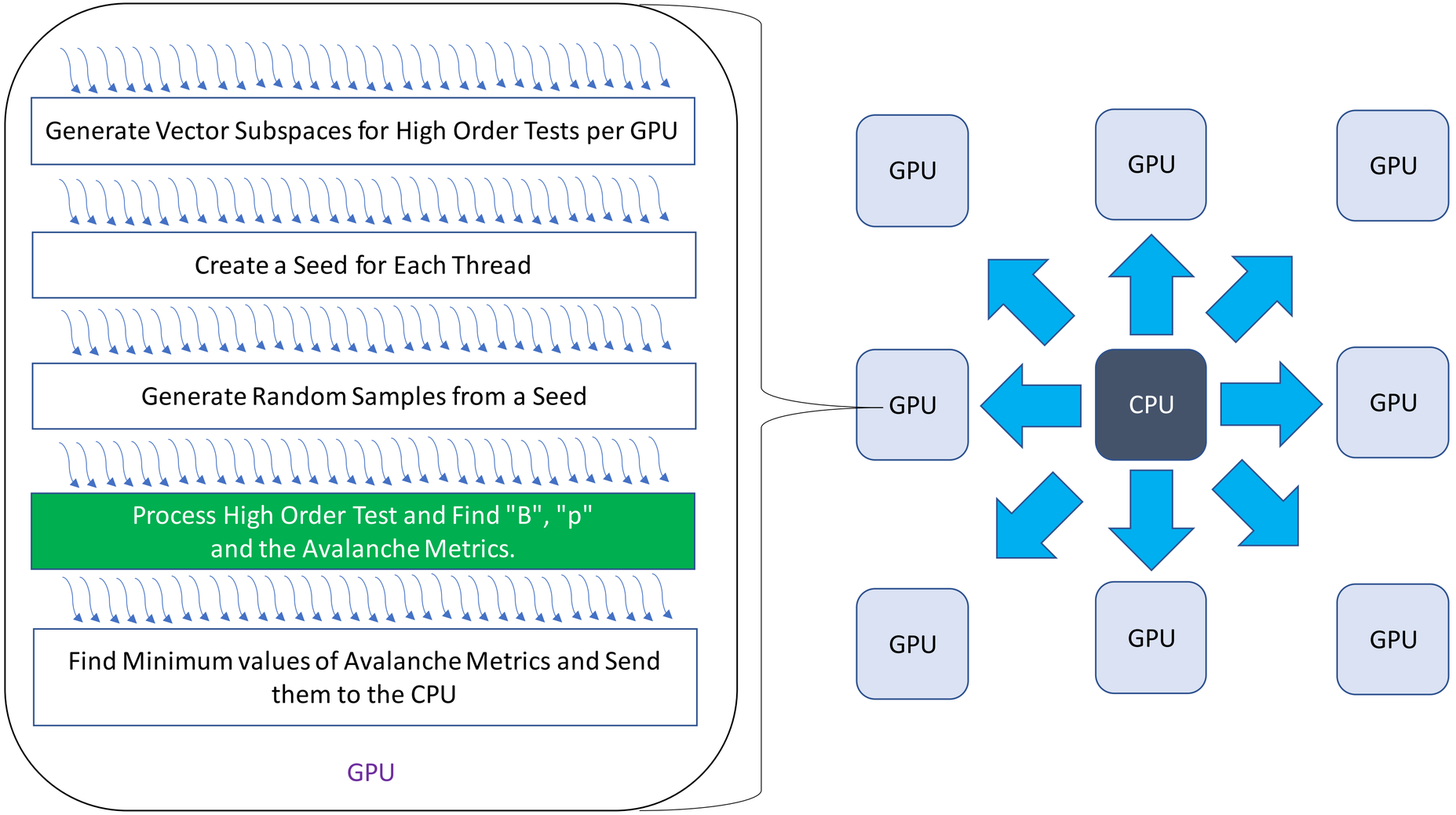
Model Level Parallelism Distribution

- Main contribution is from the number of vector spaces, i.e. $\binom{n}{d}$
- Number of threads should correspond to the number of vector spaces, i.e. $\binom{n}{d}$
- If we have 8 GPUs, then for each GPU the number of threads a.k.a vector spaces will be $\binom{n}{d}/8$
- If we have 32 blocks in each GPU, we need $\binom{n}{d}/8/32$ threads per block per GPU

Model Level Parallelism

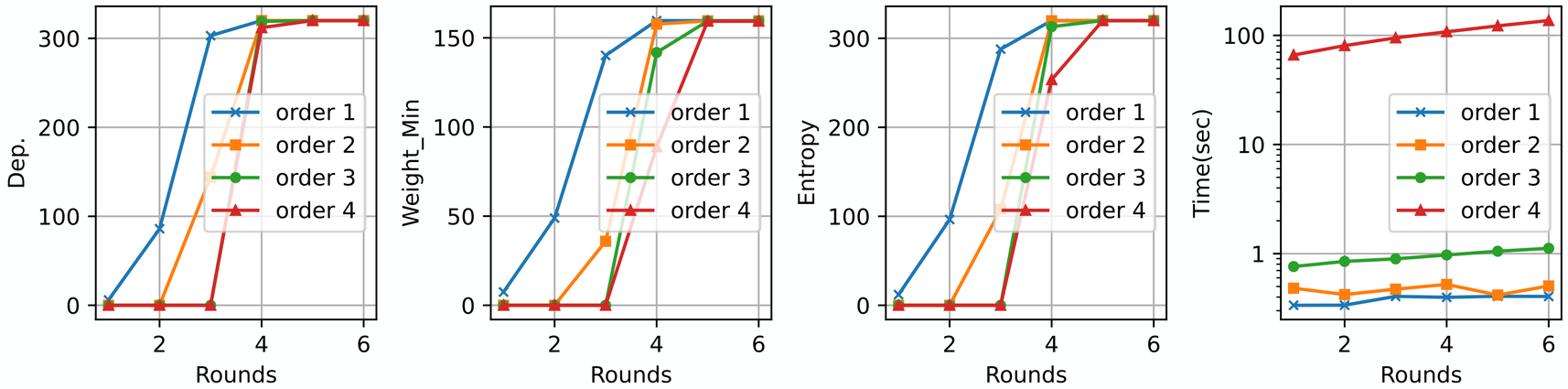


Workload in the GPU

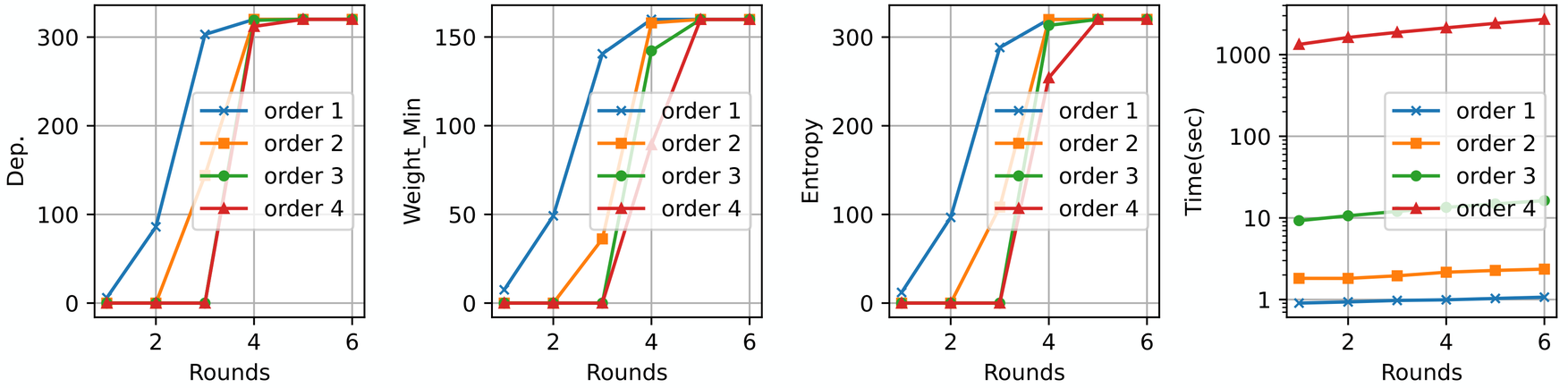


Report on ASCON

Samples 5000

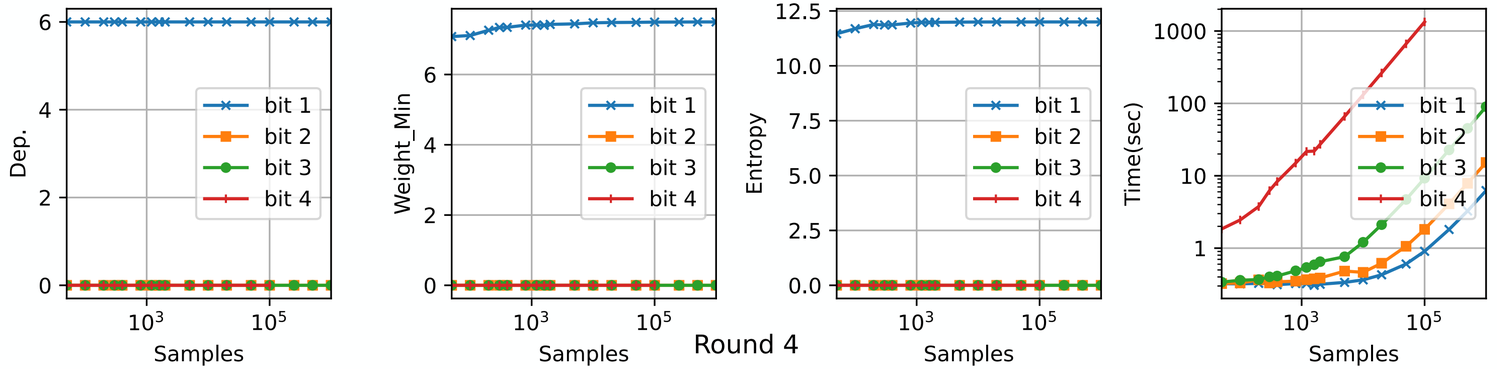


Samples 100000

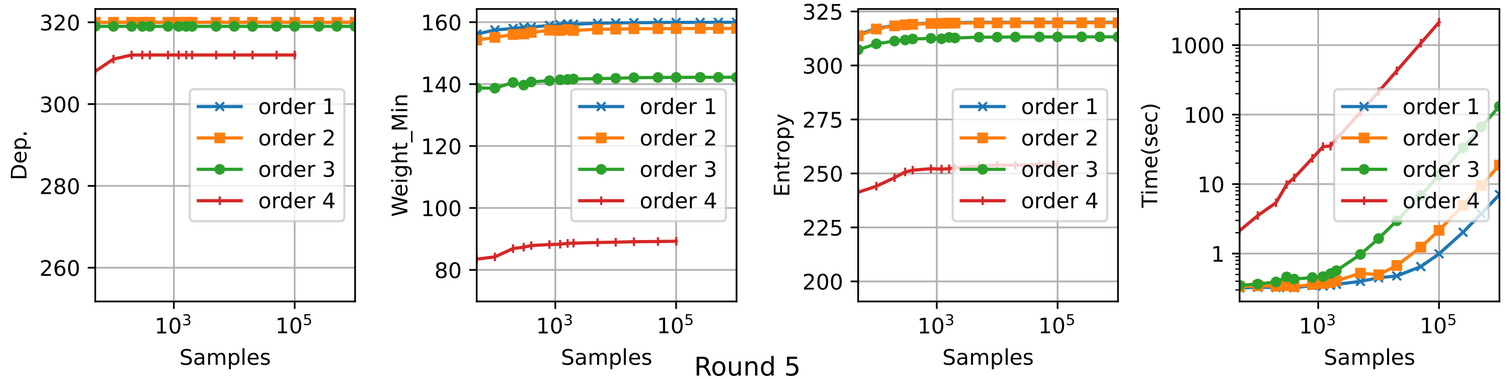


Report on ASCON

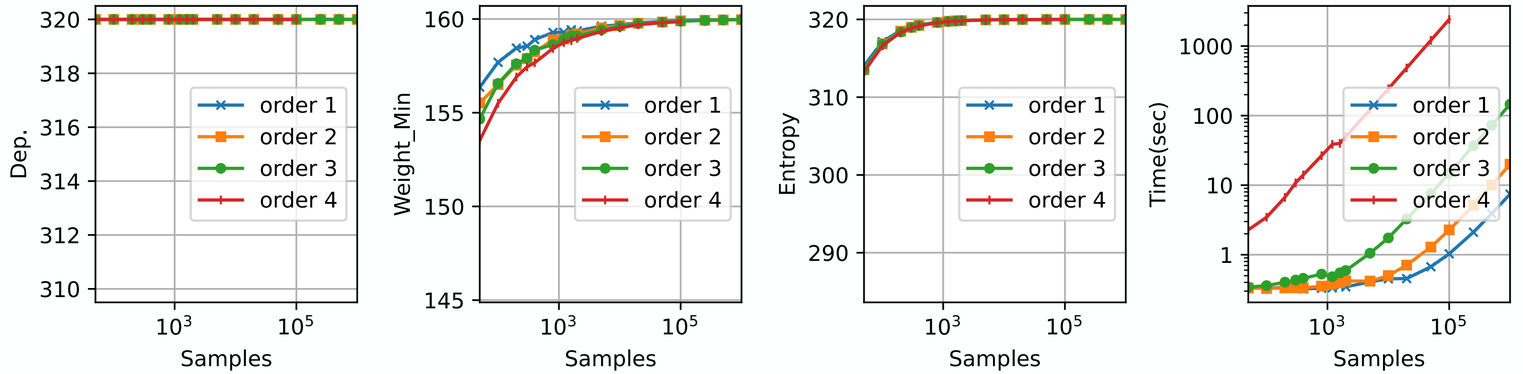
Round 1



Round 4



Round 5



Conclusion

- We propose and evaluate high-order avalanche tests on multi-GPU platforms,
- Use in the evaluation of parameters in the design phase of symmetric ciphers.
- This framework is general and can be easily adapted to test other block ciphers and permutations on several GPU models,
- We provided a detailed analysis of the permutation of the ASCON
  - Verified all the Ascon distinguishers presented by [Rohit2021] in minutes, while for them, it took weeks.
- The main challenge of this test was the huge number of cipher evaluations that need to be performed, especially for order 4 tests.
- Manage inter-block GPU synchronization, avoid communication between GPU-CPU, etc.
- We leave for future work to optimize the code and reach even higher orders and to study how the newly found biases could be exploited as a base to mount new or improve existing attacks.


Technology
Innovation
Institute
Cryptography
Research
Centre

tii.ae