KU LEUVEN

Threshold Implementations with Non-Uniform Inputs

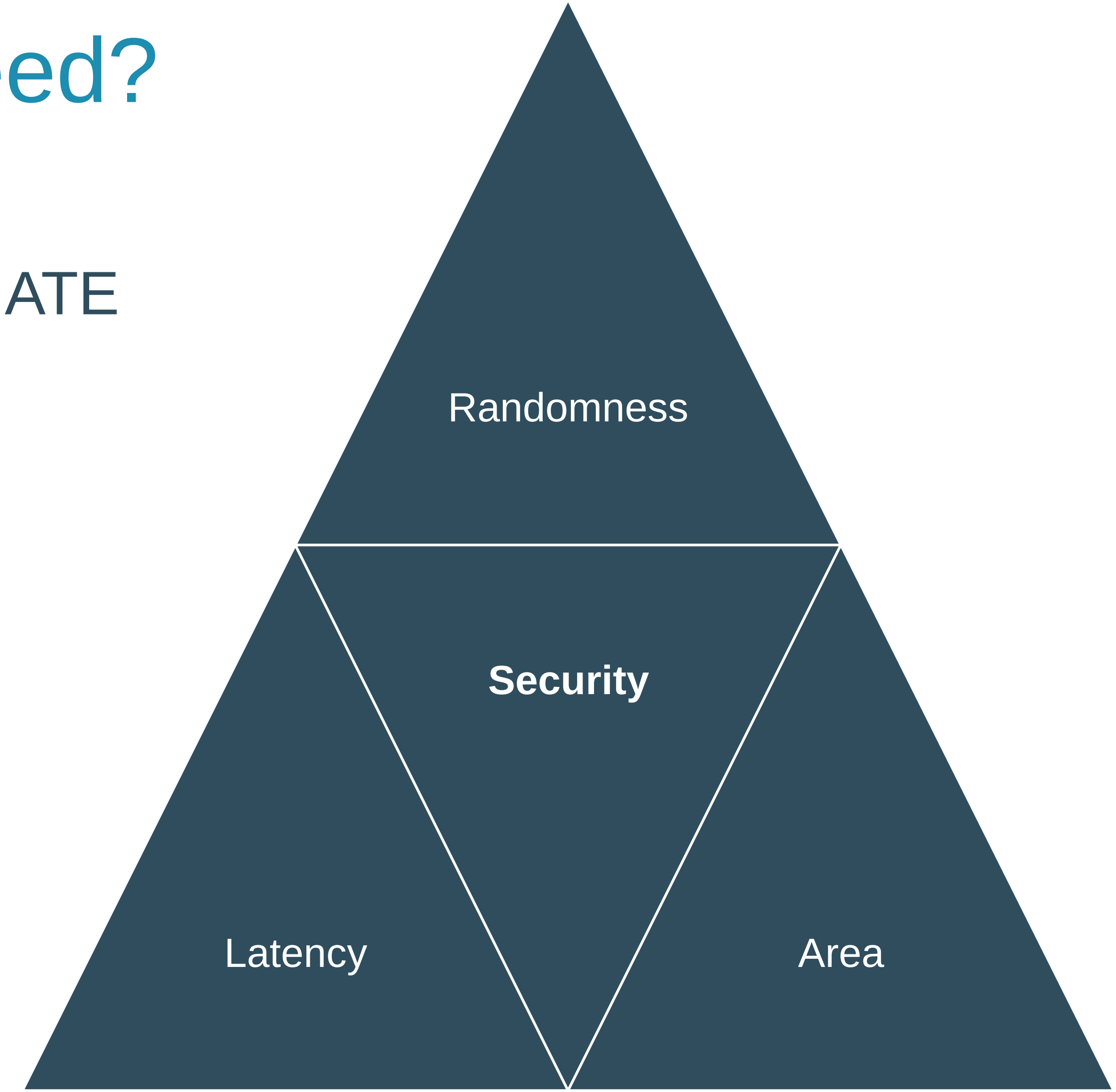
Siemen Dhooghe & Artemii Ovchinnikov

SAC 2023

COSIC

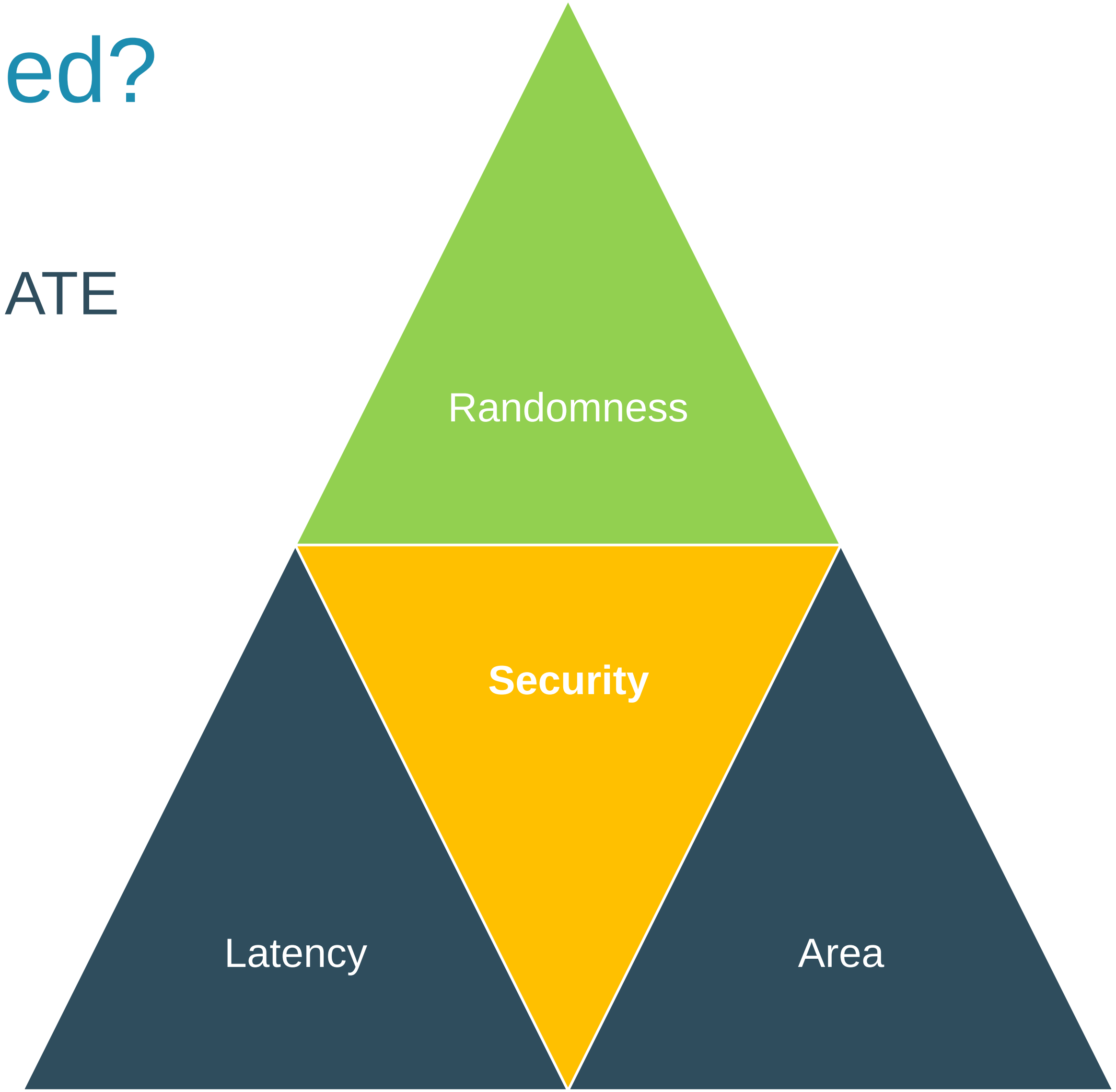
Hardware vs side-channel attacks: How much security do we need?

What if we will not EVALUATE, but ESTIMATE security…



Hardware vs side-channel attacks: How much security do we need?

What if we will not EVALUATE, but ESTIMATE security…



In our work we consider 1st order TIs

Definition 1 (Threshold implementations). *Let $F : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$ be a function and $\bar{F} : \mathbb{F}_2^{ns_x} \rightarrow \mathbb{F}_2^{ms_y}$ be a masking of F. The masking $\bar{F}$ is said to be*

1. correct *if* $\forall x^0, \ldots, x^{s_x-1} \in \mathbb{F}_2^n$, $\sum_{i=0}^{s_y-1} F^i(x^0, \ldots, x^{s_x-1}) = F(\sum_{i=0}^{s_x-1} x_i)$,
2. non-complete *if any function share F^i depends on at most $s_x - 1$ input shares,*
3. uniform *if $\bar{F}$ maps a uniform random masking of any $x \in \mathbb{F}_2^n$ to a uniform random masking of $F(x) \in \mathbb{F}_2^m$.*

Glitch-extended bounded-query probing model

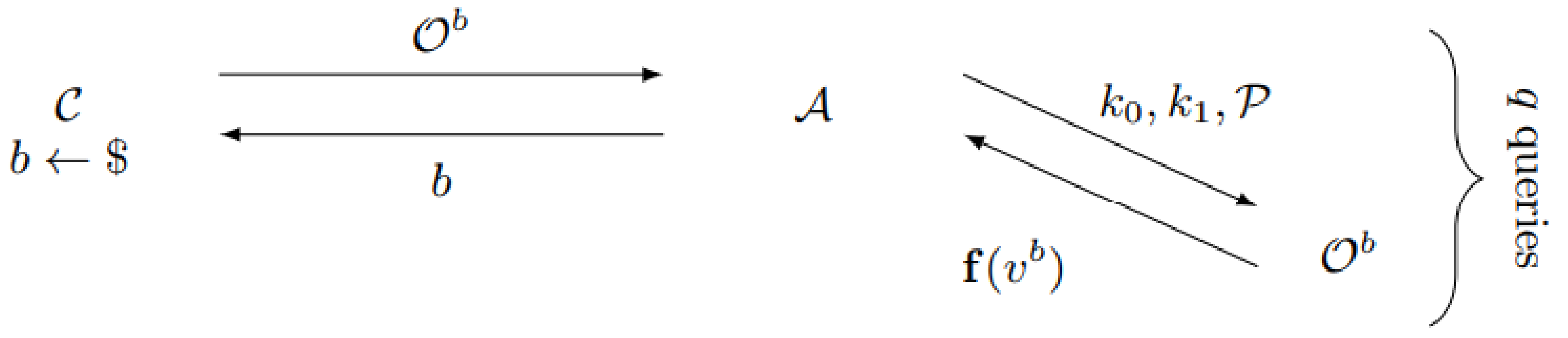


Figure 1: The privacy model for the glitch-extended t-threshold-probing security consisting of a challenger $\mathcal{C}$, an adversary $\mathcal{A}$, a left-right oracle $\mathcal{O}^b$, two inputs k_0, k_1, a set of probes $\mathcal{P}$, and a noisy leakage function $\mathbf{f}(v^b)$ of the probed wire values v^b in the circuit $C(k_b)$.

Adversary advantage

Theorem 1. *Let $\mathcal{A}$ be a noisy threshold-probing adversary for a circuit C. Take $\lambda \geq 1$, and $\varepsilon \leq 1$ as non-negative real numbers. Assume that for every query made by $\mathcal{A}$ on the oracle $\mathcal{O}^b$ with result $\mathbf{z}$, there exists a partitioning (depending only on the probe positions) of the probed wire values into two random variables $\mathbf{x}$ ('good') and $\mathbf{y}$ ('bad') such that*

1. *The noisy leakage function $\mathbf{f}$ such that $\mathbf{z} = \mathbf{f}(\mathbf{x}, \mathbf{y})$ is λ-noisy.*
2. *The conditional probability distribution $p_{\mathbf{y}|\mathbf{x}}$ satisfies $\mathbb{E}_{\mathbf{x}} \|\widehat{p}_{\mathbf{y}|\mathbf{x}}\|_2^2 \leq \varepsilon$.*
3. *Any t-threshold-probing adversary for the same circuit C and making the same oracle queries as $\mathcal{A}$, but which only receives the 'good' wire values (i.e. corresponding to $\mathbf{x}$) for each query, has advantage zero.*

The advantage of $\mathcal{A}$ can be upper bounded as

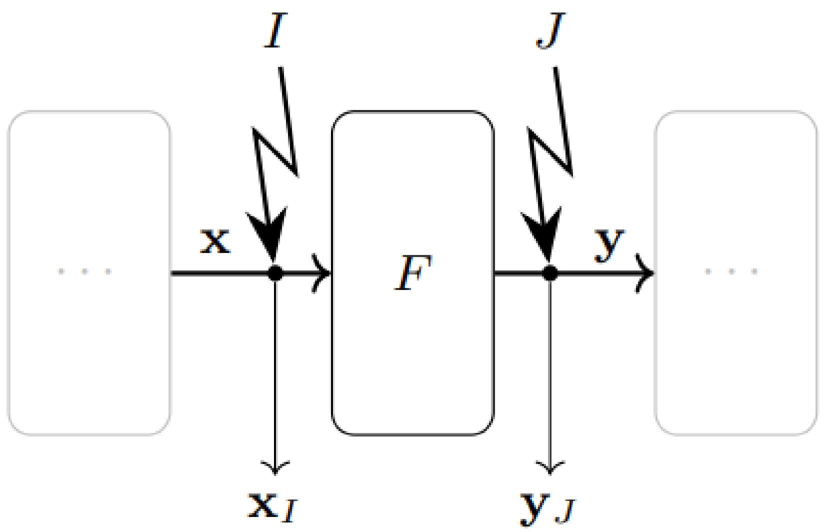
$$\boxed{\mathrm{Adv}_{\mathsf{noisy}}(\mathcal{A}) \leq \sqrt{2q\,\varepsilon/\lambda}\,,}$$

where q is the number of queries to the oracle $\mathcal{O}^b$.

$$\varepsilon := \|\widehat{p}_{\mathbf{z}} - \delta_0\|_2^2 \leq |\mathrm{supp}\,\widehat{p}_{\mathbf{z}}|\,\|\widehat{p}_{\mathbf{z}} - \delta_0\|_\infty^2 \leq 2^{(\#wires)^{\#probes}} \left|C_{u,v}^S\right|^{(active\_cells)^2}$$

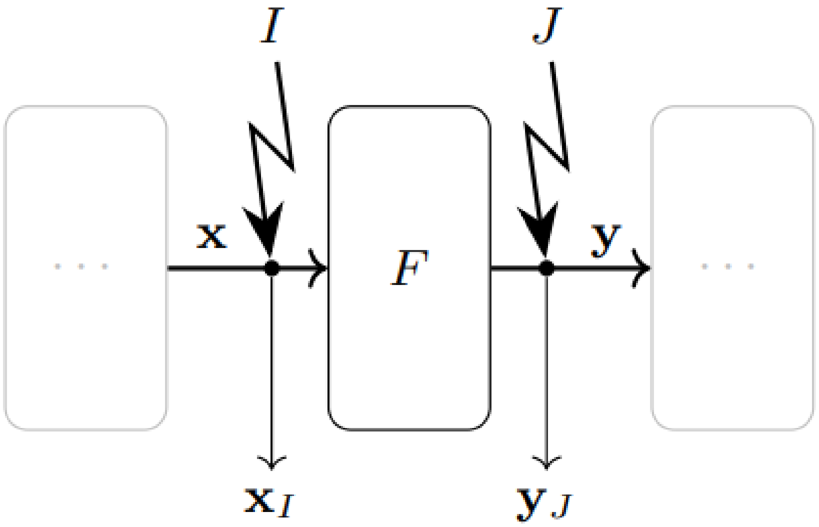
The previous model

Two probes => find correlation

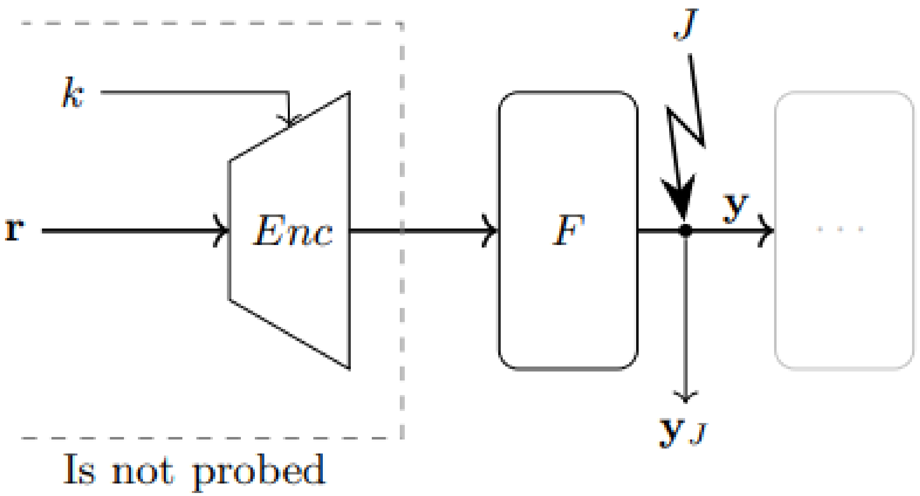


Our adaptation

Two probes => find correlation



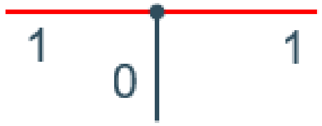
One probe => find correlation (via input patterns)

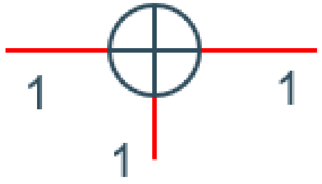


Linear cryptanalysis of masked ciphers

- **Linear mask** – tuple of bits used for linear approximations of functions. If the tuple is all 0s, then the mask is not applied;
- **Active bits** – all bits with value “1” in the linear mask. Ex: [0,**1**,0,0] or [**1**,0,0,**1**];

Let’s imagine we have <u>**a single active bit**</u> represented by <u>**wire**</u> with value “1”. There are two main types of operations with bits for linear/affine functions:

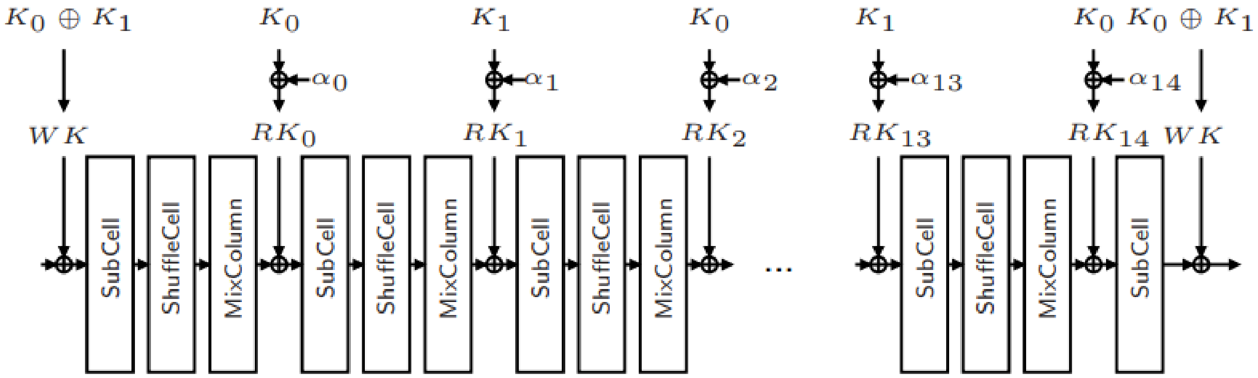
- **Branching**: 

node works as XOR-gate for any 2 inputs;
- **XOR-ing**: 

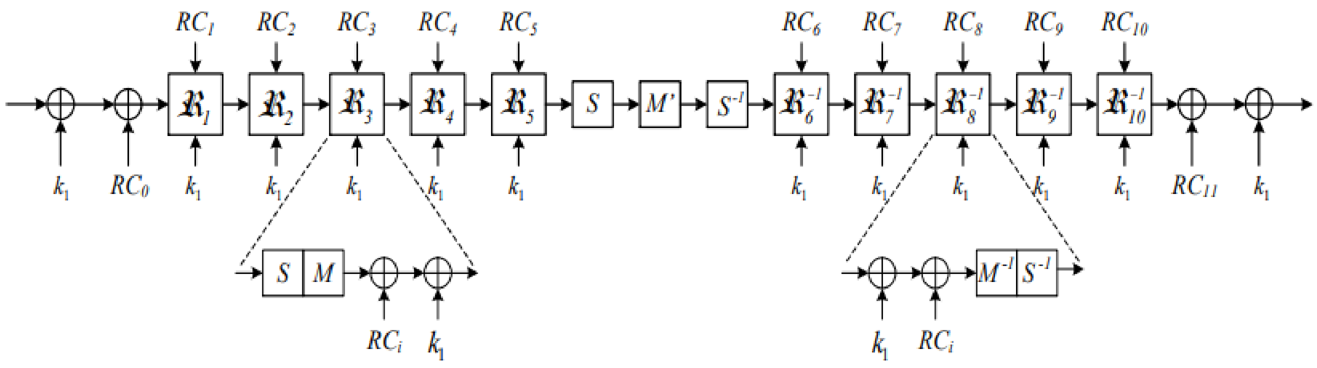
XOR-gate works as if it is a node in hardware.

Chosen ciphers

Midori64 overview:



Prince (core):



	Midori64	**Prince**
#Shares	3	3
State size	64	64
Random. bits	128	128
Latency	32	36
Area (GE)	7324	8353
Absolute Correlation	2^{-2}	$2^{-1.41}$

Practical evaluation

PROLEAD

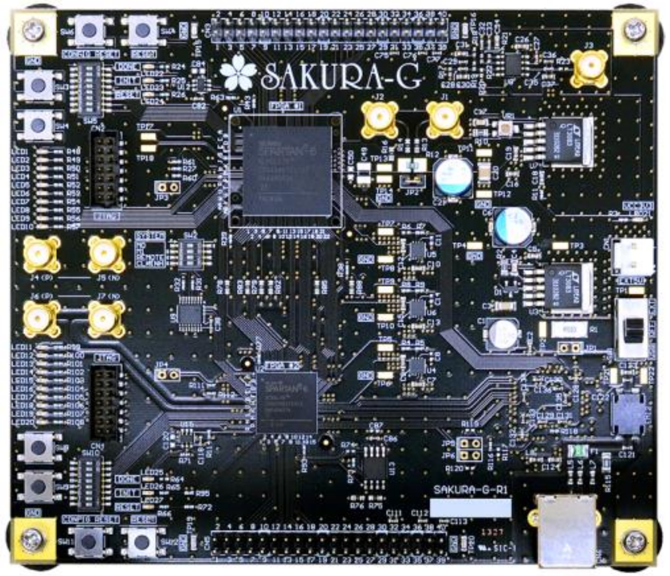
- G-test
- $\lambda = 1$ $\quad \mathrm{Adv}_{\mathsf{noisy}}(\mathcal{A}) \leq \sqrt{2q\,\varepsilon}$

```
Cycle 34: @[\uut.main_part.midori_nonlinear_layer.
Sbox[3].Sbox_i.register.in[1](34)] ==> [
\uut.main_part.first_register_1.state_out[50](34),
\uut.main_part.first_register_2.state_out[50](34),
\uut.main_part.first_register_2.state_out[48](34),
\uut.main_part.first_register_2.state_out[49](34),
\uut.main_part.first_register_1.state_out[48](34),
\uut.main_part.first_register_1.state_out[49](34)]
-log10(p) = 2.08937 --> OKAY
```

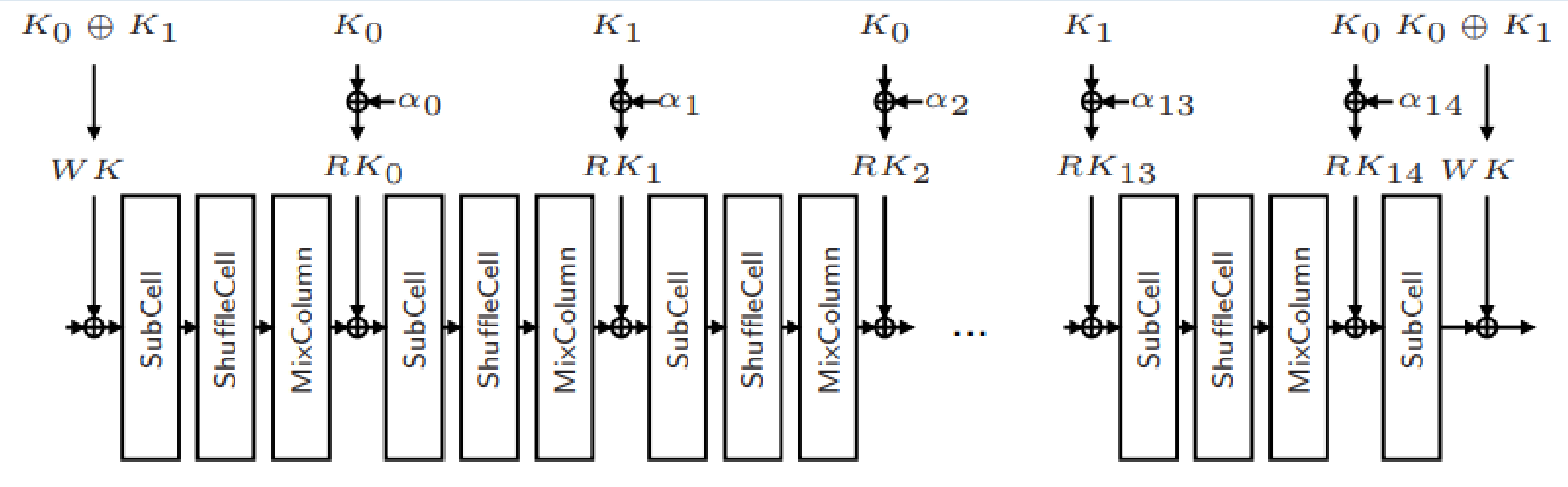
https://github.com/ChairImpSec/PROLEAD

FPGA

- t-test
- $\lambda \approx 2^9$ $\quad \mathrm{Adv}_{\mathsf{noisy}}(\mathcal{A}) \leq \sqrt{2q\,\varepsilon/\lambda}$



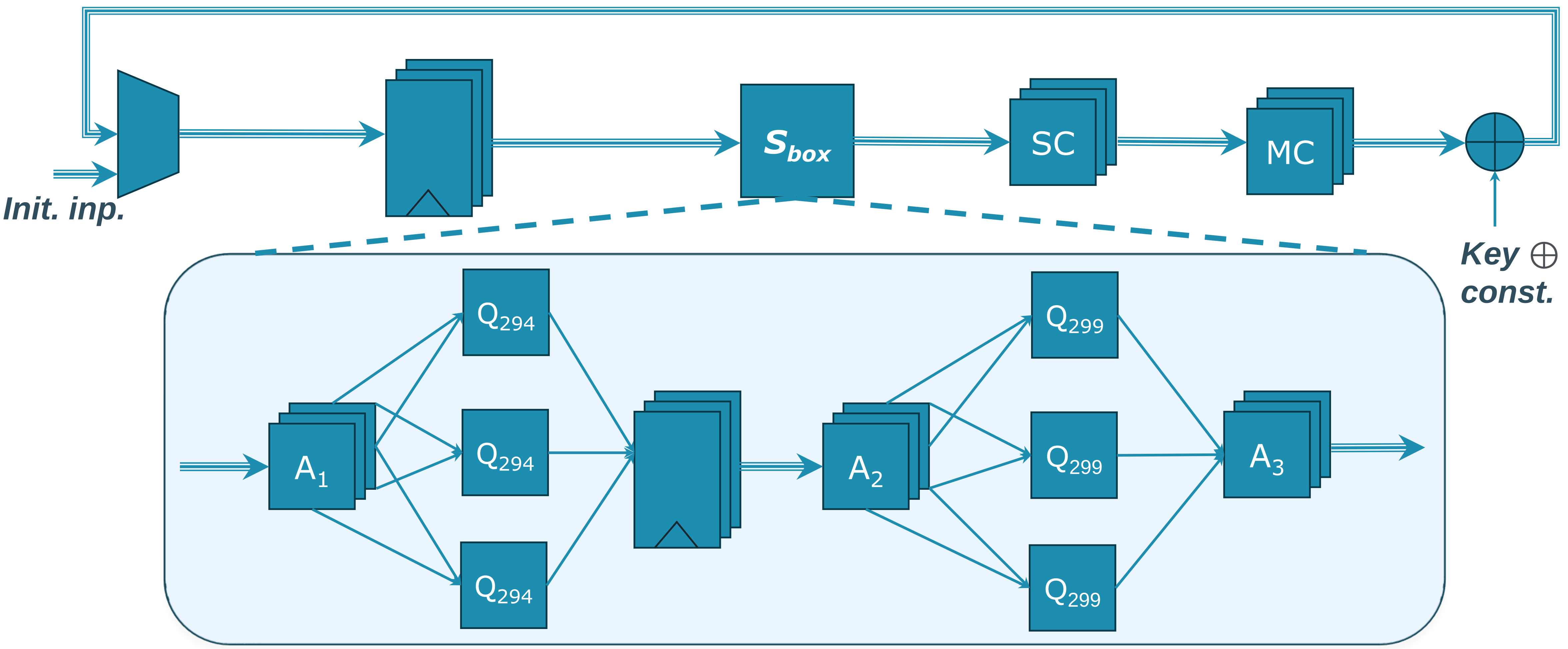
Midori64



$K_0 \oplus K_1$
K_0
K_1
K_0
K_1
K_0
$K_0 \oplus K_1$
α_0
α_1
α_2
α_{13}
α_{14}
WK
RK_0
RK_1
RK_2
RK_{13}
RK_{14}
WK
SubCell
ShuffleCell
MixColumn
...

Midori64 round

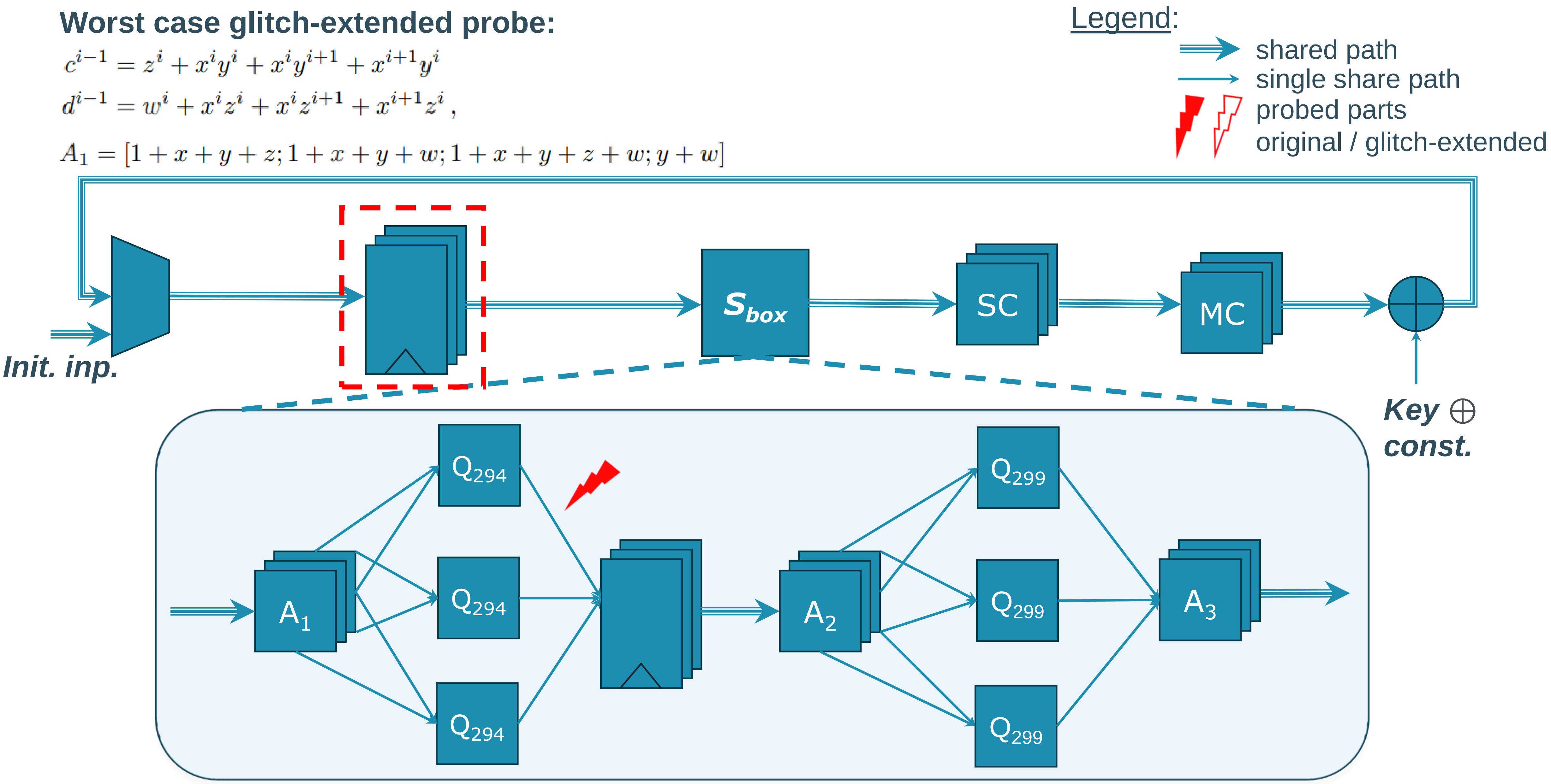
Legend:
shared path
single share path



Worst case glitch-extended probe:

$$c^{i-1} = z^i + x^i y^i + x^i y^{i+1} + x^{i+1} y^i$$
$$d^{i-1} = w^i + x^i z^i + x^i z^{i+1} + x^{i+1} z^i \,,$$

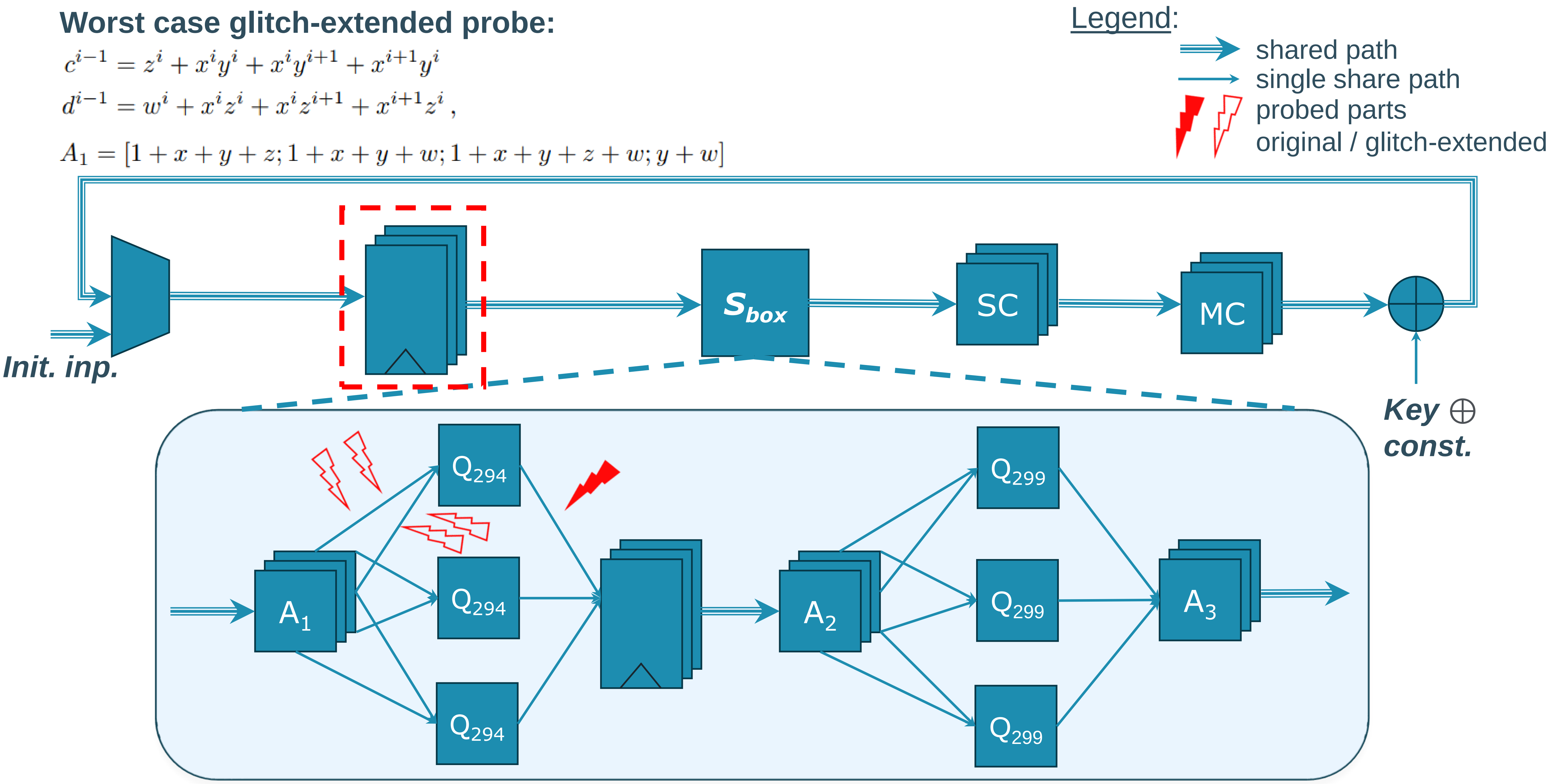
$$A_1 = [1 + x + y + z; 1 + x + y + w; 1 + x + y + z + w; y + w]$$



Worst case glitch-extended probe:

$$c^{i-1} = z^i + x^i y^i + x^i y^{i+1} + x^{i+1} y^i$$
$$d^{i-1} = w^i + x^i z^i + x^i z^{i+1} + x^{i+1} z^i \,,$$

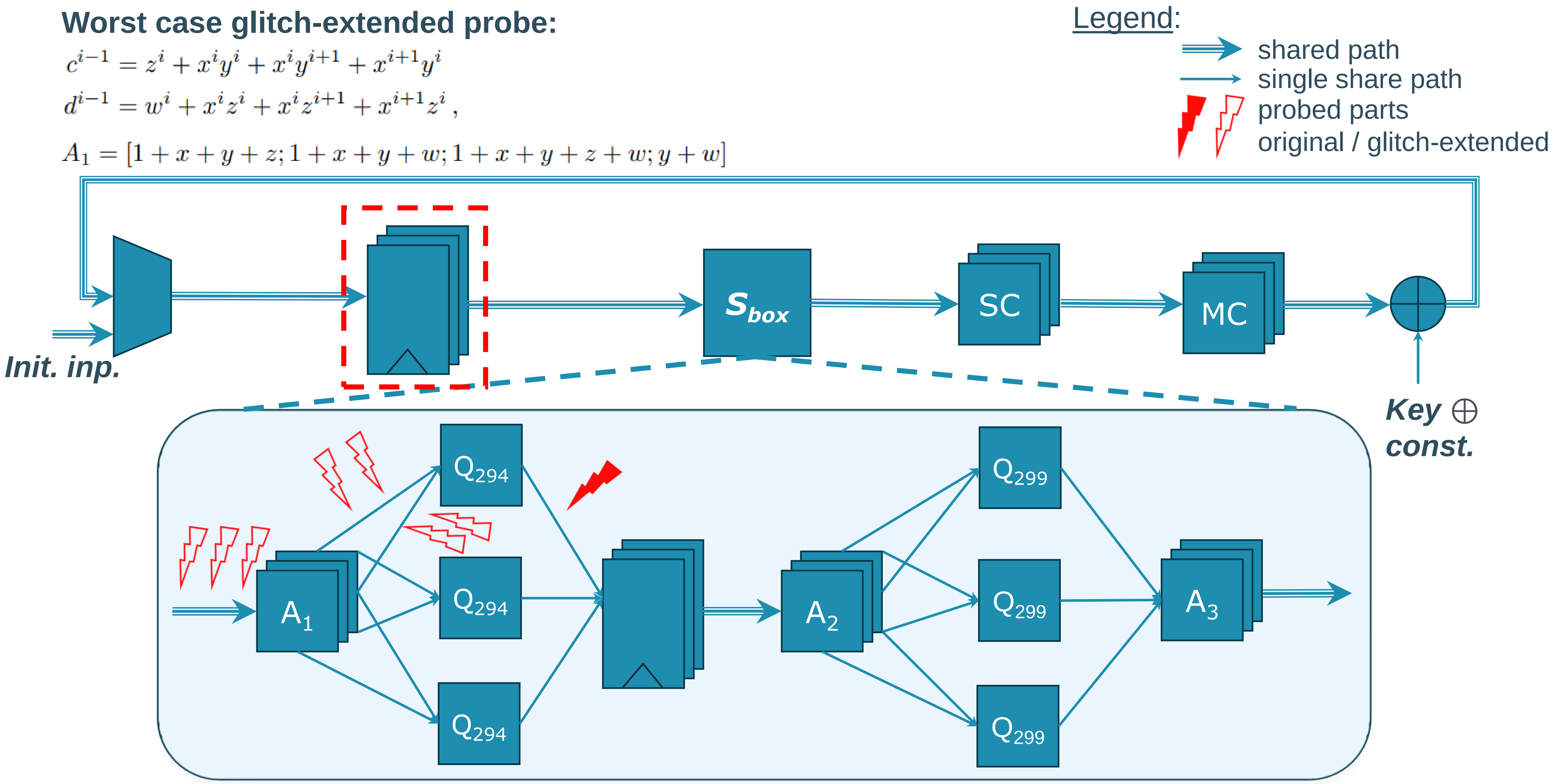
$$A_1 = [1 + x + y + z; 1 + x + y + w; 1 + x + y + z + w; y + w]$$



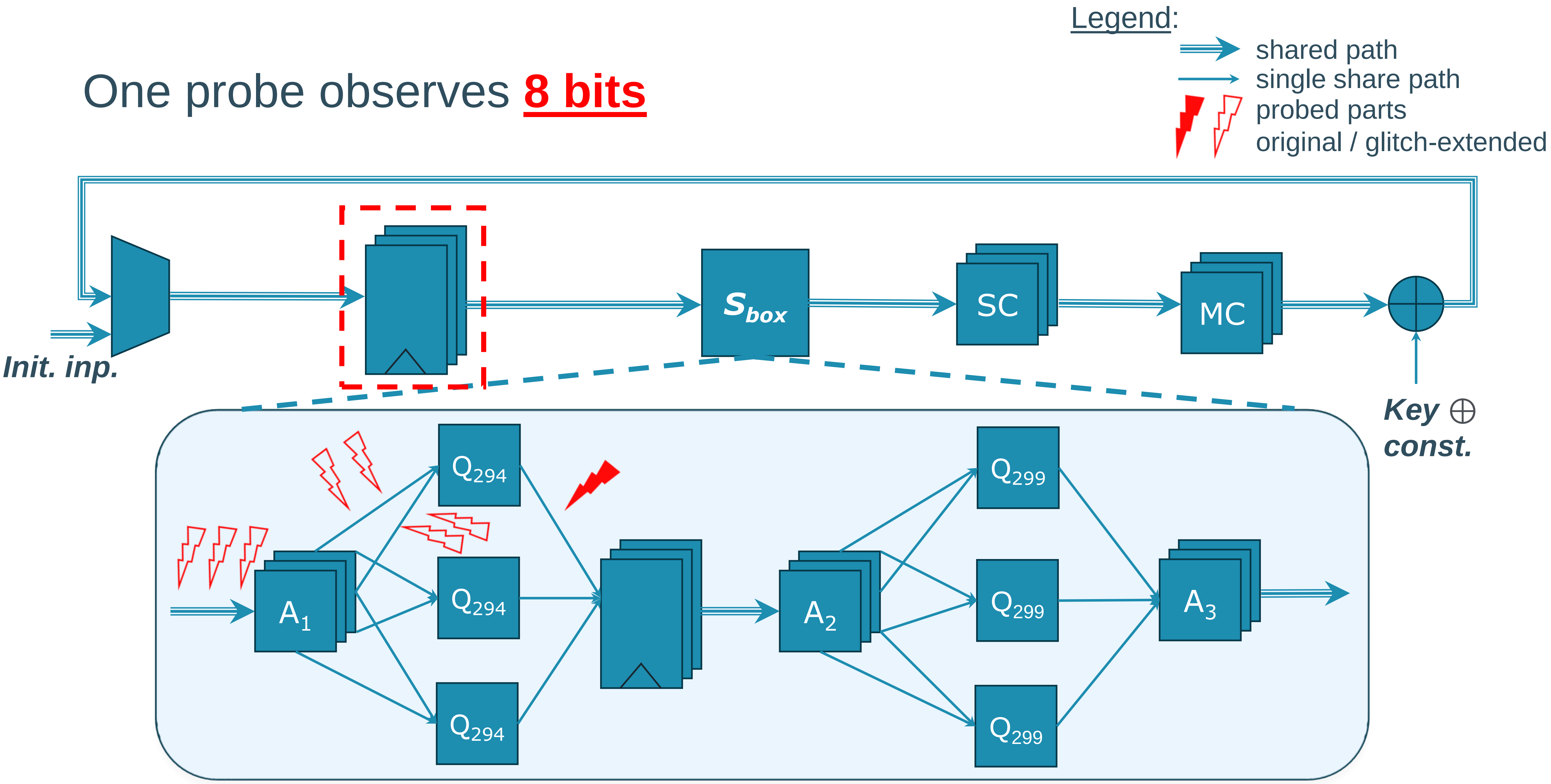
Worst case glitch-extended probe:

$$c^{i-1} = z^i + x^i y^i + x^i y^{i+1} + x^{i+1} y^i$$
$$d^{i-1} = w^i + x^i z^i + x^i z^{i+1} + x^{i+1} z^i ,$$

$$A_1 = [1 + x + y + z; 1 + x + y + w; 1 + x + y + z + w; y + w]$$



One probe observes **8 bits**

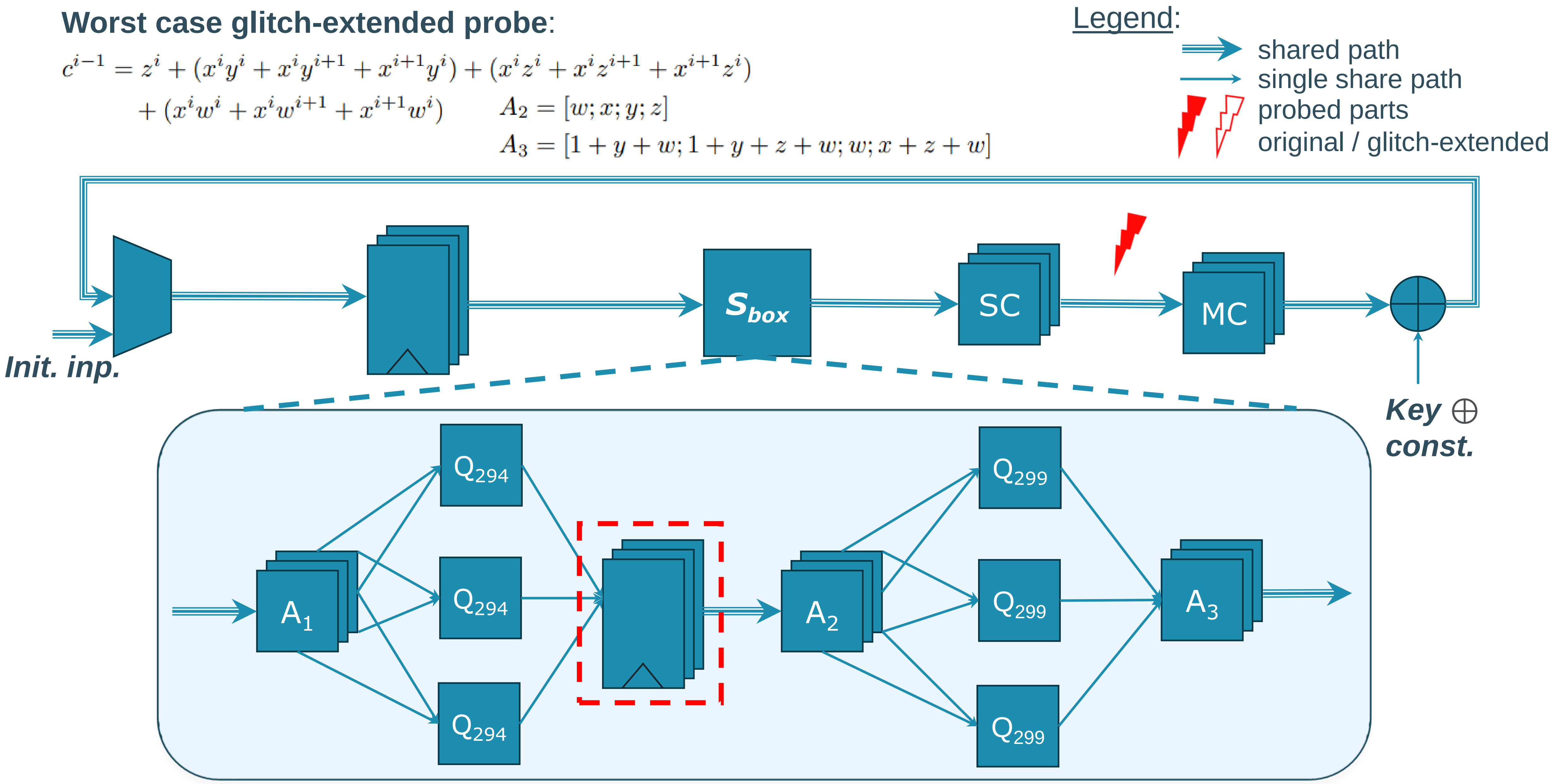


Worst case glitch-extended probe:

$$c^{i-1} = z^i + (x^iy^i + x^iy^{i+1} + x^{i+1}y^i) + (x^iz^i + x^iz^{i+1} + x^{i+1}z^i) + (x^iw^i + x^iw^{i+1} + x^{i+1}w^i)$$

$$A_2 = [w; x; y; z]$$

$$A_3 = [1 + y + w; 1 + y + z + w; w; x + z + w]$$



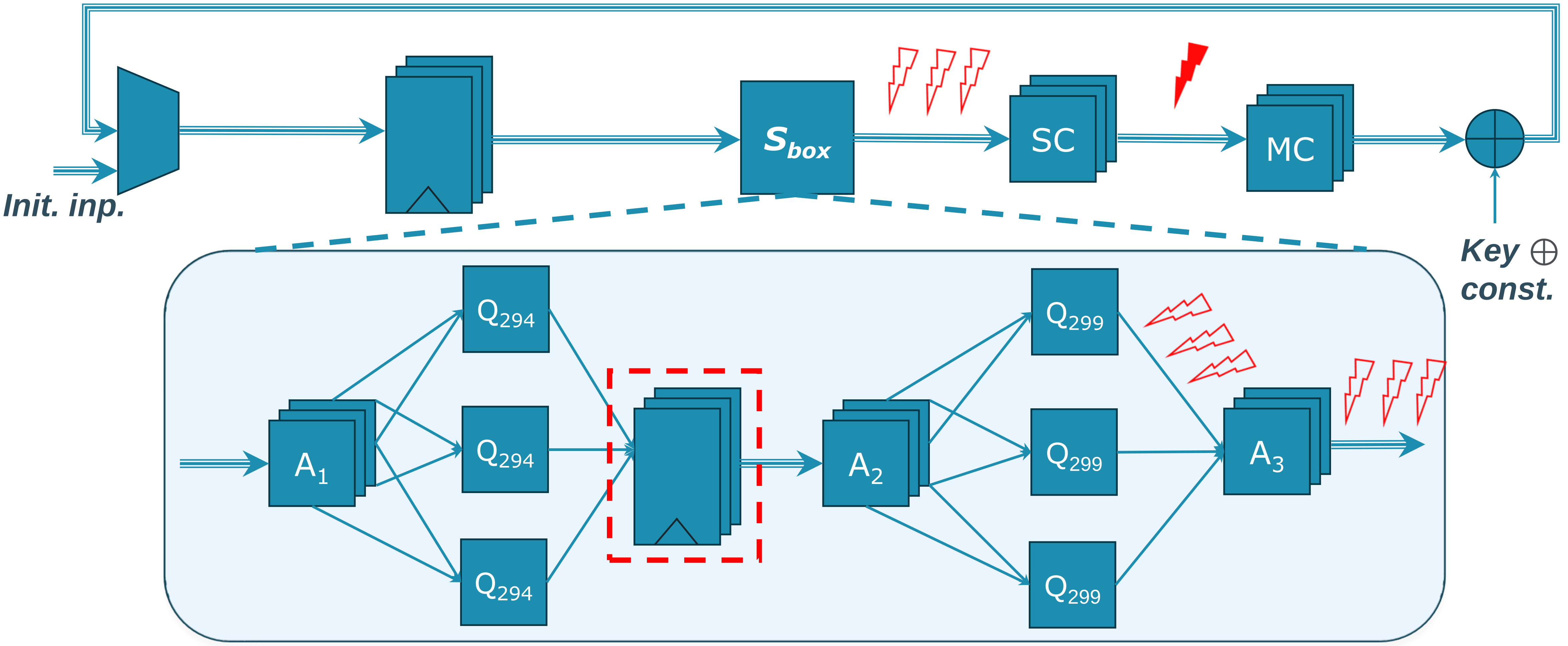
Worst case glitch-extended probe:

$$c^{i-1} = z^i + (x^i y^i + x^i y^{i+1} + x^{i+1} y^i) + (x^i z^i + x^i z^{i+1} + x^{i+1} z^i) + (x^i w^i + x^i w^{i+1} + x^{i+1} w^i)$$

$$A_2 = [w; x; y; z]$$

$$A_3 = [1 + y + w; 1 + y + z + w; w; x + z + w]$$

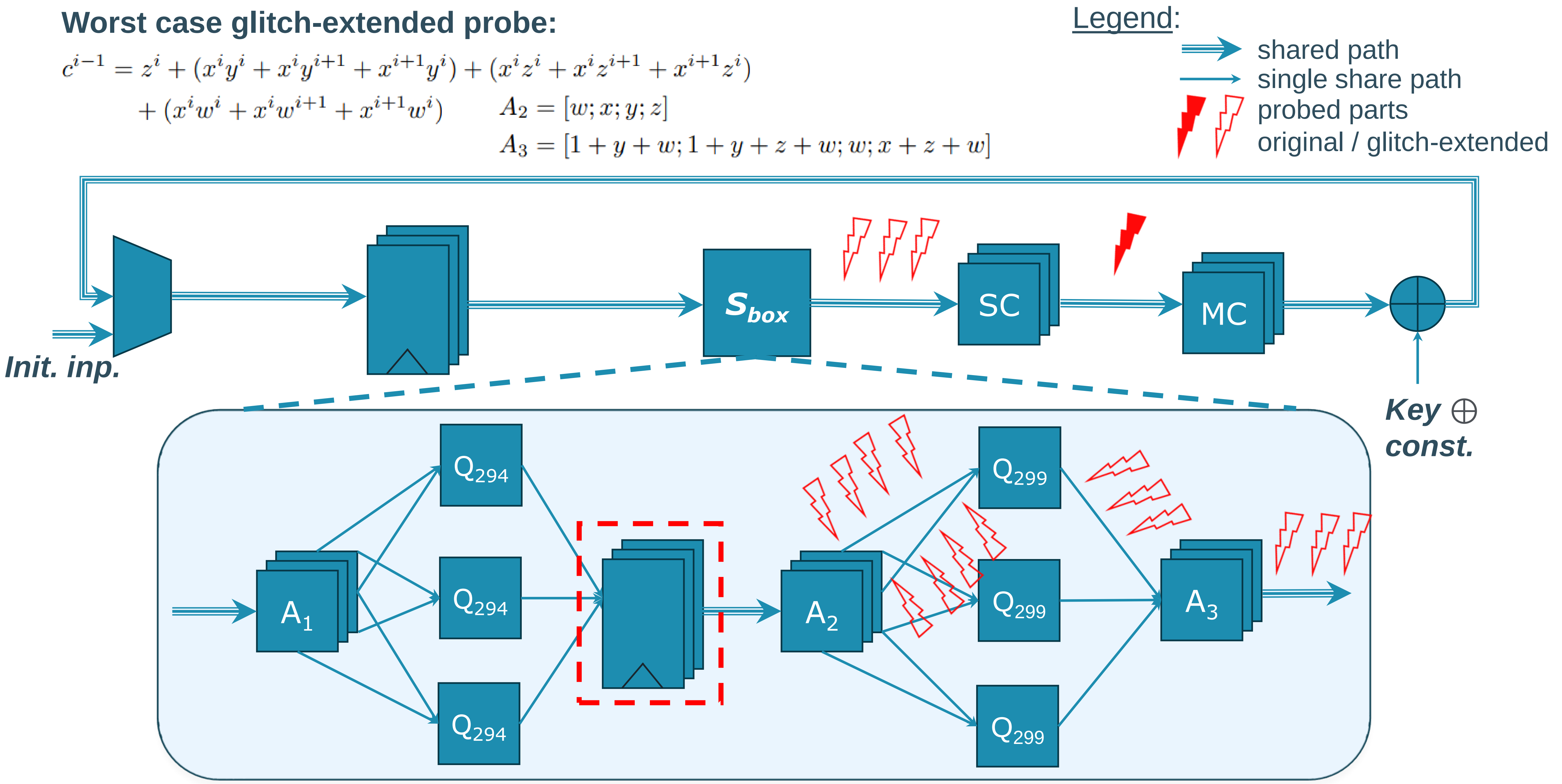
Legend:
- shared path
- single share path
- probed parts original / glitch-extended



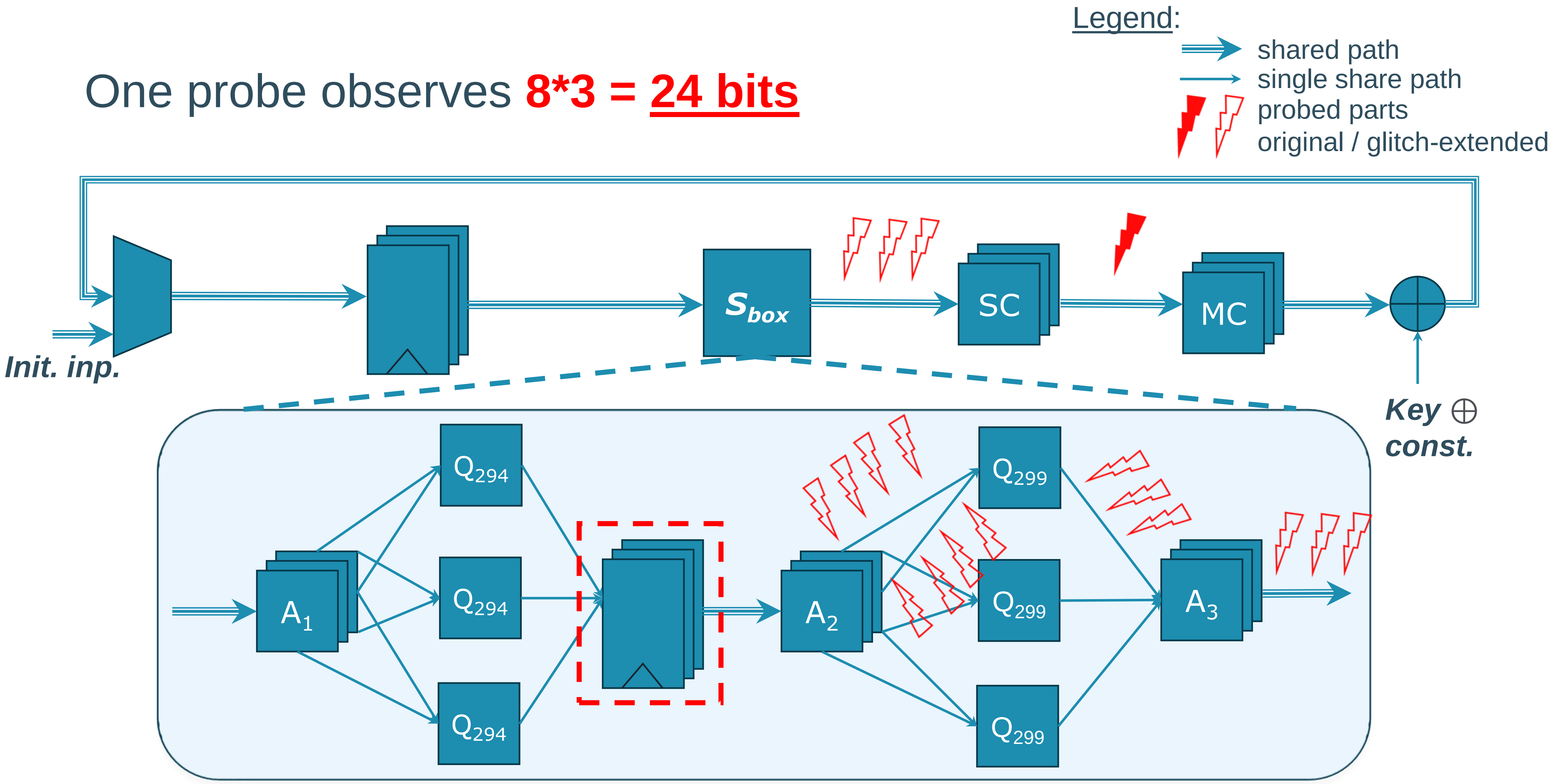
Worst case glitch-extended probe:

$$c^{i-1} = z^i + (x^i y^i + x^i y^{i+1} + x^{i+1} y^i) + (x^i z^i + x^i z^{i+1} + x^{i+1} z^i)$$
$$+ (x^i w^i + x^i w^{i+1} + x^{i+1} w^i)$$

$$A_2 = [w; x; y; z]$$
$$A_3 = [1 + y + w; 1 + y + z + w; w; x + z + w]$$



One probe observes 8*3 = 24 bits



Midori64 Trail – 1 Round

S0	S4	S8	S12
S1	S5	S9	S13
S2	S6	S10	S14
S3	S7	S11	S15

After S-box

S0	S14	S9	S7
S10	S4	S3	S13
S5	S11	S12	S2
S15	S1	S6	S8

After ShuffleCell

S0	S14	S9	S7
S10	S4	S3	S13
S5	S11	S12	S2
S15	S1	S6	S8

After MixColumns

Midori64 Trail – 1 Round

S0	S4	S8	S12
S1	S5	S9	S13
S2	S6	S10	S14
S3	S7	S11	S15

After S-box

S0	S14	S9	S7
S10	S4	S3	S13
S5	S11	S12	S2
S15	S1	S6	S8

After ShuffleCell

S0	S14	S9	S7
S10	S4	S3	S13
S5	S11	S12	S2
S15	S1	S6	S8

After MixColumns

Midori64 Trail – 1 Round

S0	S4	S8	S12
S1	S5	S9	S13
S2	S6	S10	S14
S3	S7	S11	S15

After S-box

S0	S14	S9	S7
S10	S4	S3	S13
S5	S11	S12	S2
S15	S1	S6	S8

After ShuffleCell

S0	S14	S9	S7
S10	S4	S3	S13
S5	S11	S12	S2
S15	S1	S6	S8

After MixColumns

Midori64 Trail – 1 Round

S0	S4	S8	S12
S1	S5	S9	S13
S2	S6	S10	S14
S3	S7	S11	S15

After S-box

S0	S14	S9	S7
S10	S4	S3	S13
S5	S11	S12	S2
S15	S1	S6	S8

After ShuffleCell

S0	S14	S9	S7
S10	S4	S3	S13
S5	S11	S12	S2
S15	S1	S6	S8

After MixColumns

Midori64 Trail – 1 Round

S0	S4	S8	S12
S1	S5	S9	S13
S2	S6	S10	S14
S3	S7	S11	S15

S0	S14	S9	S7
S10	S4	S3	S13
S5	S11	S12	S2
S15	S1	S6	S8

S0	S14	S9	S7
S10	S4	S3	S13
S5	S11	S12	S2
S15	S1	S6	S8

$$\varepsilon := \|\widehat{p}_{\mathbf{z}} - \delta_0\|_2^2 \le |\mathrm{supp}\,\widehat{p}_{\mathbf{z}}|\,\|\widehat{p}_{\mathbf{z}} - \delta_0\|_\infty^2 \le 2^{(\#wires)^{\#probes}} \left|C^S_{u,v}\right|^{(active\_cells)^2} >> 1$$

Midori64 Trail – 2 Rounds

S0	S4	S8	S12
S1	S5	S9	S13
S2	S6	S10	S14
S3	S7	S11	S15

After S-box

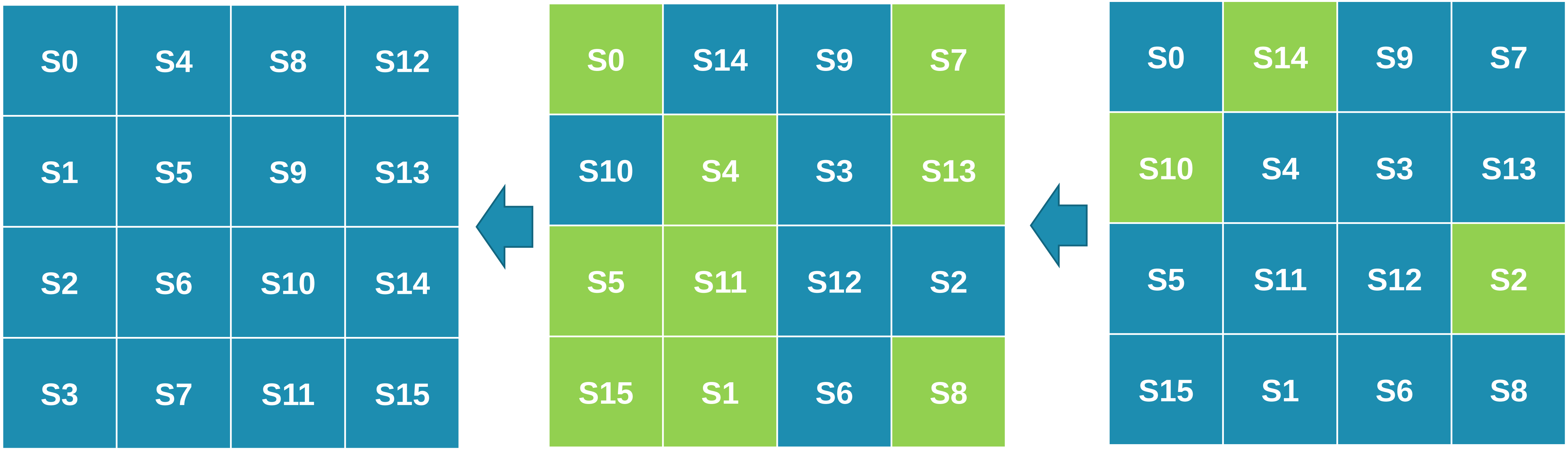
S0	S14	S9	S7
S10	S4	S3	S13
S5	S11	S12	S2
S15	S1	S6	S8

After ShuffleCell

S0	S14	S9	S7
S10	S4	S3	S13
S5	S11	S12	S2
S15	S1	S6	S8

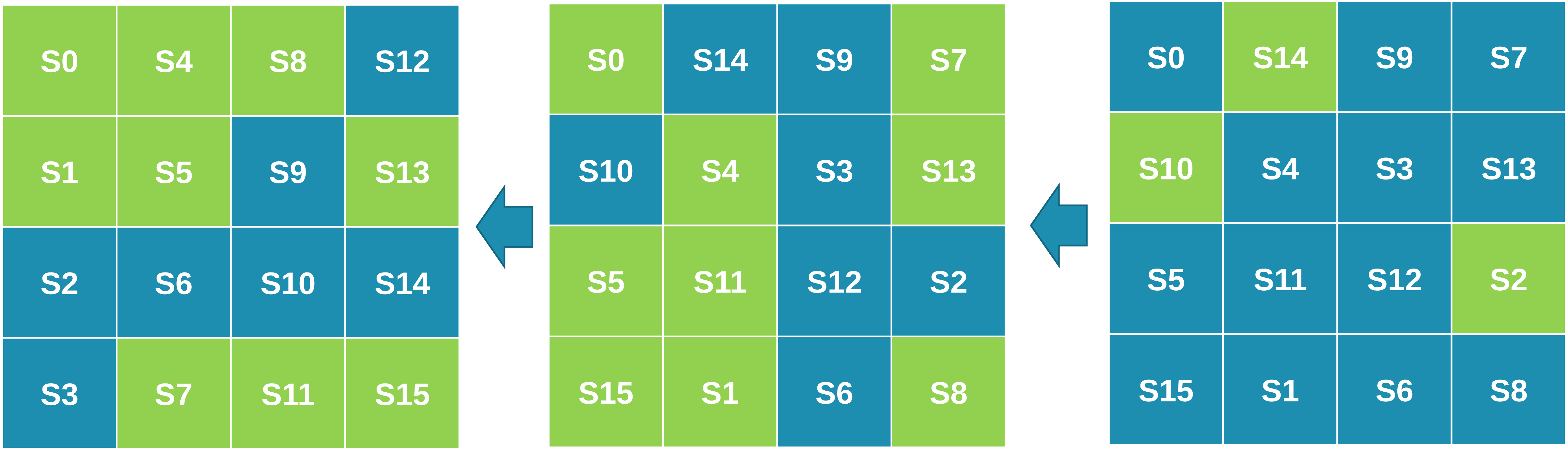
After MixColumns

Midori64 Trail – 2 Rounds



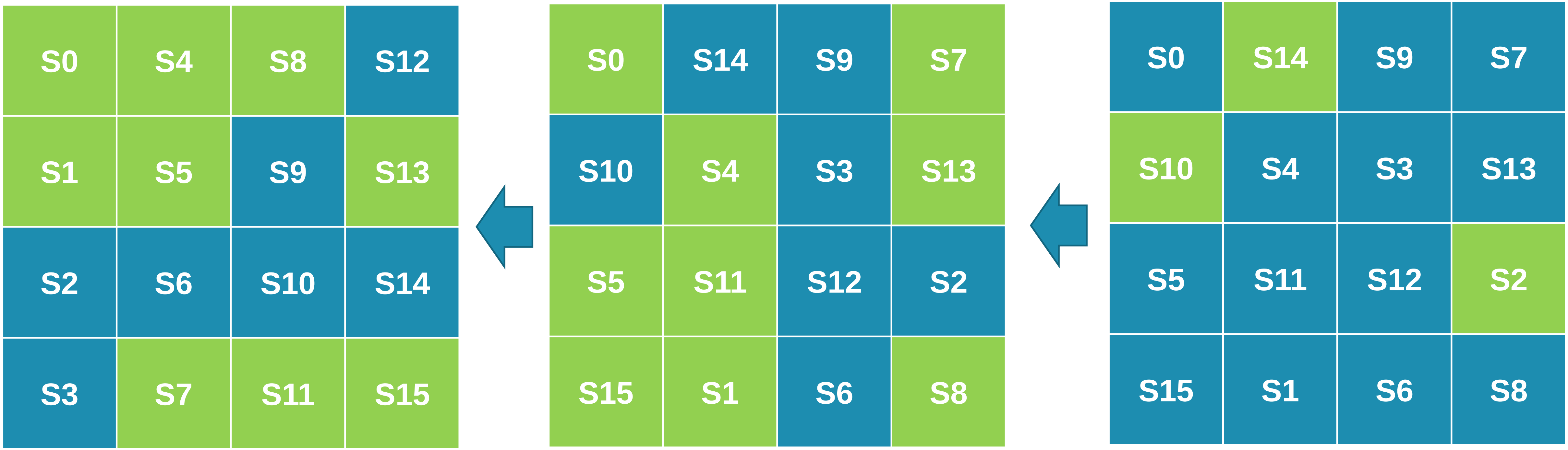
After S-box After ShuffleCell After MixColumns

Midori64 Trail – 2 Rounds



After S-box After ShuffleCell After MixColumns

Midori64 Trail – 2 Rounds



$$\varepsilon := \|\widehat{p}_{\mathbf{z}} - \delta_0\|_2^2 \leq |\operatorname{supp} \widehat{p}_{\mathbf{z}}| \, \|\widehat{p}_{\mathbf{z}} - \delta_0\|_\infty^2 \leq 2^{24}\, 2^{-48} = 2^{-24}$$

Midori64: Bound

$$\mathrm{Adv}_{\text{2-thr}}(\mathcal{A}) \leq \sqrt{\frac{q}{\lambda 2^{23}}}$$

	λ	q (Adv=1)
No noise	1	≈ 8 million
FPGA noise	$<2^9$	≈ 4 billion

Midori64: Non-uniform inputs

Insecure

$$\begin{pmatrix} r_1 & r_2 & r_3 & r_4 \\ r_2 & r_1 & r_4 & r_3 \\ r_3 & r_4 & r_1 & r_2 \\ r_4 & r_3 & r_2 & r_1 \end{pmatrix}$$

Secure

$$\begin{pmatrix} r_1 & r_1 & r_1 & r_1 \\ r_2 & r_2 & r_2 & r_2 \\ r_3 & r_3 & r_3 & r_3 \\ r_4 & r_4 & r_4 & r_4 \end{pmatrix}$$

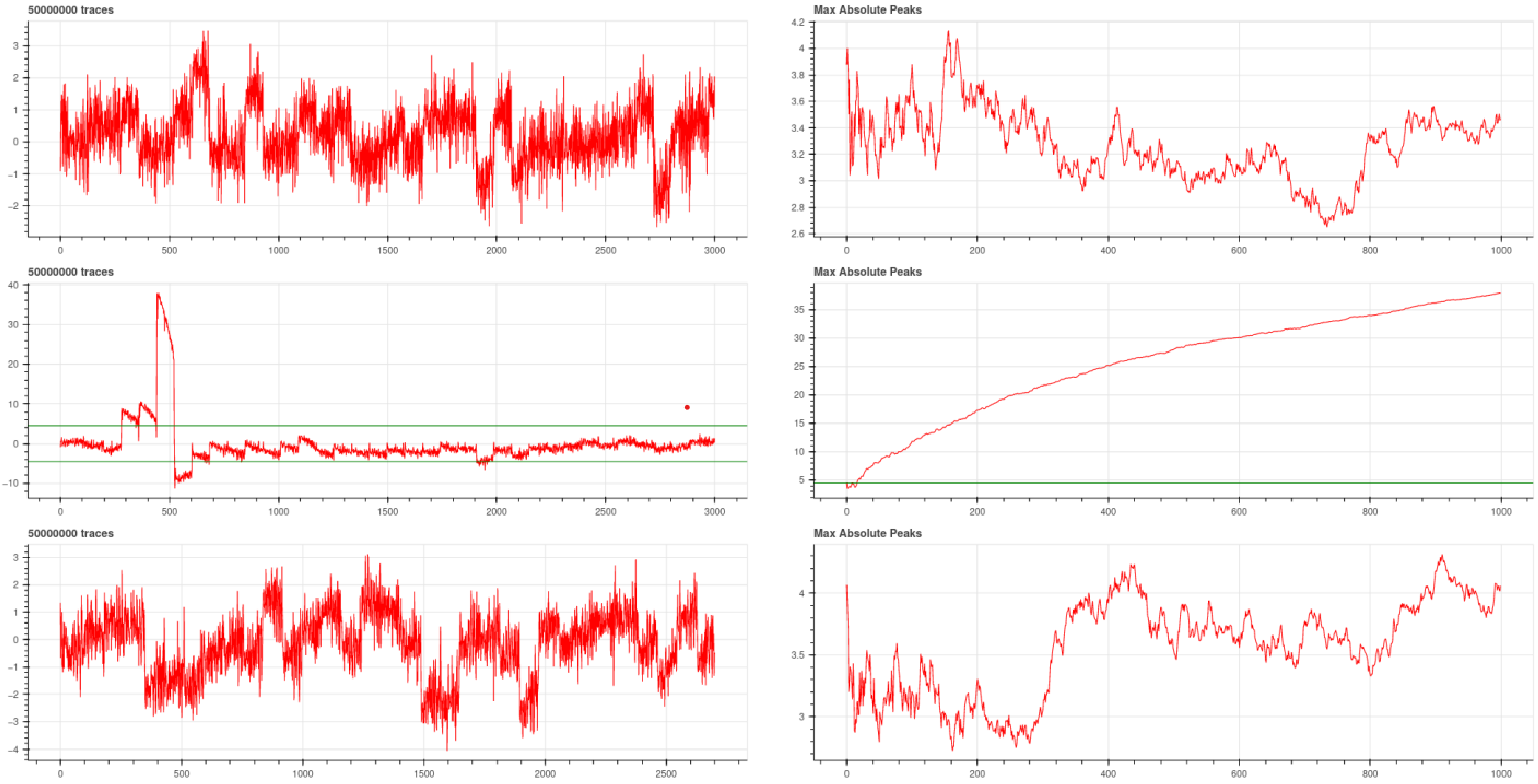
$r_1..r_4$ – random bytes (two nibbles), meaning: $r_i = r_{i1} \,||\, r_{i2}$, where r_{i1}, r_{i2} – plaintext masks to make 3 shared version.

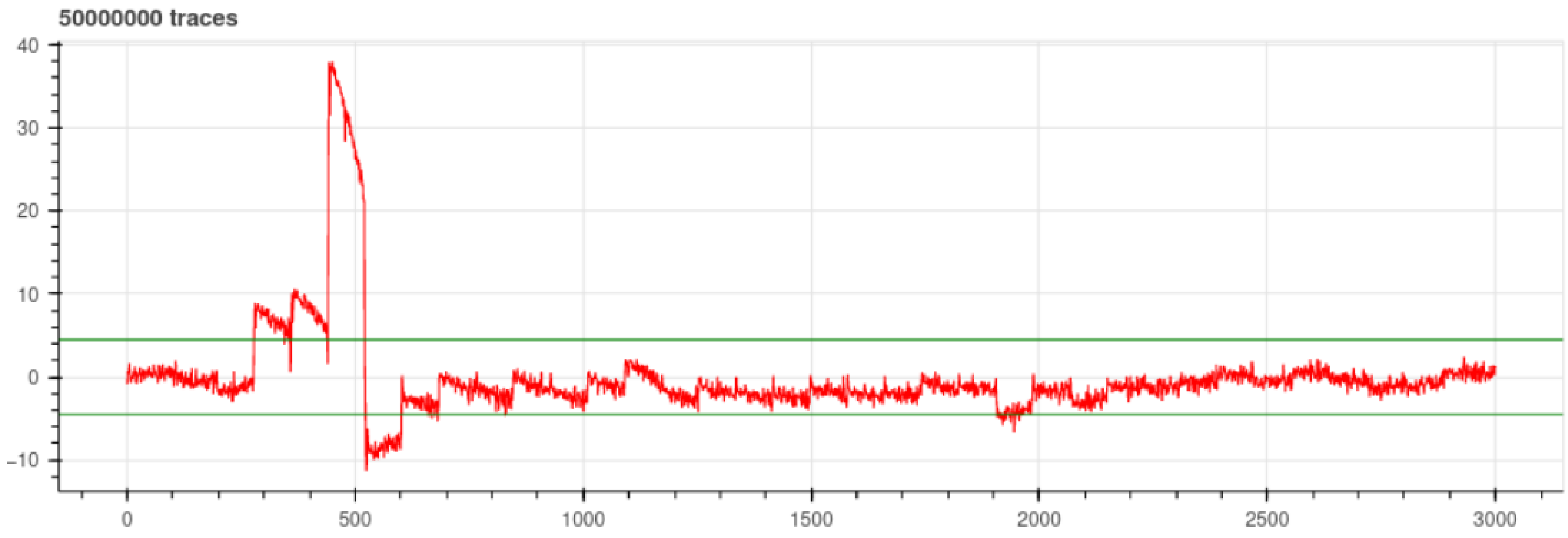
Midori64: PROLEAD tests

Cipher	Case	Mode	Passed	#Traces	#Cycle	#Round
Midori	Uniform	compact	✓	100M	NA	NA
	"Insecure" Non-Uniform	compact	✗	1M	5,6,7	2,3
		normal	✗	128k	5,6,7	2,3
	"Secure" Non-Uniform	compact	✗	2M	7	3
		normal	✗	6.4M	8	3

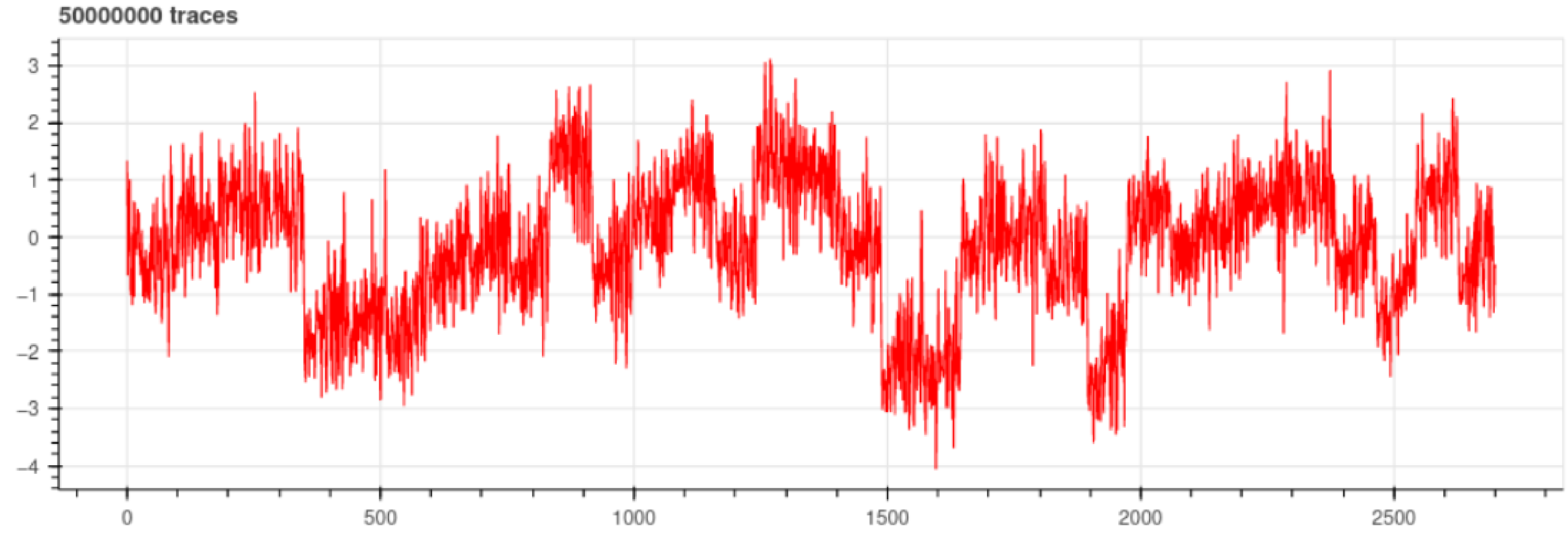
Midori64: PROLEAD tests

Cipher	Case	Mode	Passed	#Traces	#Cycle	#Round
Midori	Uniform	compact	✓	100M	NA	NA
	"Insecure" Non-Uniform	compact	✗	1M	5,6,7	2,3
		normal	✗	128k	5,6,7	2,3
	"Secure" Non-Uniform	compact	✗	2M	7	3
		normal	✗	6.4M	8	3

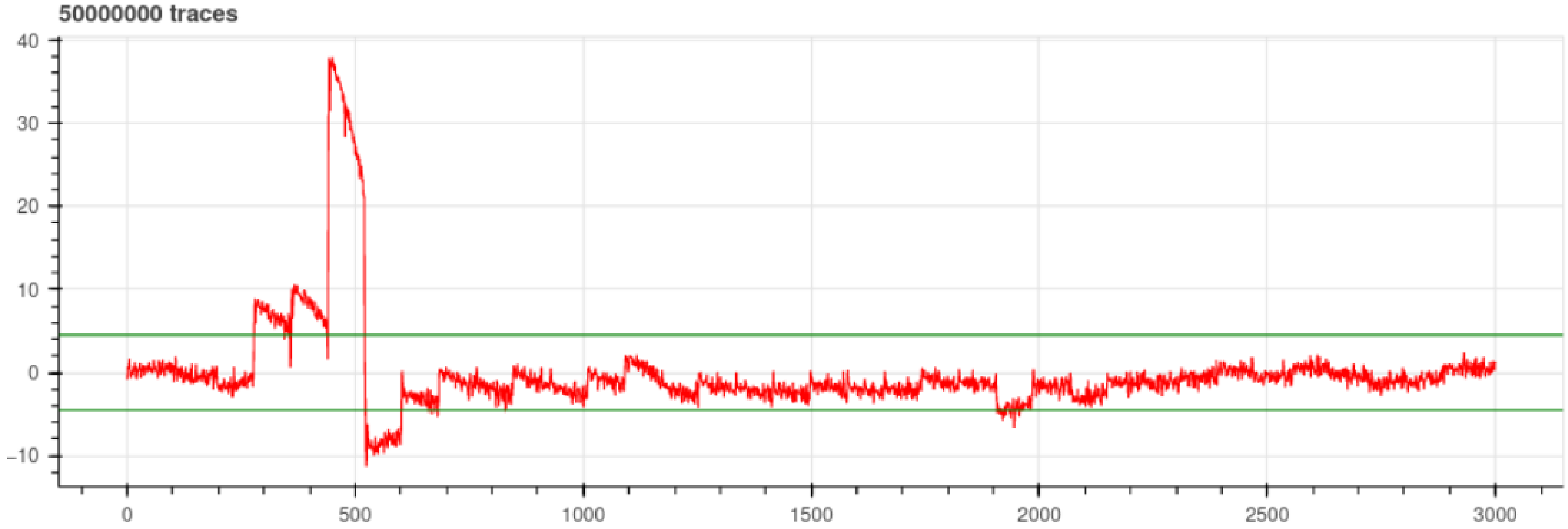

50000000 traces
Max Absolute Peaks
50000000 traces
Max Absolute Peaks
50000000 traces
Max Absolute Peaks



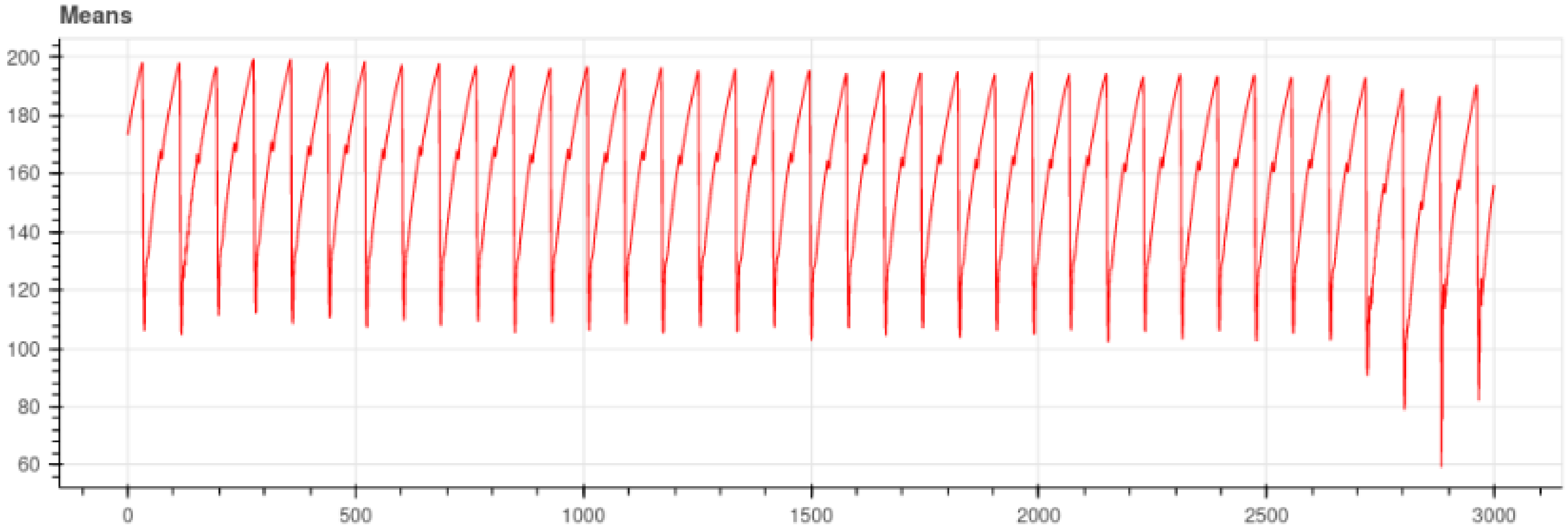
50000000 traces
40
30
20
10
0
−10
0
500
1000
1500
2000
2500
3000



50000000 traces
3
2
1
0
−1
−2
−3
−4
0
500
1000
1500
2000
2500

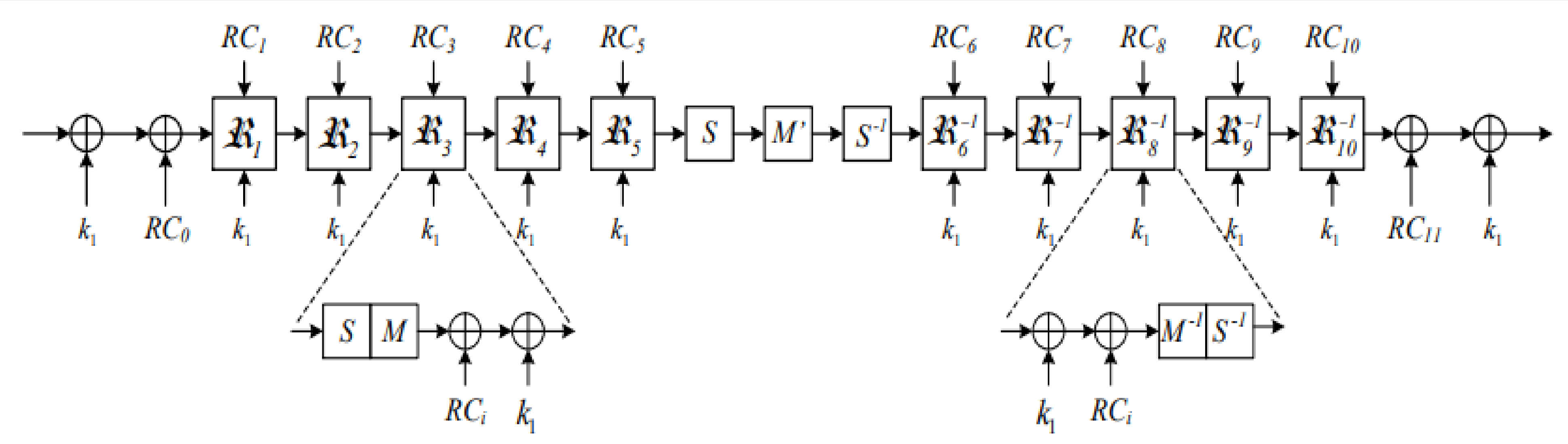


50000000 traces
40
30
20
10
0
−10
0
500
1000
1500
2000
2500
3000



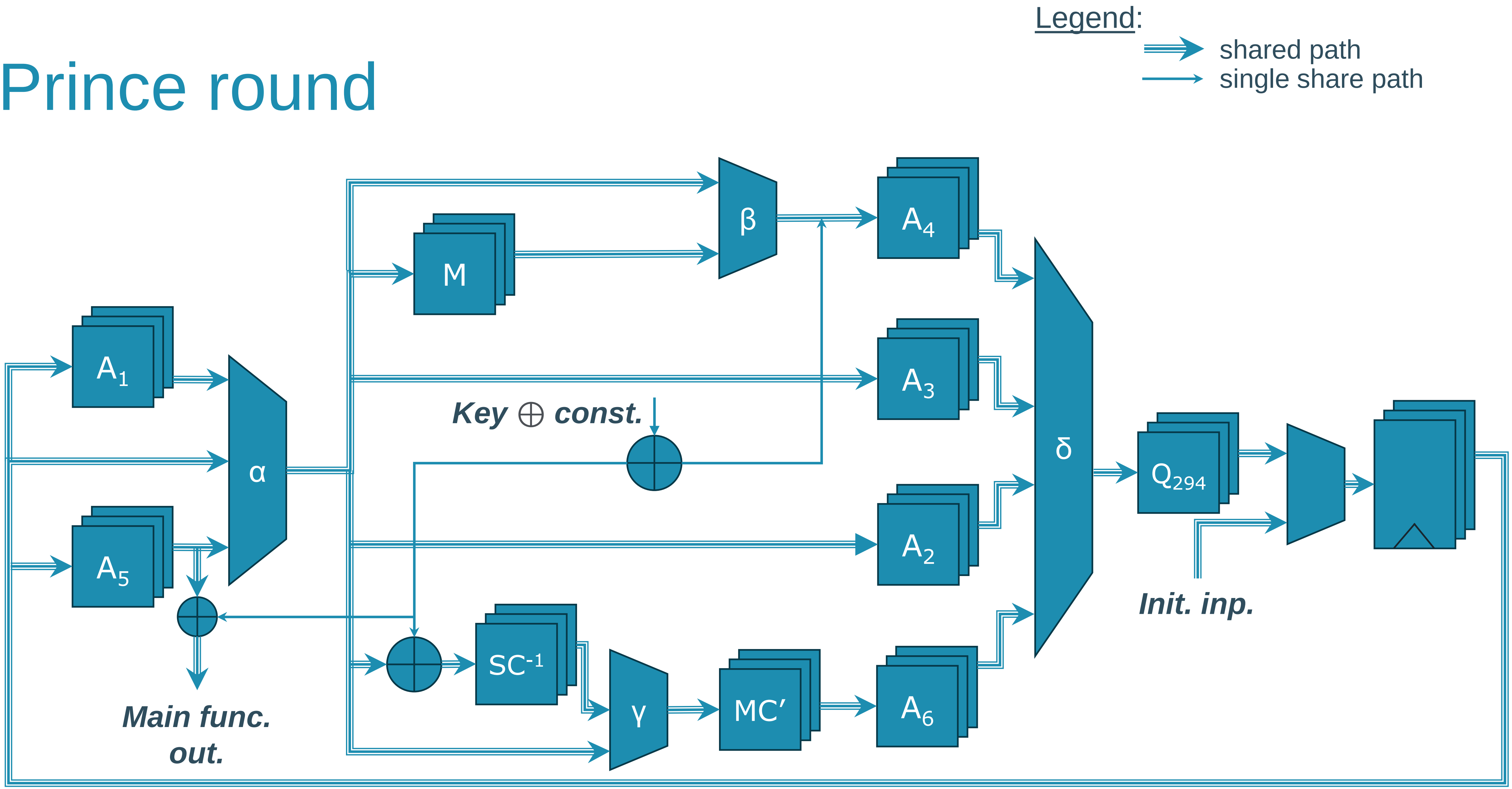
Means
200
180
160
140
120
100
80
60
0
500
1000
1500
2000
2500
3000

PRINCE



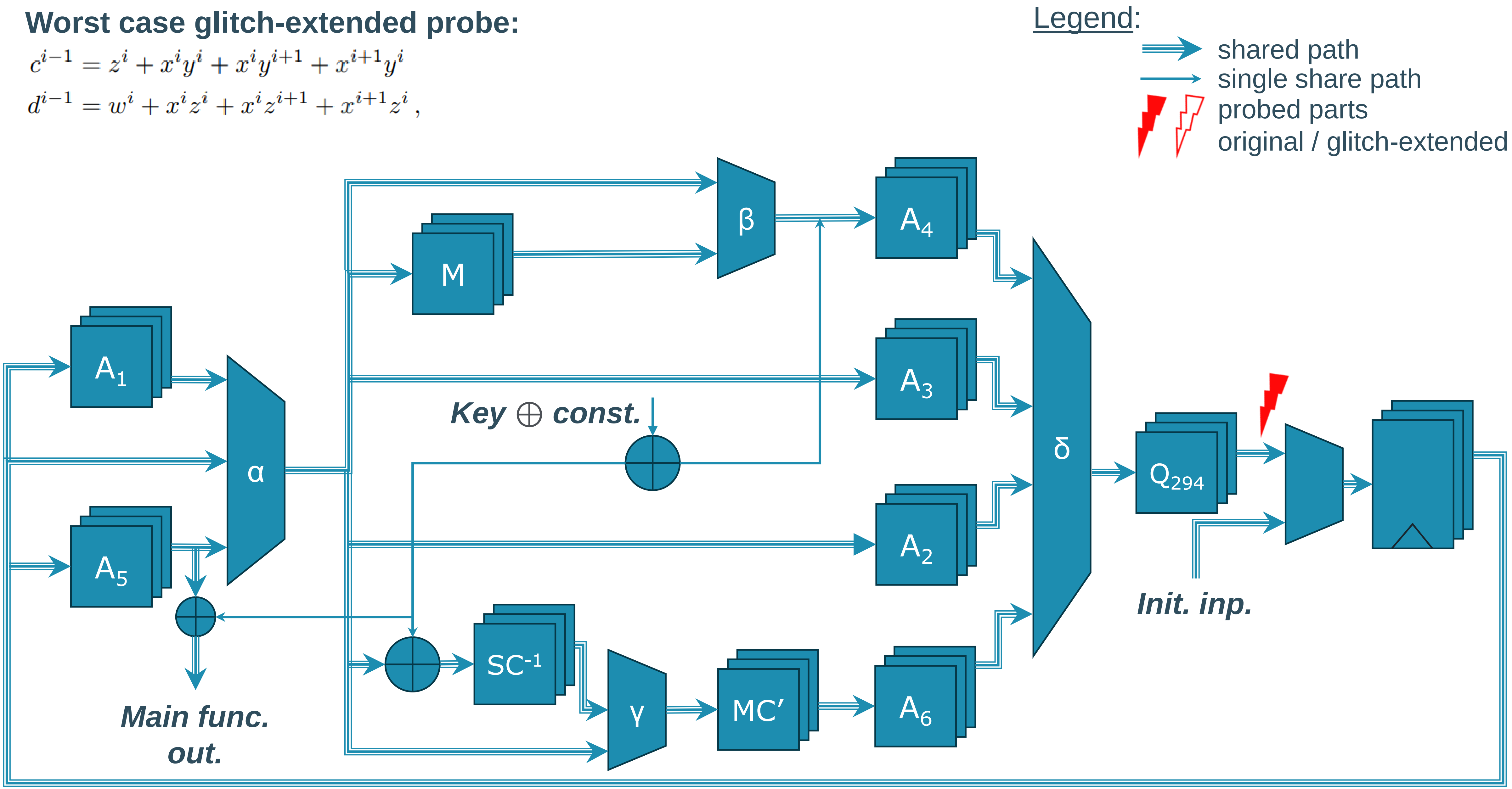
RC_1 RC_2 RC_3 RC_4 RC_5 RC_6 RC_7 RC_8 RC_9 RC_{10}
$\mathcal{R}_1$ $\mathcal{R}_2$ $\mathcal{R}_3$ $\mathcal{R}_4$ $\mathcal{R}_5$ S M' S^{-1} $\mathcal{R}_6^{-1}$ $\mathcal{R}_7^{-1}$ $\mathcal{R}_8^{-1}$ $\mathcal{R}_9^{-1}$ $\mathcal{R}_{10}^{-1}$
k_1 RC_0 k_1 k_1 k_1 k_1 k_1 k_1 k_1 k_1 k_1 k_1 RC_{11} k_1
S M
RC_i k_1
M^{-1} S^{-1}
k_1 RC_i

Prince round



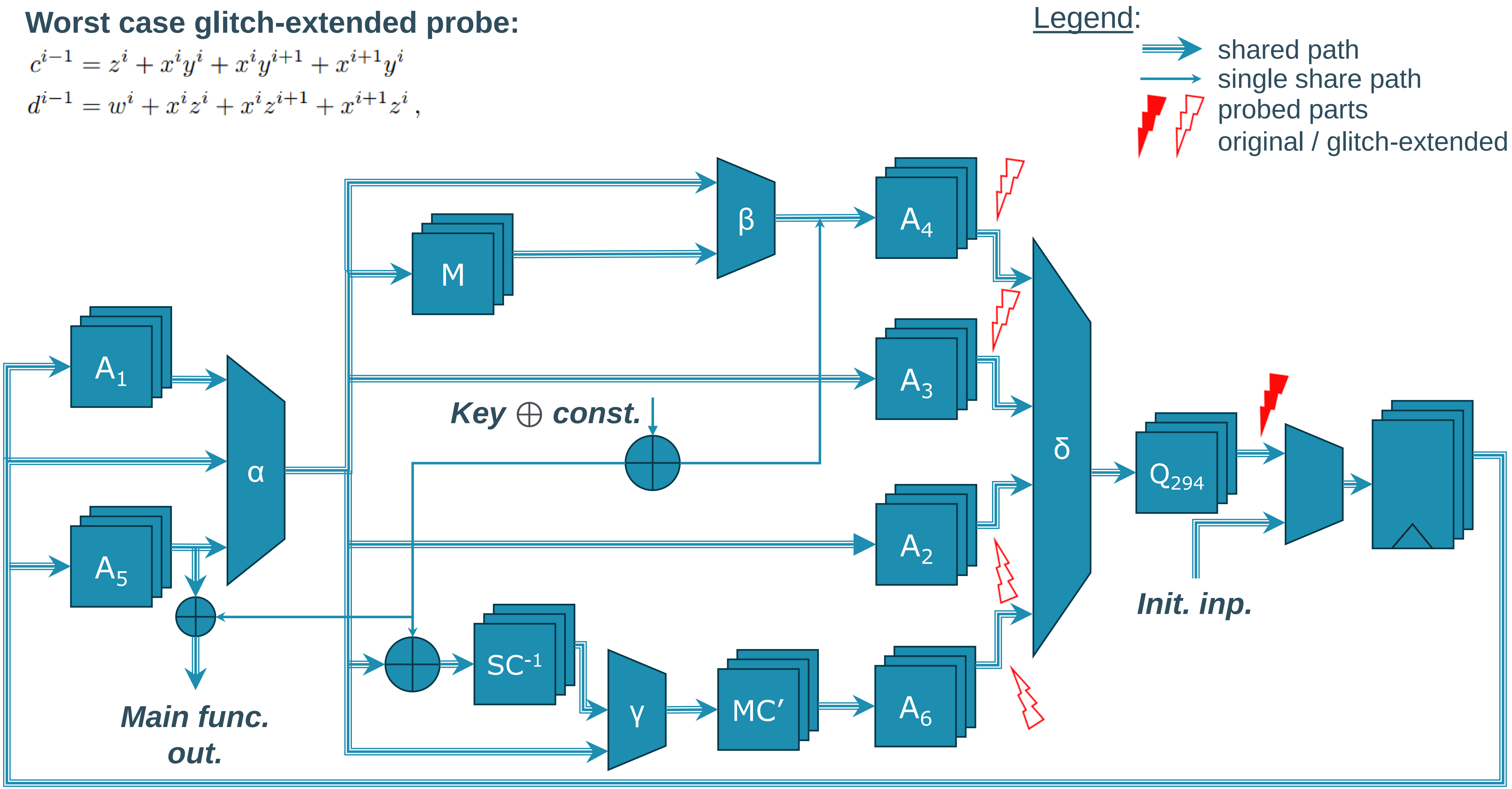
Worst case glitch-extended probe:

$$c^{i-1} = z^i + x^i y^i + x^i y^{i+1} + x^{i+1} y^i$$
$$d^{i-1} = w^i + x^i z^i + x^i z^{i+1} + x^{i+1} z^i \,,$$



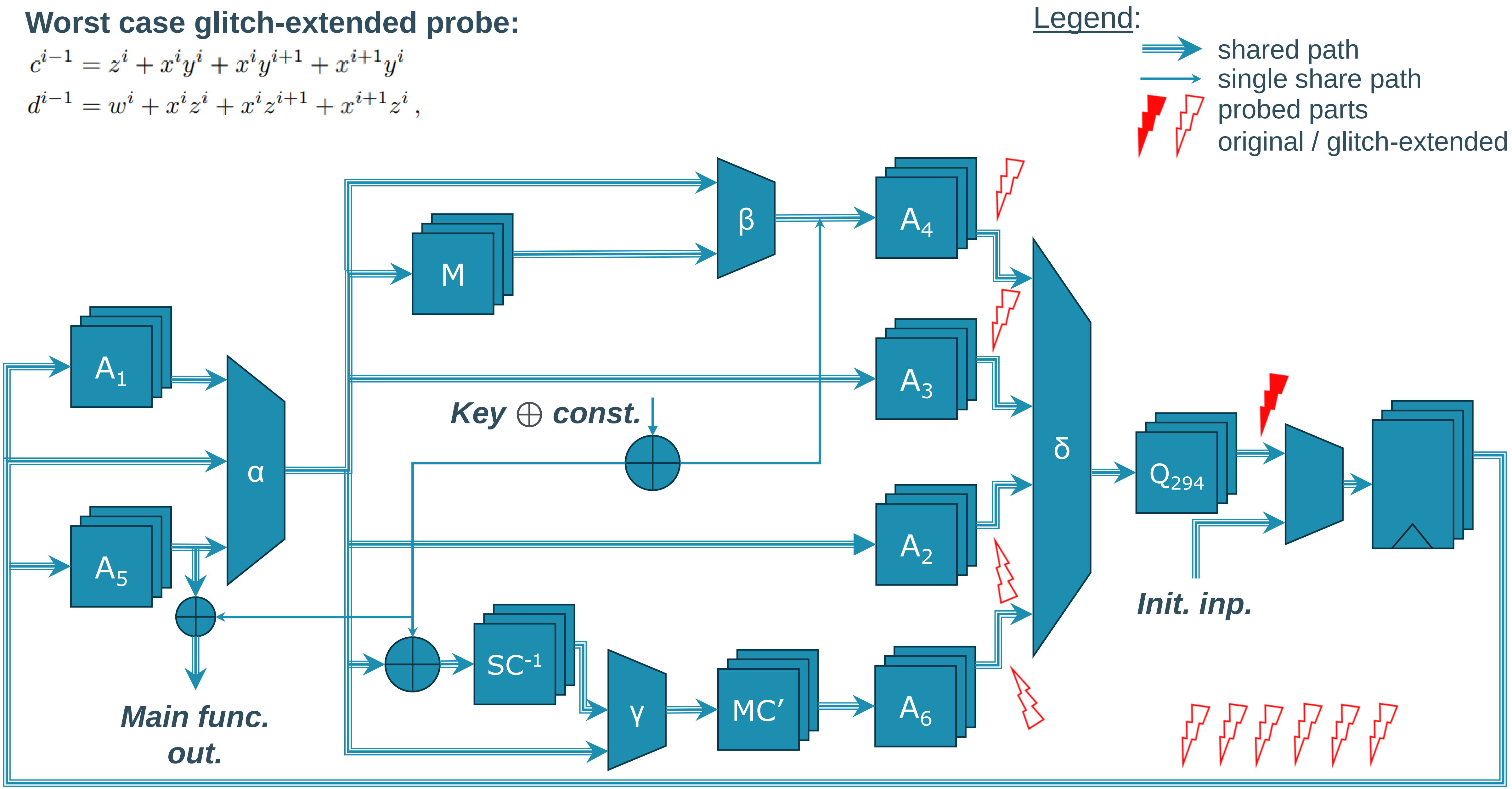
Worst case glitch-extended probe:

$$c^{i-1} = z^i + x^i y^i + x^i y^{i+1} + x^{i+1} y^i$$
$$d^{i-1} = w^i + x^i z^i + x^i z^{i+1} + x^{i+1} z^i \,,$$

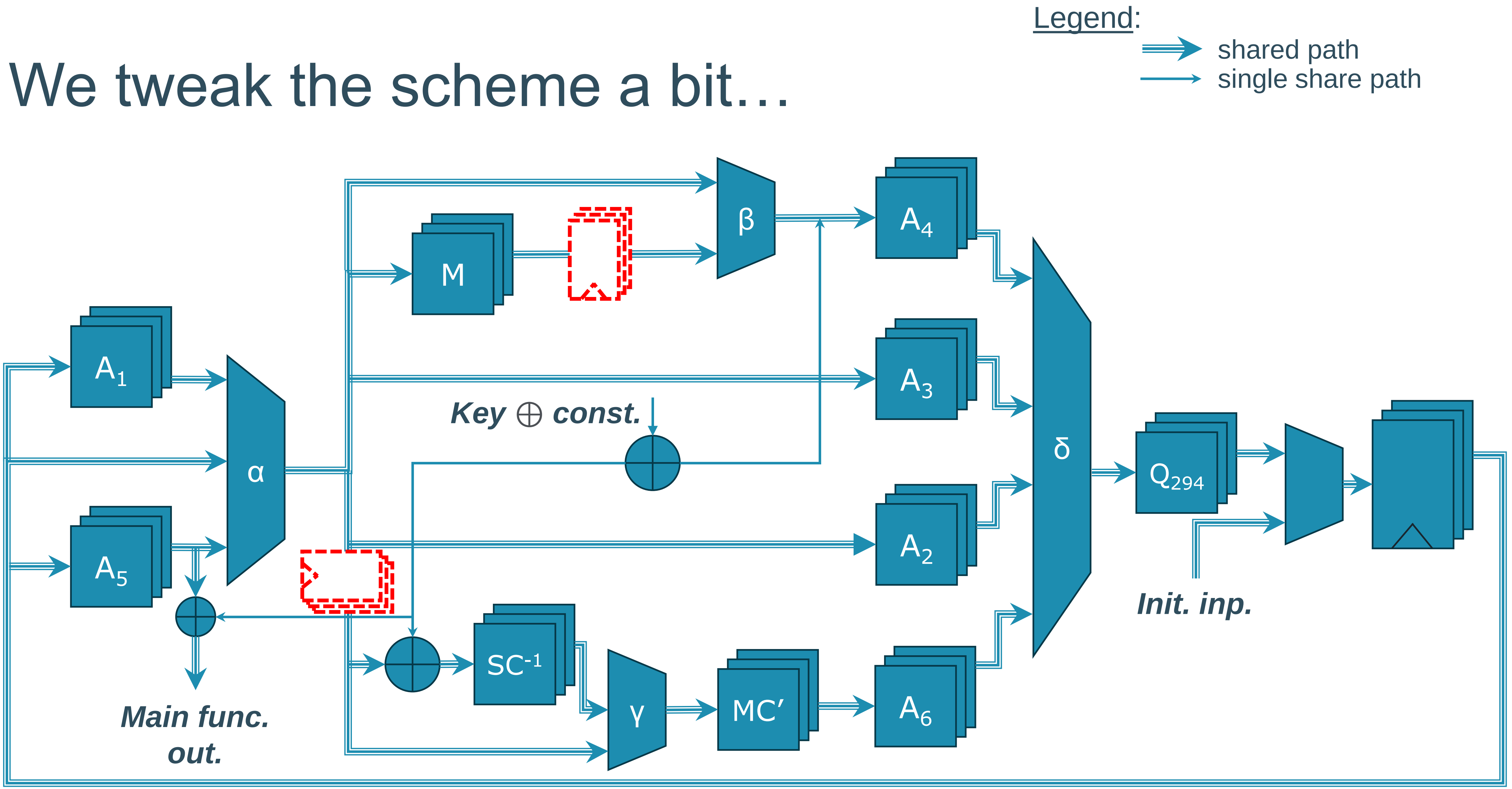


Worst case glitch-extended probe:

$$c^{i-1} = z^i + x^i y^i + x^i y^{i+1} + x^{i+1} y^i$$
$$d^{i-1} = w^i + x^i z^i + x^i z^{i+1} + x^{i+1} z^i\,,$$

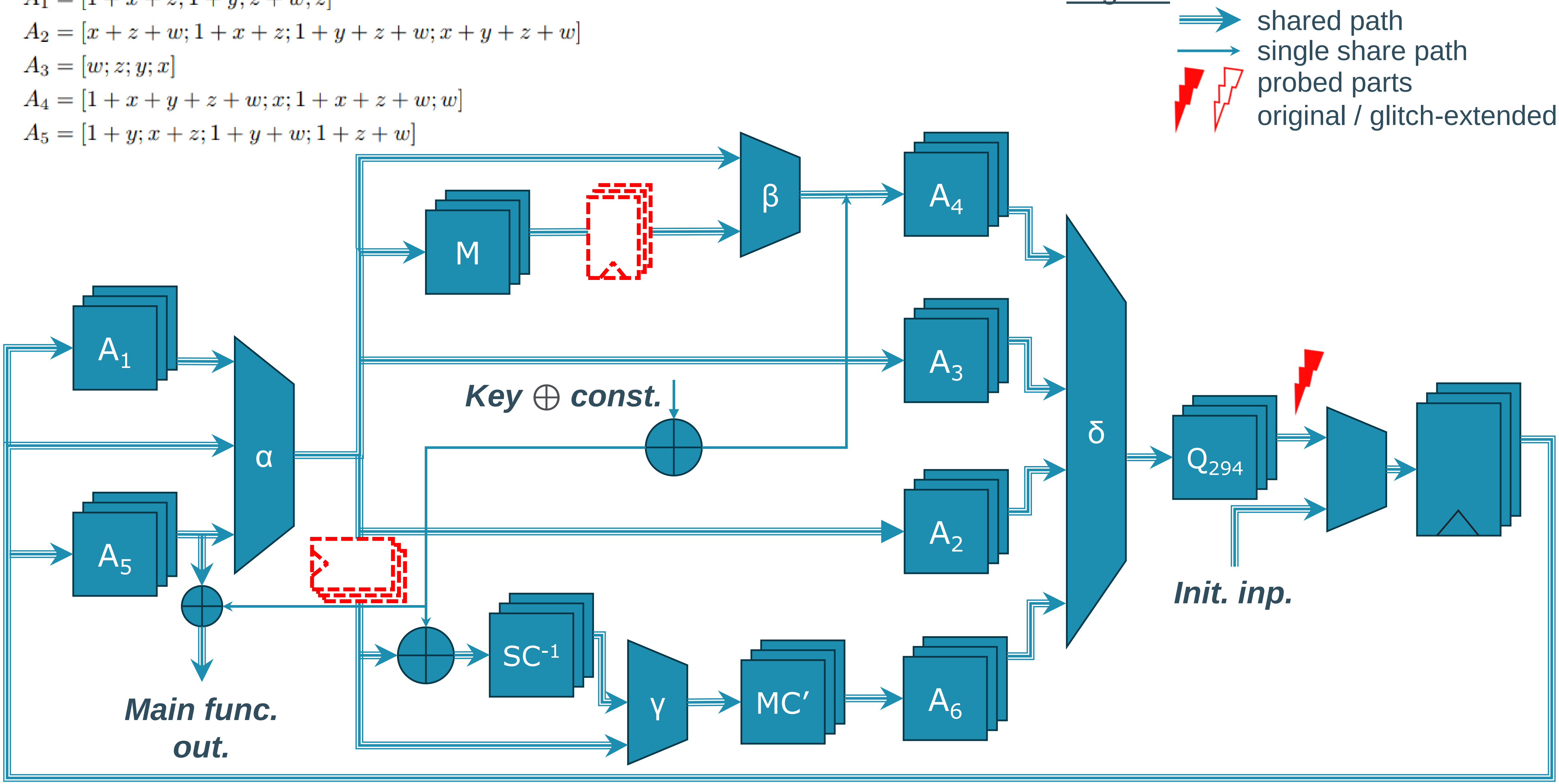


We tweak the scheme a bit…

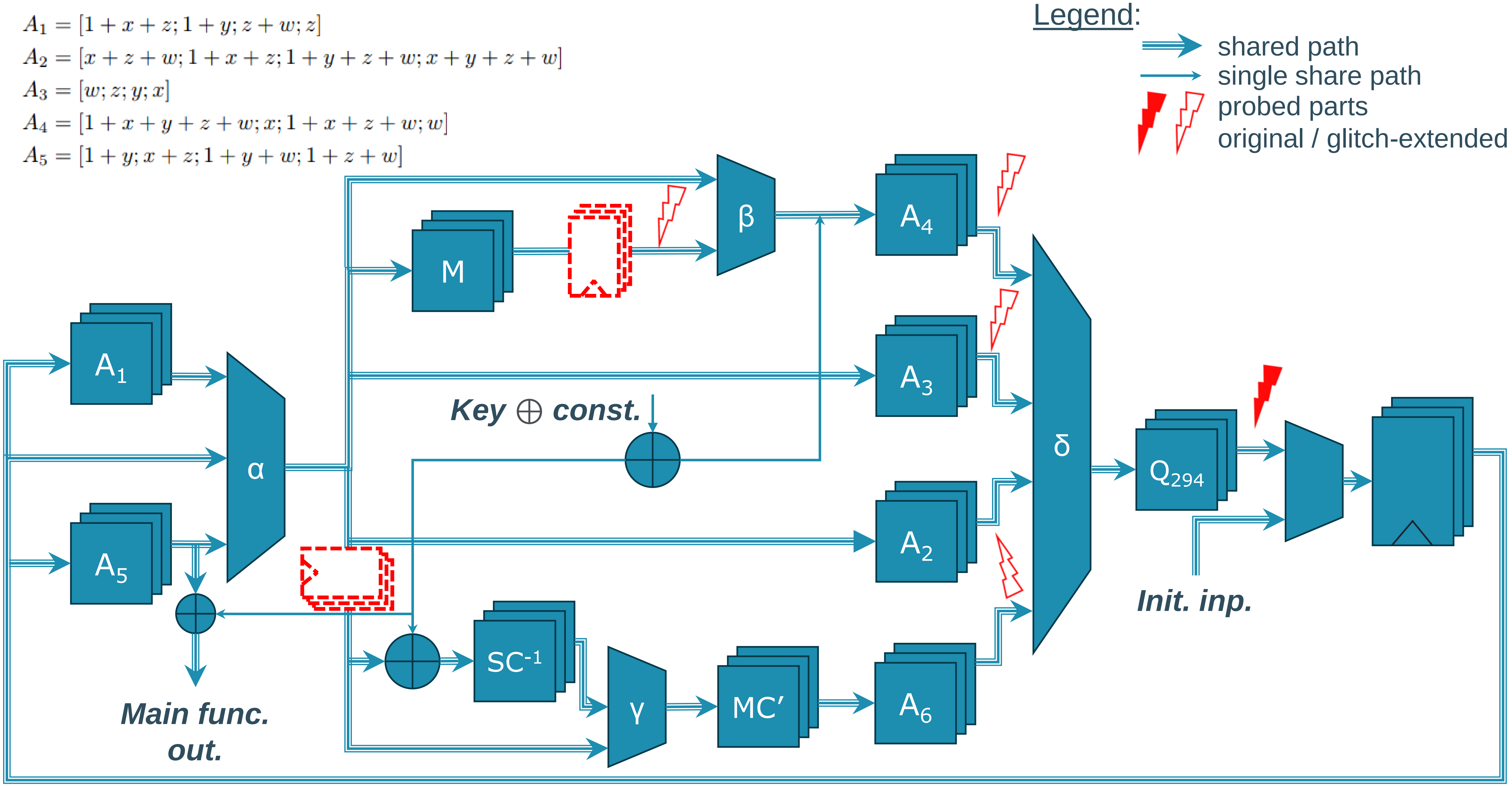


$$A_1 = [1+x+z; 1+y; z+w; z]$$
$$A_2 = [x+z+w; 1+x+z; 1+y+z+w; x+y+z+w]$$
$$A_3 = [w; z; y; x]$$
$$A_4 = [1+x+y+z+w; x; 1+x+z+w; w]$$
$$A_5 = [1+y; x+z; 1+y+w; 1+z+w]$$

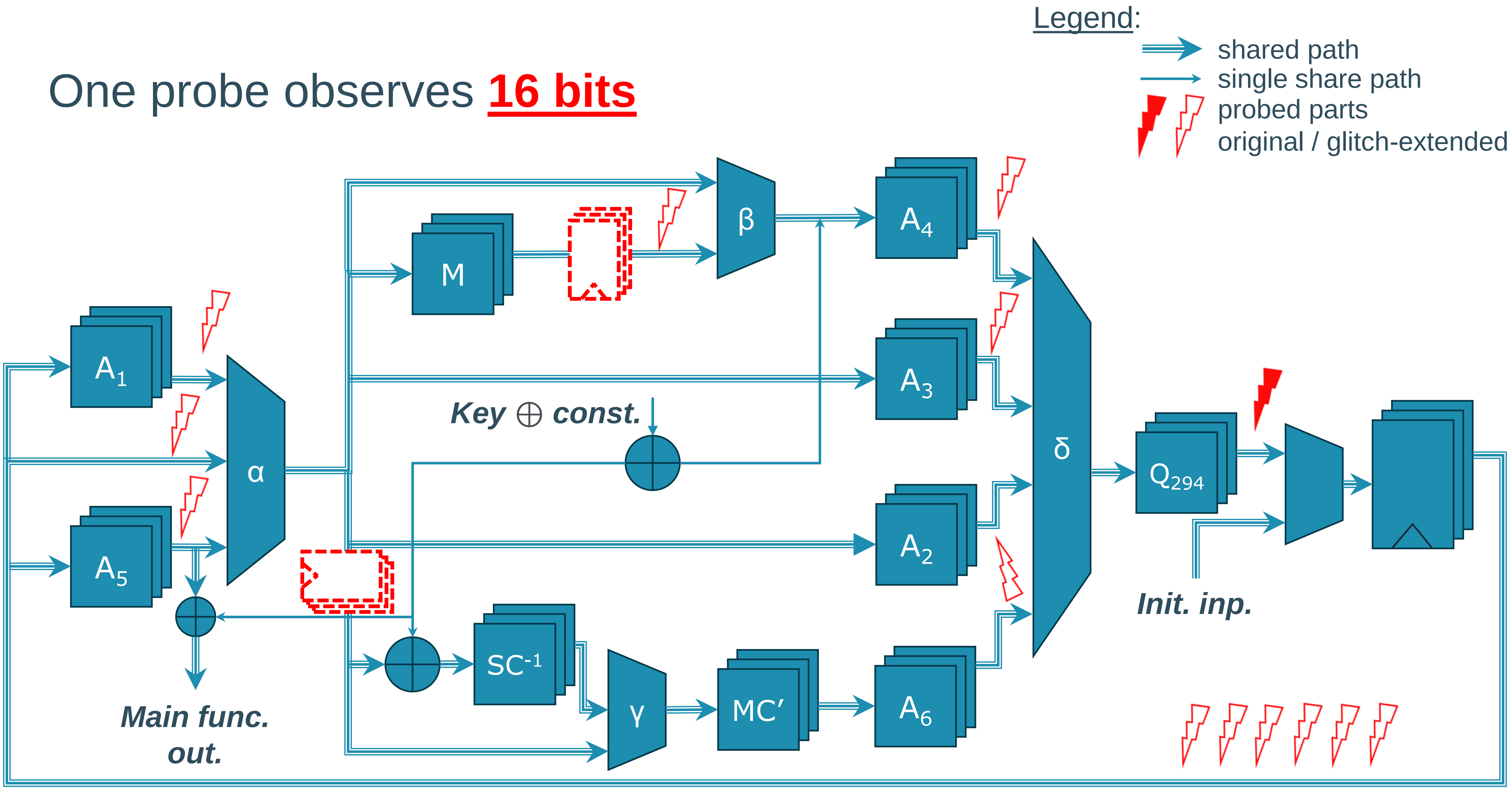
Legend:



$$A_1 = [1+x+z; 1+y; z+w; z]$$
$$A_2 = [x+z+w; 1+x+z; 1+y+z+w; x+y+z+w]$$
$$A_3 = [w; z; y; x]$$
$$A_4 = [1+x+y+z+w; x; 1+x+z+w; w]$$
$$A_5 = [1+y; x+z; 1+y+w; 1+z+w]$$



One probe observes **16 bits**



PRINCE Trail – 1 Round

S0	S4	S8	S12
S1	S5	S9	S13
S2	S6	S10	S14
S3	S7	S11	S15

After S-box

S0	S4	S8	S12
S1	S5	S9	S13
S2	S6	S10	S14
S3	S7	S11	S15

After MixColumns

S0	S4	S8	S12
S5	S9	S13	S1
S10	S14	S2	S6
S15	S3	S7	S11

After ShiftRows

PRINCE Trail – 1 Round

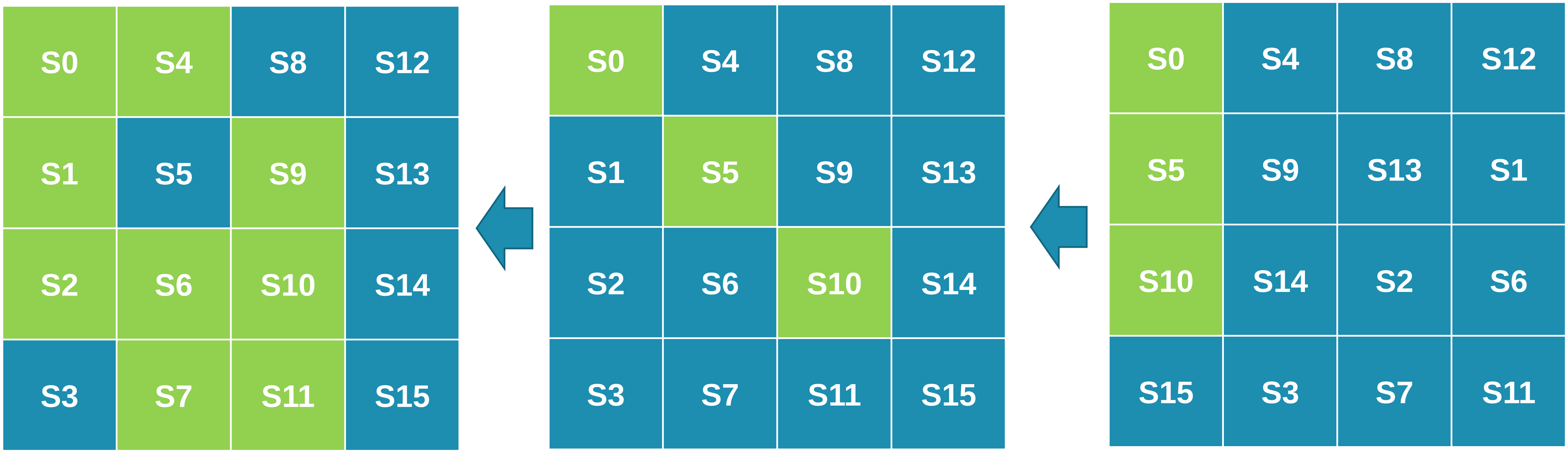
S0	S4	S8	S12
S1	S5	S9	S13
S2	S6	S10	S14
S3	S7	S11	S15

S0	S4	S8	S12
S1	S5	S9	S13
S2	S6	S10	S14
S3	S7	S11	S15

S0	S4	S8	S12
S5	S9	S13	S1
S10	S14	S2	S6
S15	S3	S7	S11

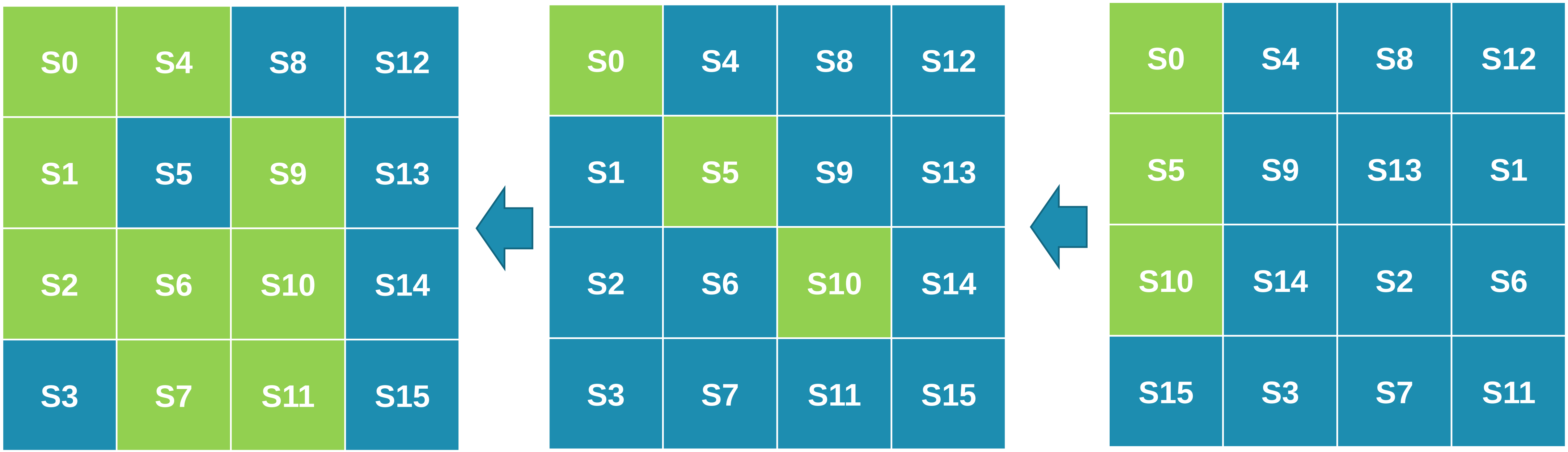
$$\varepsilon := \|\widehat{p}_{\mathbf{z}} - \delta_0\|_2^2 \leq |\mathrm{supp}\,\widehat{p}_{\mathbf{z}}|\,\|\widehat{p}_{\mathbf{z}} - \delta_0\|_\infty^2 \leq 2^{(\#wires)^{\#probes}} \left|C_{u,v}^{S}\right|^{(active\_cells)^2} >> 1$$

PRINCE Trail – 2 Rounds



After S-box After MixColumns After ShiftRows

PRINCE Trail – 2 Rounds



$$\varepsilon := \|\widehat{p}_{\mathbf{z}} - \delta_0\|_2^2 \le |\operatorname{supp}\widehat{p}_{\mathbf{z}}|\,\|\widehat{p}_{\mathbf{z}} - \delta_0\|_\infty^2 \le 2^{16}\,2^{-33.84} = 2^{-17.84}$$

PRINCE: Bound

$$\mathrm{Adv}_{\text{2-thr}}(\mathcal{A}) \leq \sqrt{\frac{q}{\lambda 2^{16.84}}}$$

	λ	q (Adv=1)
No noise	1	≈ 131k
FPGA noise	$<2^9$	≈ 67 million

PRINCE: Non-uniform inputs

Insecure

$$\begin{pmatrix} r_1 & r_2 & r_3 & r_4 \\ r_1 & r_2 & r_3 & r_4 \\ r_1 & r_2 & r_3 & r_4 \\ r_1 & r_2 & r_3 & r_4 \end{pmatrix}$$

Secure

$$\begin{pmatrix} r_1 & r_1 & r_1 & r_1 \\ r_2 & r_2 & r_2 & r_2 \\ r_3 & r_3 & r_3 & r_3 \\ r_4 & r_4 & r_4 & r_4 \end{pmatrix}$$

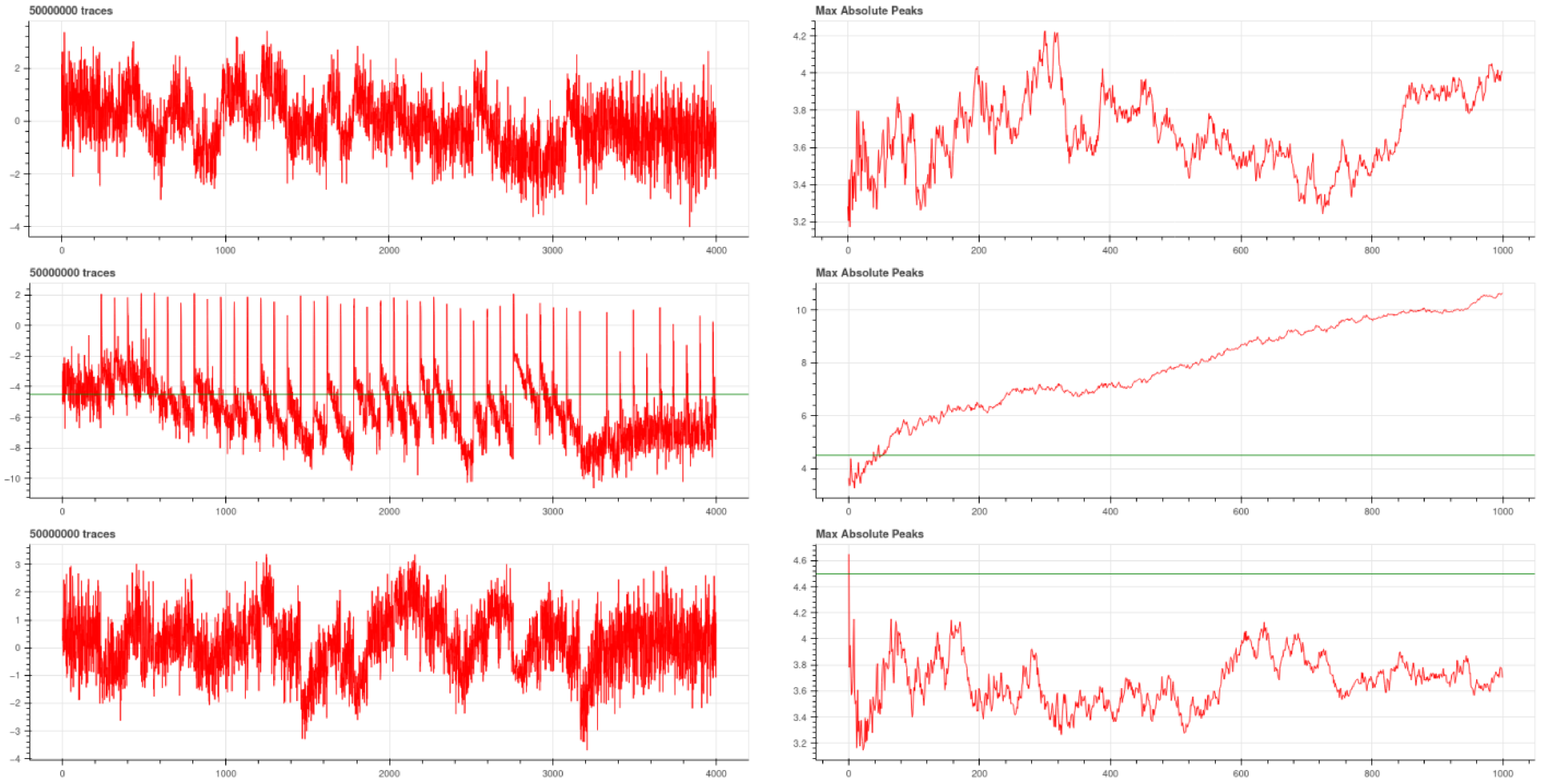
$r_1..r_4$ – random bytes (two nibbles), meaning: $r_i = r_{i1} \,||\, r_{i2}$, where r_{i1}, r_{i2} – plaintext masks to make 3 shared version.

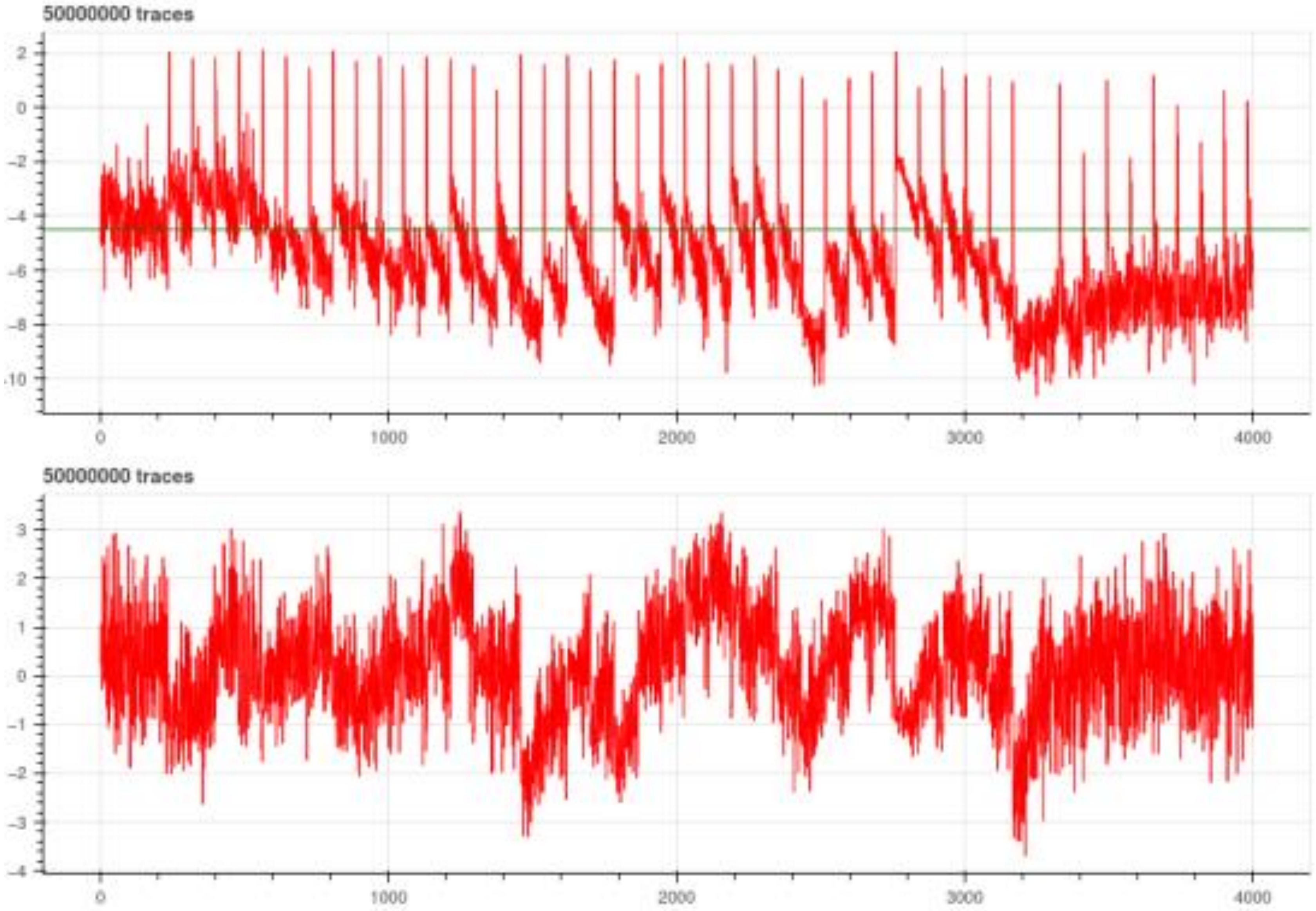
PRINCE: PROLEAD tests

Cipher	Case	Mode	Passed	#Traces	#Cycle	#Round
Prince	Uniform	compact	✓	100M	NA	NA
	"Insecure" Non-Uniform	compact	✗	1M	4,...,10	1,2,3
		normal	✗	128k	4,...,10	1,2,3
	"Secure" Non-Uniform	compact	✗	48M	10	3
		normal	✗	3.8M	10	3

PRINCE: PROLEAD tests

Cipher	Case	Mode	Passed	#Traces	#Cycle	#Round
Prince	Uniform	compact	✓	100M	NA	NA
	"Insecure" Non-Uniform	compact	✗	1M	4,...,10	1,2,3
		normal	✗	128k	4,...,10	1,2,3
	"Secure" Non-Uniform	compact	✗	48M	10	3
		normal	✗	3.8M	10	3


50000000 traces
Max Absolute Peaks
50000000 traces
Max Absolute Peaks
50000000 traces
Max Absolute Peaks


50000000 traces
50000000 traces

Conclusions and future work

- Possible to reduce the randomness providing reasonable security

	Midori64	Prince
#Shares	3	3
State size	64	64
Random. bits	32 (-75%)	32 (-75%)
Latency	32	48 (+33%*)
Area (GE)	7324	11050 (+32%*)

** Applicable to PROLEAD tests only, FPGA test are passed without additional overhead*

Conclusions and future work

- Possible to reduce the randomness providing reasonable security
- Not only the randomness entropy is important, but also its placement

	Midori64	Prince
#Shares	3	3
State size	64	64
Random. bits	32 (-75%)	32 (-75%)
Latency	32	48 (+33%*)
Area (GE)	7324	11050 (+32%*)

** Applicable to PROLEAD tests only, FPGA test are passed without additional overhead*

Conclusions and future work

- Possible to reduce the randomness providing reasonable security
- Not only the randomness entropy is important, but also its placement
- Depends on the algorithm structure and its hardware implementation

	Midori64	**Prince**
#Shares	3	3
State size	64	64
Random. bits	32 (-75%)	32 (-75%)
Latency	32	48 (+33%*)
Area (GE)	7324	11050 (+32%*)

** Applicable to PROLEAD tests only, FPGA test are passed without additional overhead*

Conclusions and future work

Things to work on in the future:

- Cheaper PRNGs since the randomness may be non-uniform
- Other algorithms
- Higher-order security

	Midori64	Prince
#Shares	3	3
State size	64	64
Random. bits	32 (-75%)	32 (-75%)
Latency	32	48 (+33%*)
Area (GE)	7324	11050 (+32%*)

** Applicable to PROLEAD tests only, FPGA test are passed without additional overhead*

Thank you for your attention!

Siemen Dhooghe (COSIC, KU Leuven):
siemen.dhooghe@esat.kuleuven.be

Artemii Ovchinnikov (COSIC, KU Leuven):
artemii.ovchinnikov@esat.kuleuven.be