Physical Attacks against Symmetric Cryptography Standards

Viet-Sang Nguyen

PhD defense

supervised by Vincent Grosso and Pierre-Louis Cayrel

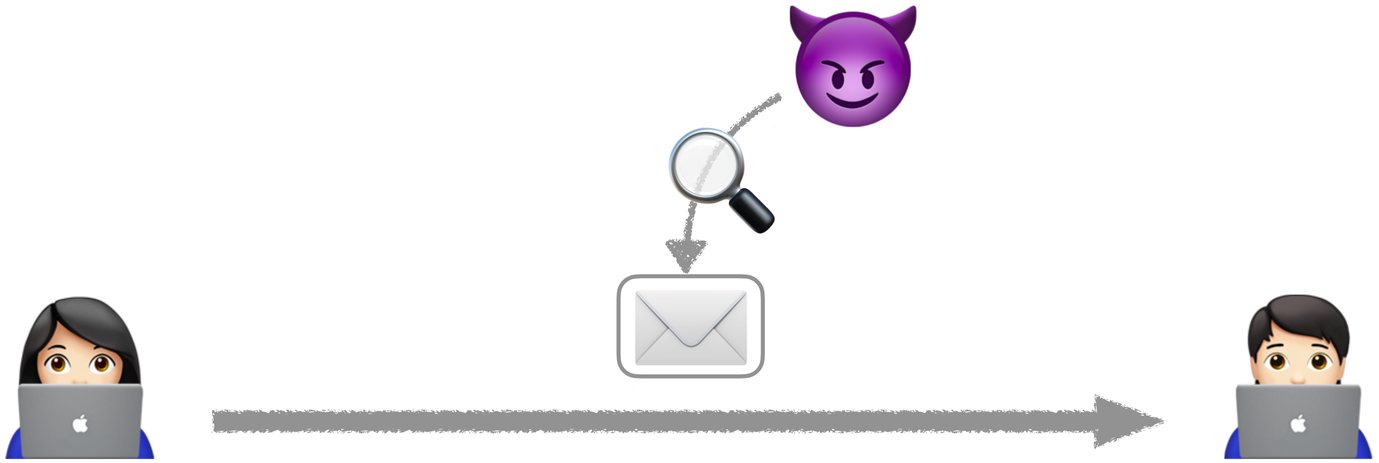
Saint-Étienne, 19 December 2025

Université de Lyon
Université Jean Monnet Saint-Étienne
Laboratoire Hubert Curien
UMR • CNRS • 5516 • Saint-Étienne
anr
PROPHY ANR-22-CE39-0008-01

Introduction

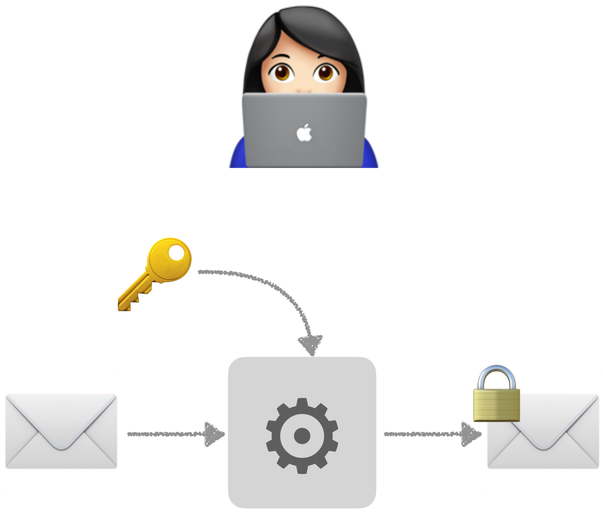
Cryptography

Cryptography

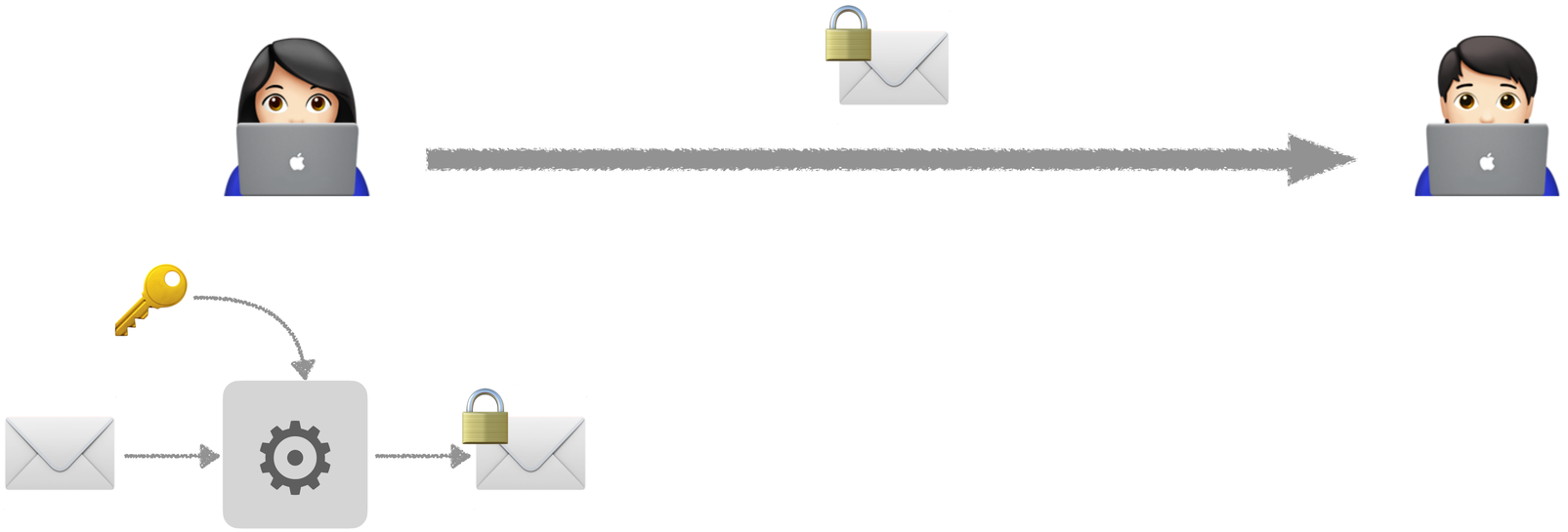


Cryptography

Cryptography

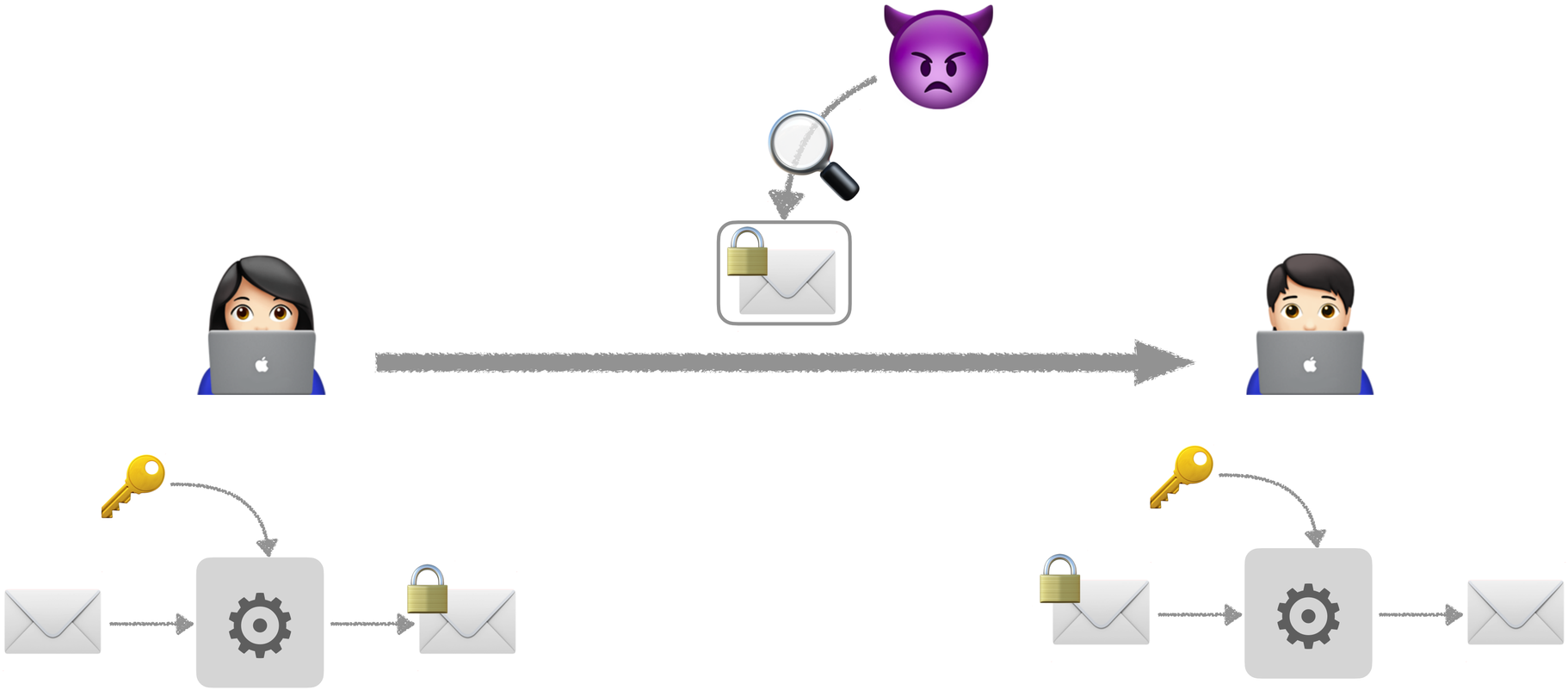


Cryptography

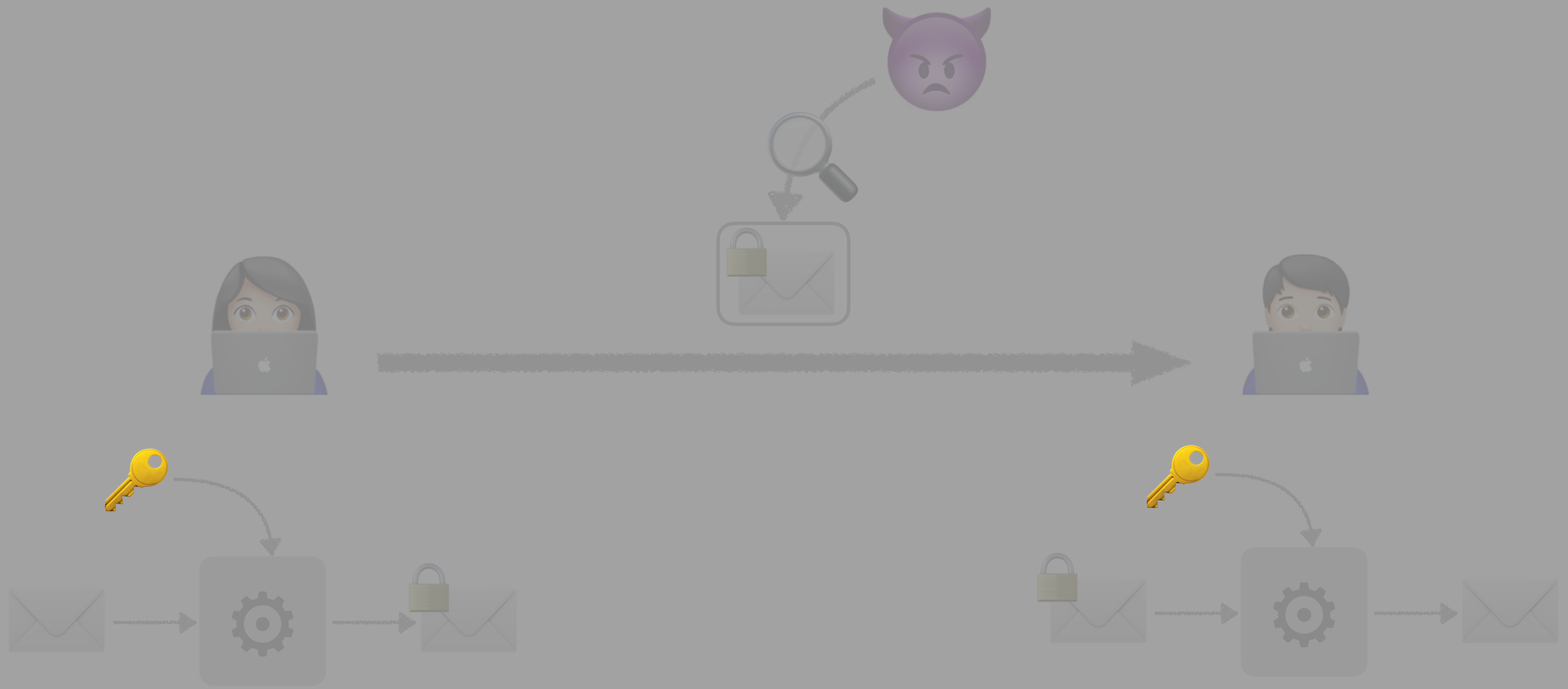


Cryptography

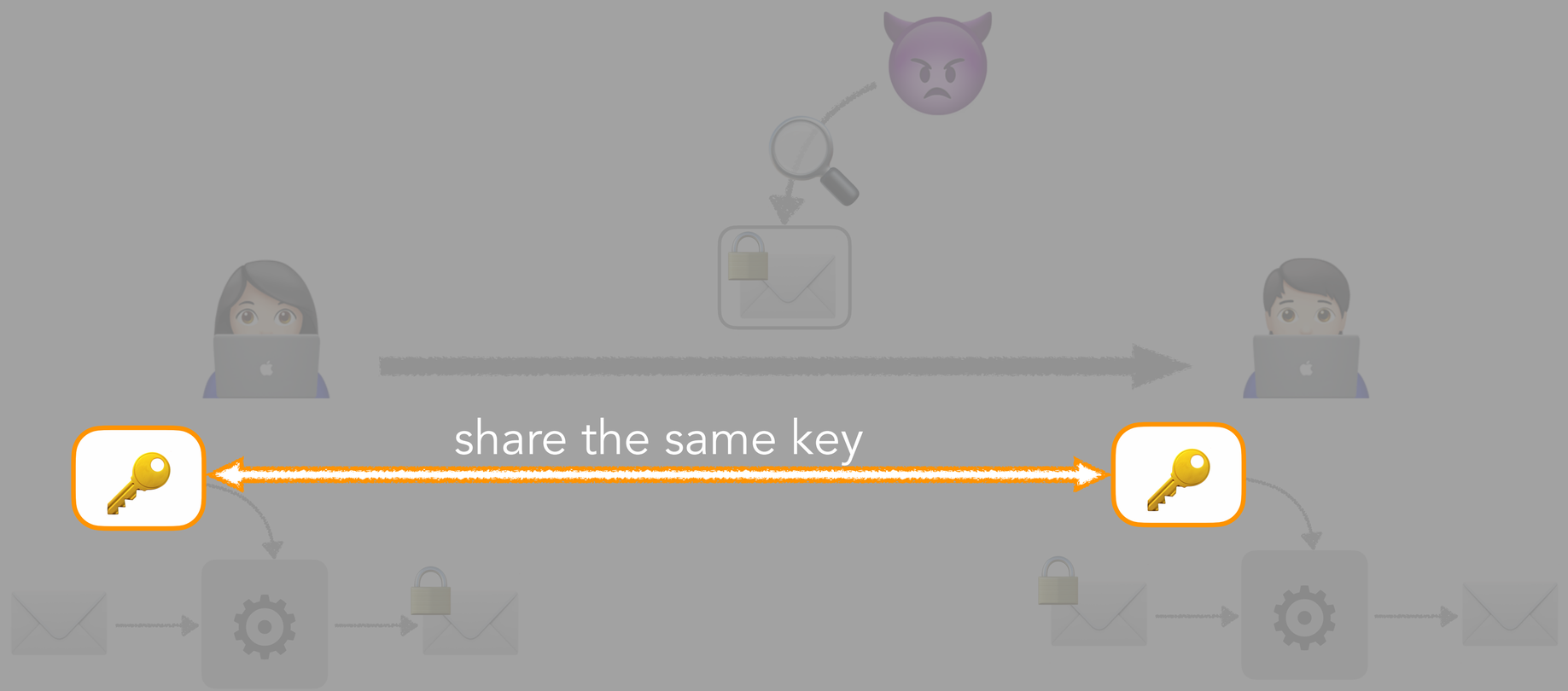
Cryptography



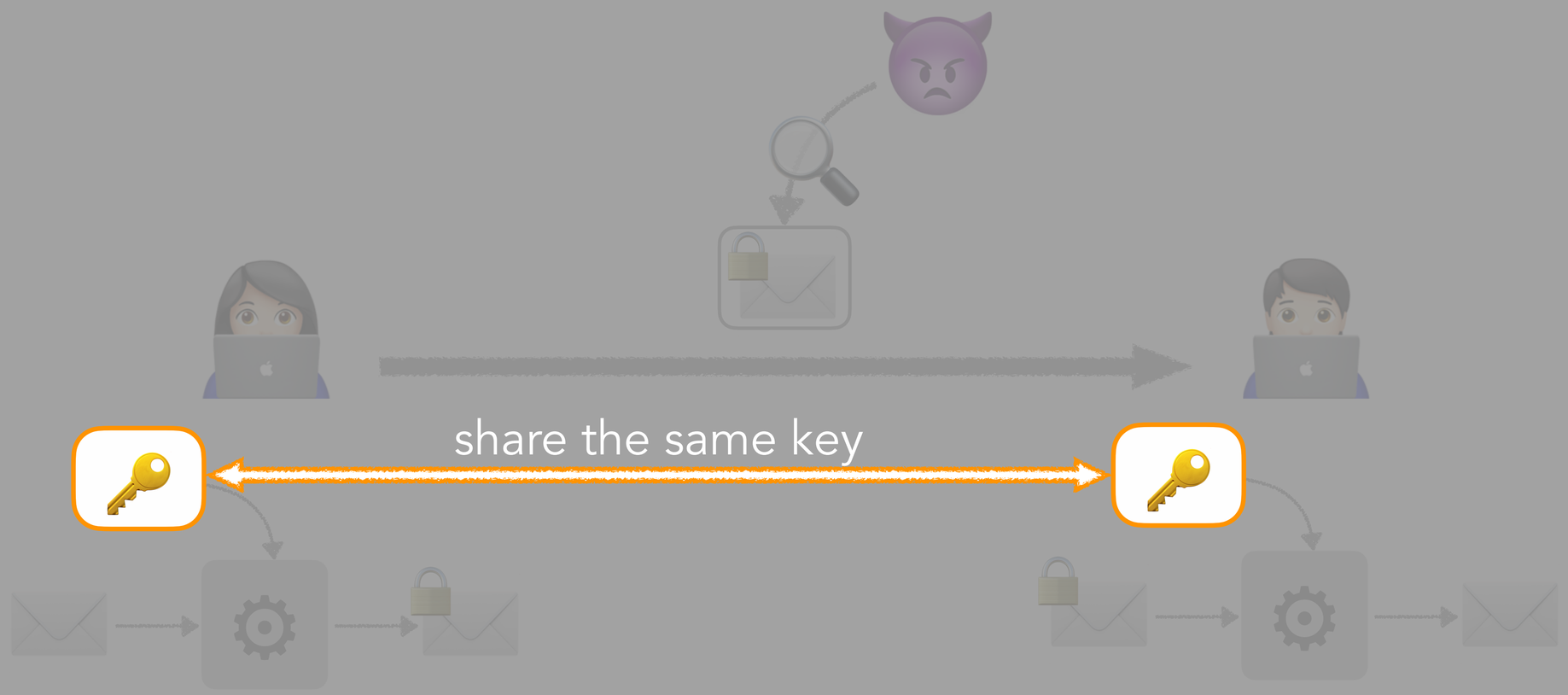
Cryptography



Cryptography



Cryptography



→ Symmetric cryptography

Symmetric cryptography standards

Symmetric cryptography standards

AES

Advanced Encryption Standard

(standardized officially in 2001)

Symmetric cryptography standards

AES

Advanced Encryption Standard

(standardized officially in 2001)

Ascon-AEAD

Ascon-based Authenticated Encryption with Associated Data

(standardized officially in 2025, selected in 2023)

Symmetric cryptography standards

AES

Advanced Encryption Standard

(standardized officially in 2001)

Ascon-AEAD

Ascon-based Authenticated Encryption with Associated Data

(standardized officially in 2025, selected in 2023)

Symmetric cryptography standards

AES

Advanced Encryption Standard

(standardized officially in 2001)

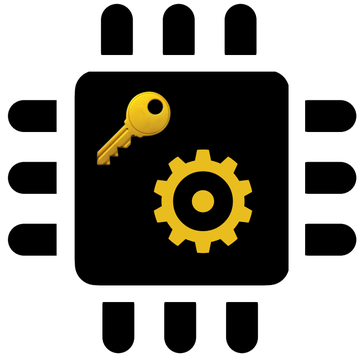
These standards are secure in black-box model

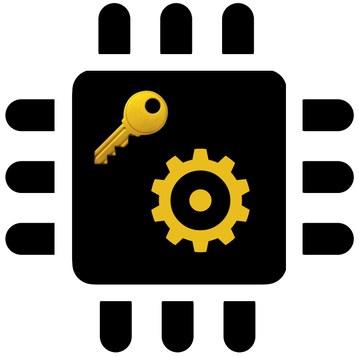
Ascon-AEAD

Ascon-based Authenticated Encryption with Associated Data

(standardized officially in 2025, selected in 2023)

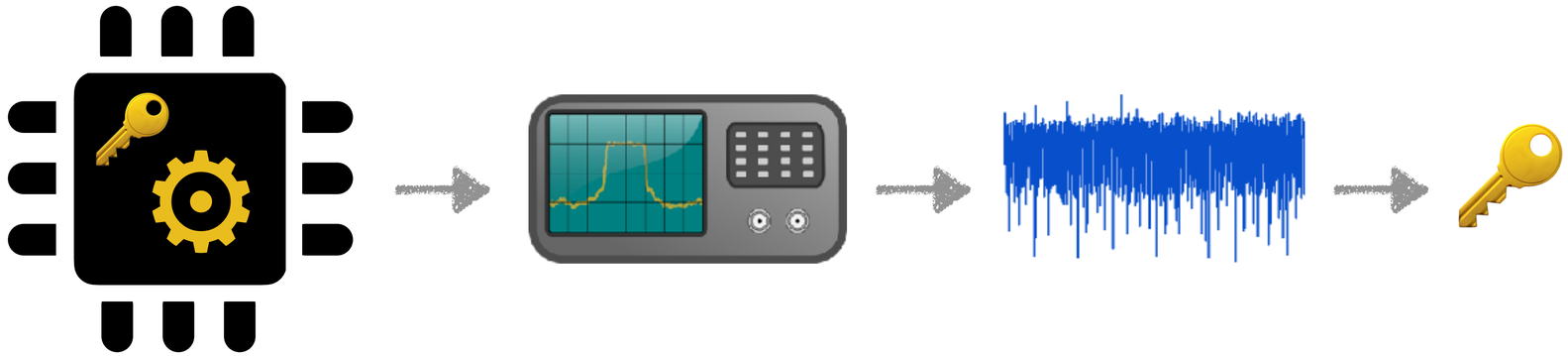
Physical attacks

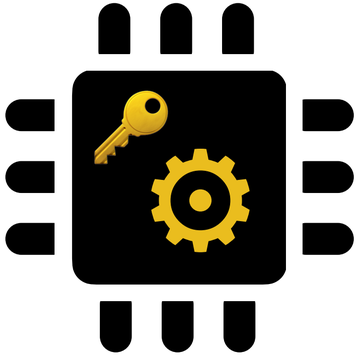




Physical attacks

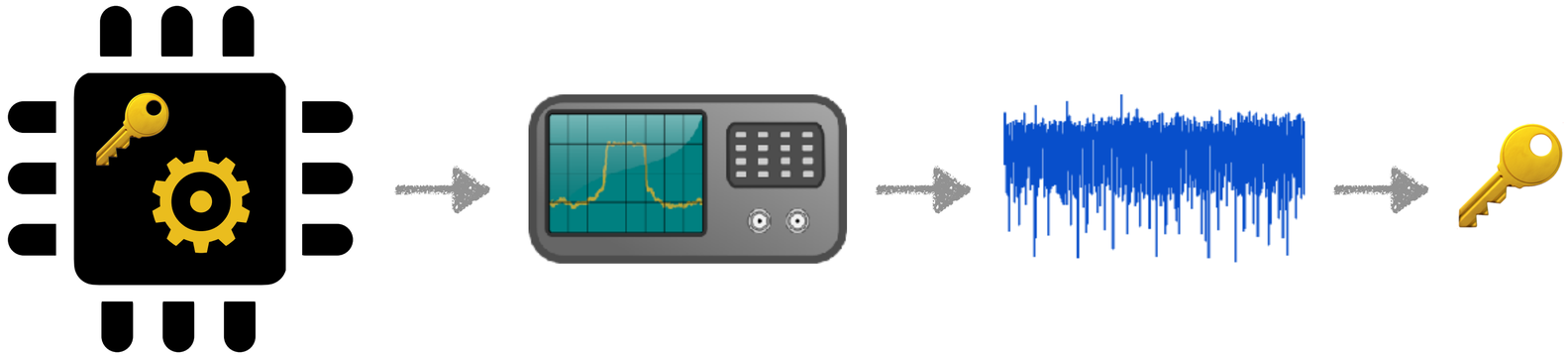
Side-Channel Attacks (SCA)



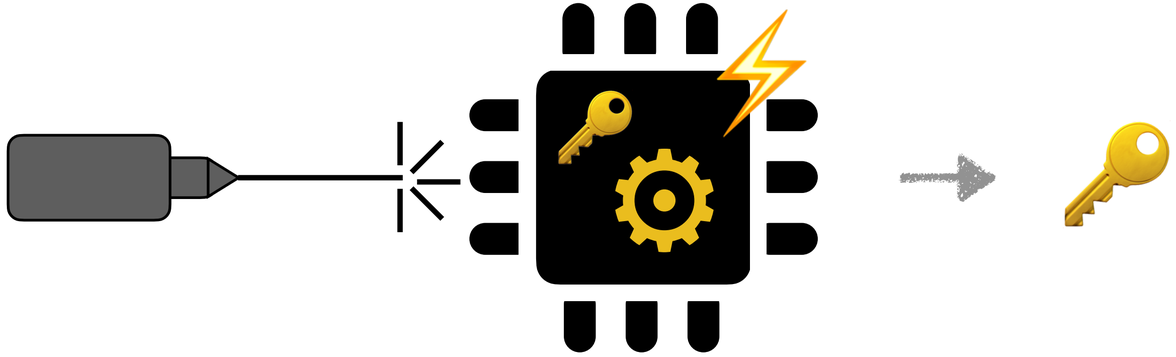


Physical attacks

Side-Channel Attacks (SCA)



Fault Attacks (FA)



Symmetric cryptography standards

AES

Advanced Encryption Standard

(standardized officially in 2001)

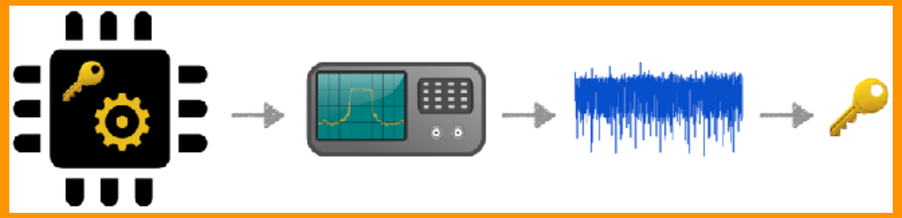
These standards are secure in black-box model

Ascon-AEAD

Ascon-based Authenticated Encryption with Associated Data

(standardized officially in 2025, selected in 2023)

→ this presentation



Correlation Power Analysis (CPA) attacks on Ascon-AEAD

👨🏻‍🎓 Contribution 1: [redacted]

👨🏻‍🎓 Contribution 2: [redacted]

Motivation

CPA attack on Ascon-AEAD

Motivation

CPA attack on Ascon-AEAD

[SD17]

First and successful

✅

Motivation

CPA attack on Ascon-AEAD

[SD17]

First and successful

[RADKA20]

Failed with 40,000 traces

(then proposed a successful deep-learning attack)

Motivation

CPA attack on Ascon-AEAD

[SD17]

First and successful

✅

[RADKA20]

Failed with 40,000 traces

(then proposed a successful deep-learning attack)

❌

no explicit explanation

Motivation

CPA attack on Ascon-AEAD

[SD17]

First and successful

✅

[RADKA20]

Failed with 40,000 traces

(then proposed a successful deep-learning attack)

❌

no explicit explanation

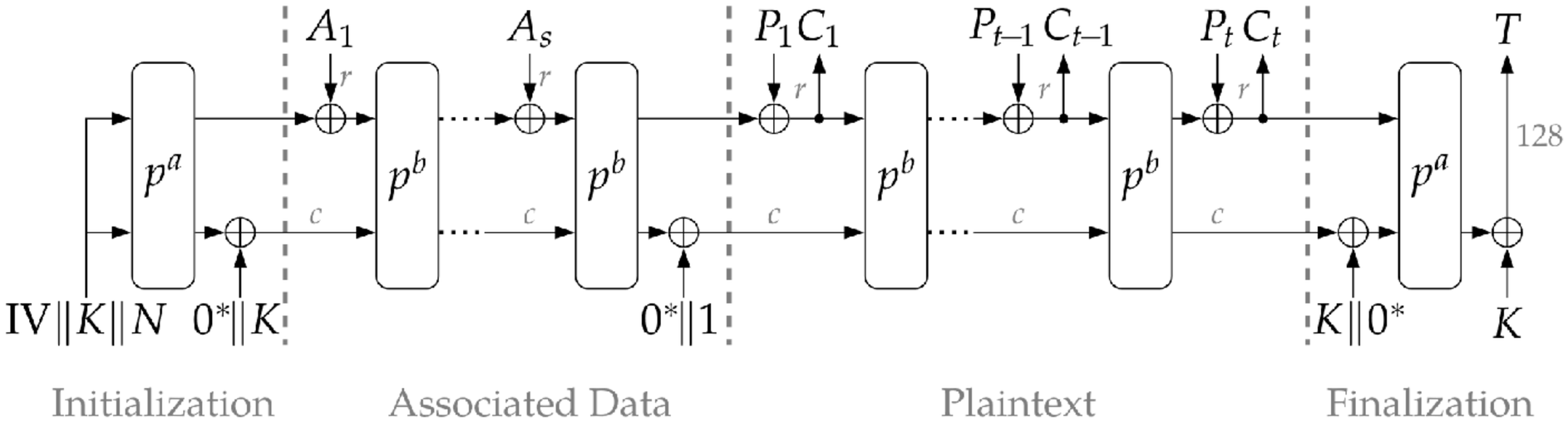
Why? Is the number of traces really the reason?

Correlation Power Analysis (CPA) attacks on Ascon-AEAD

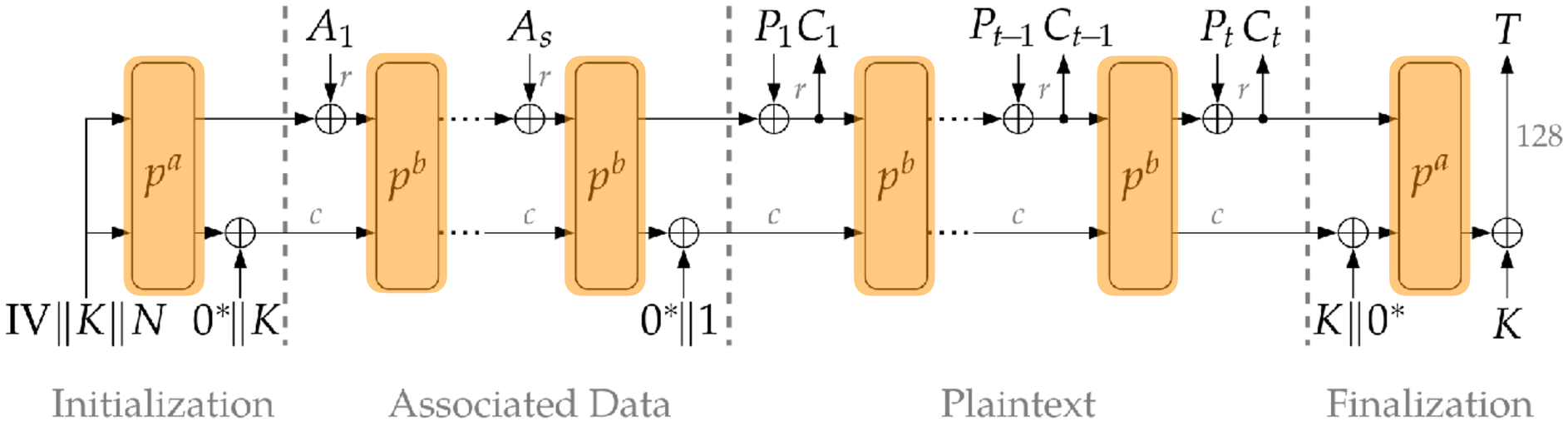
👨‍🎓 Contribution 1:

👨‍🎓 Contribution 2:

Ascon-AEAD

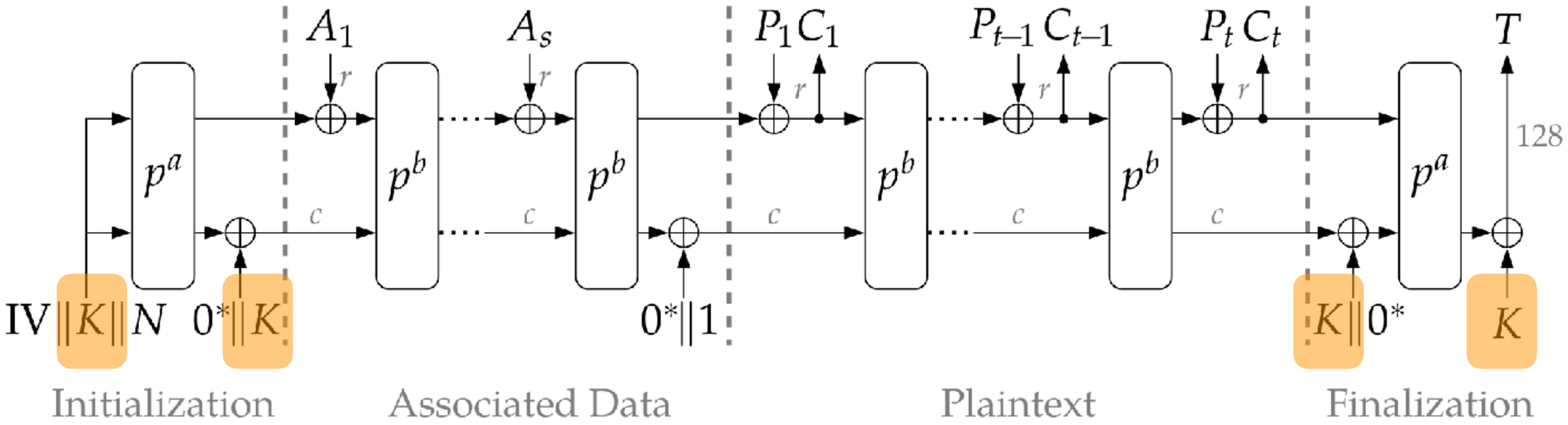

A_1
A_s
$P_1 C_1$
$P_{t-1} C_{t-1}$
$P_t C_t$
T
r
p^a
p^b
c
128
$\mathrm{IV} \| K \| N$
$0^* \| K$
$0^* \| 1$
$K \| 0^*$
K
Initialization
Associated Data
Plaintext
Finalization

Permutation blocks

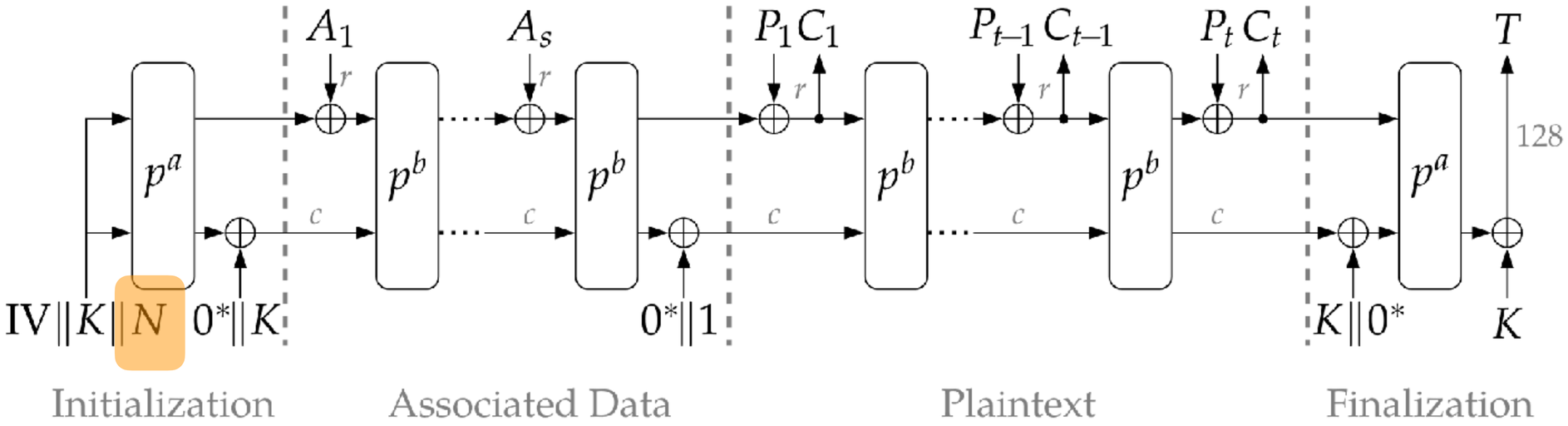


- p^a: 12 permutation rounds ($a = 12$)
- p^b: 8 permutation rounds ($b = 8$)

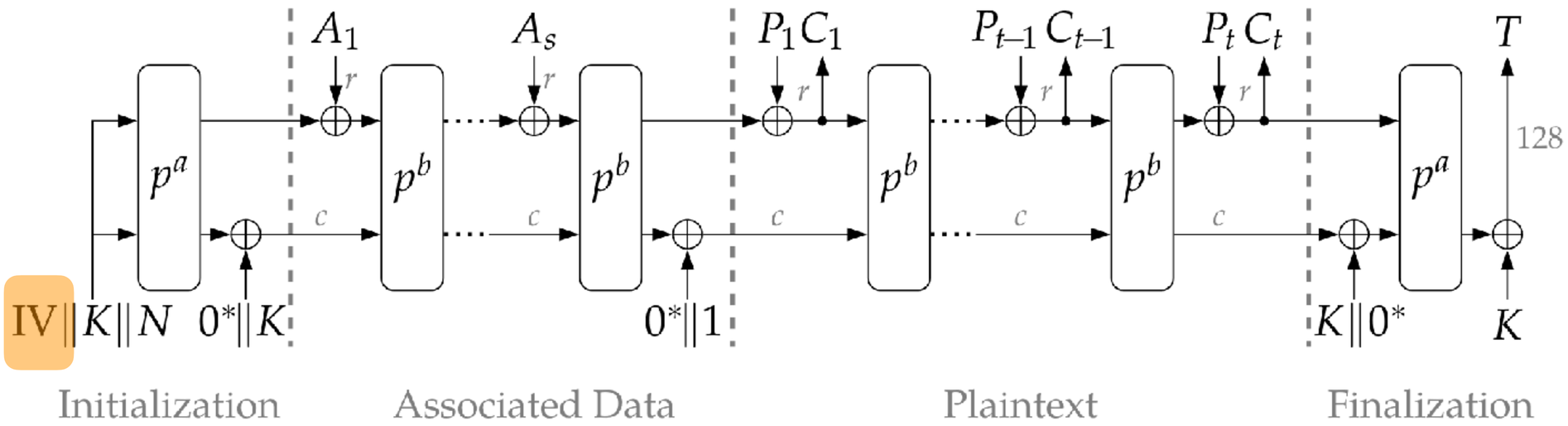
Key (128 bits)


A_1
A_s
$P_1 C_1$
$P_{t-1} C_{t-1}$
$P_t C_t$
T
p^a
p^b
r
c
128
IV ‖ K ‖ N
0* ‖ K
0* ‖ 1
K ‖ 0*
K
Initialization
Associated Data
Plaintext
Finalization

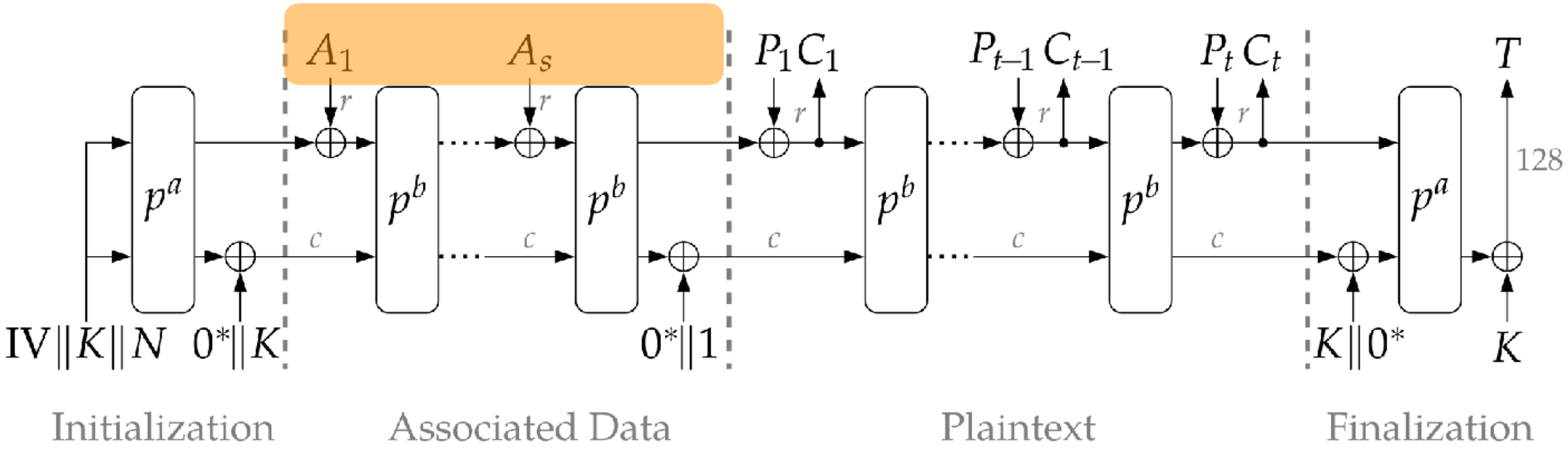
Nonce (128 bits)



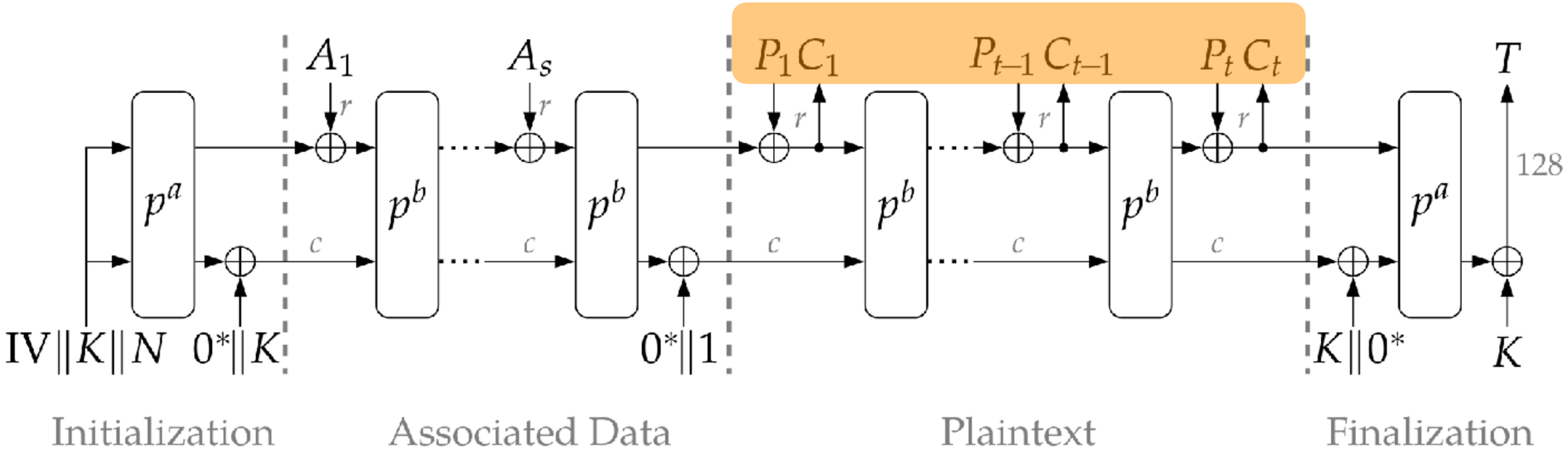
Initialization vector



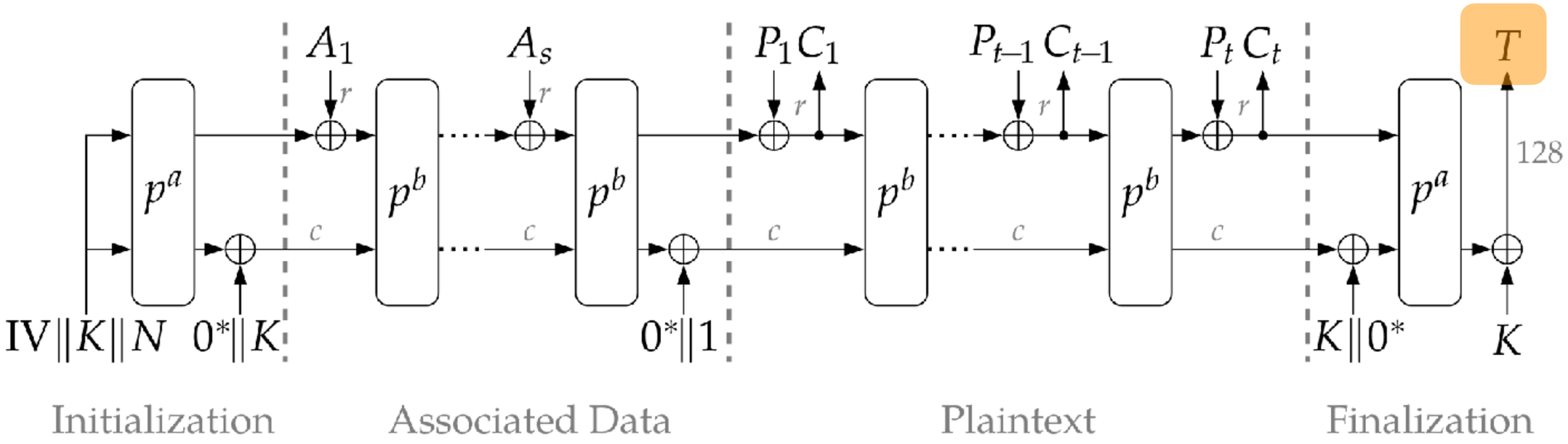
Associated data



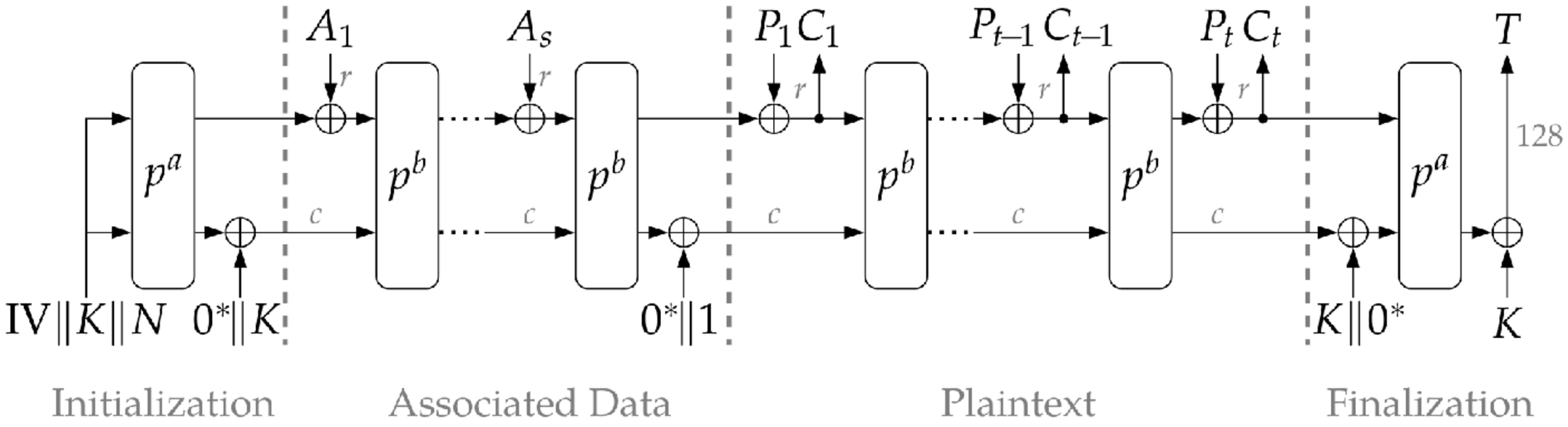
Plaintext / Ciphertext



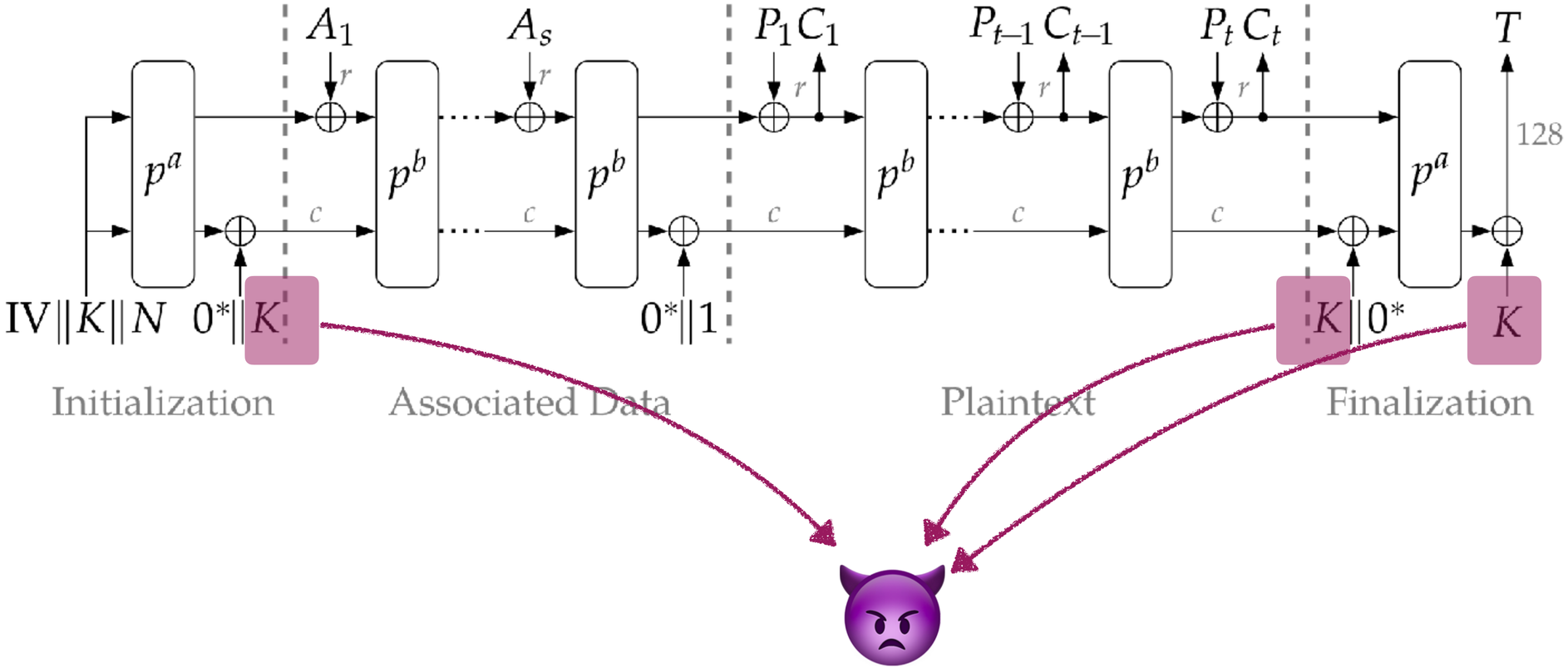
Verification tag



Focus of attack

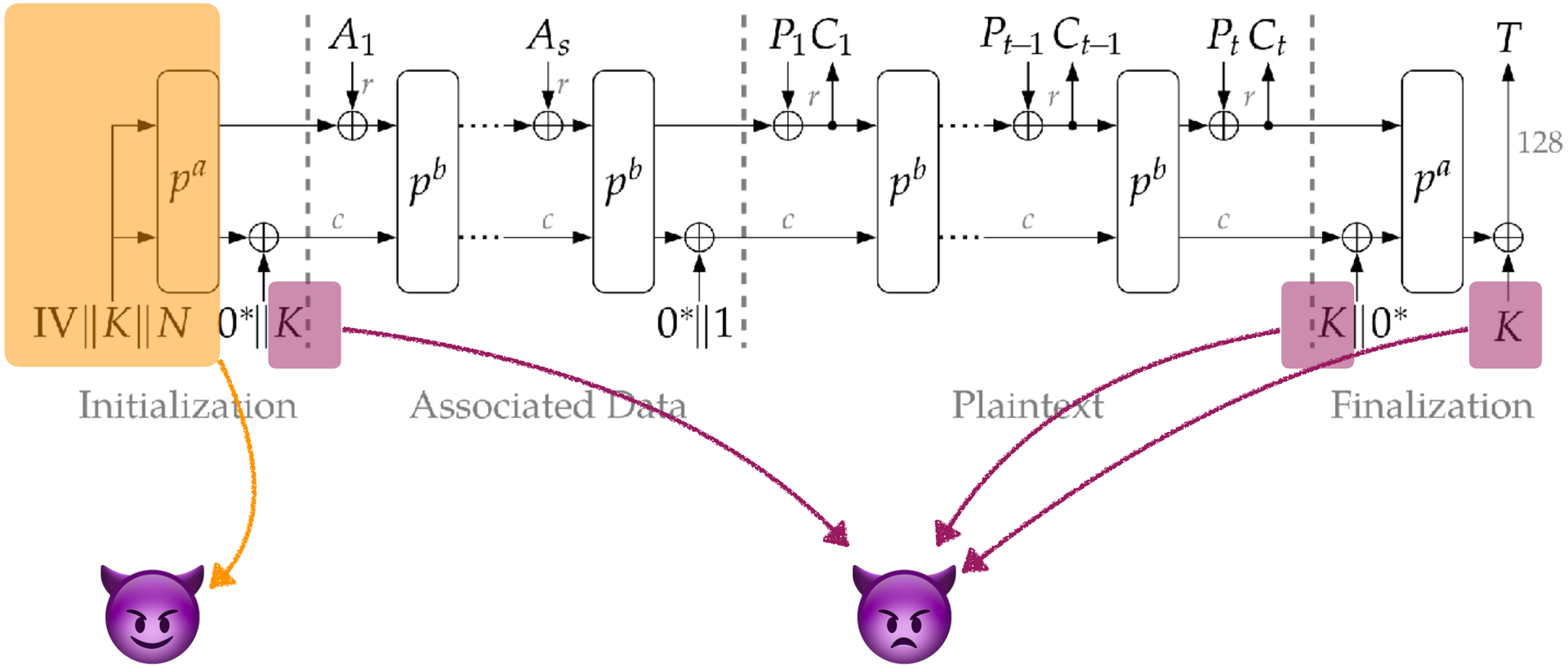


Focus of attack



can’t control full state

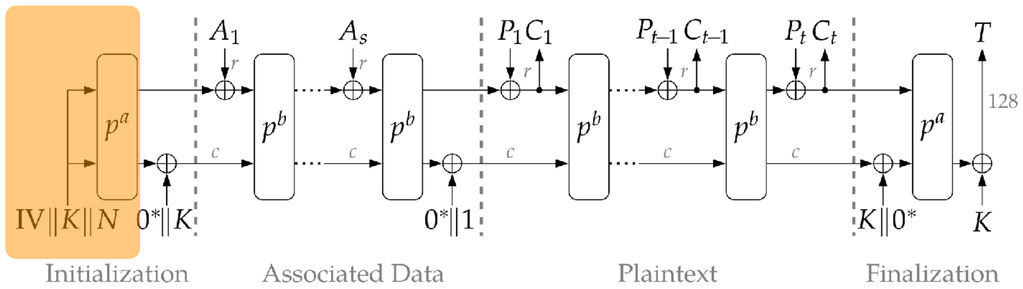
Focus of attack



know IV, control N, guess K

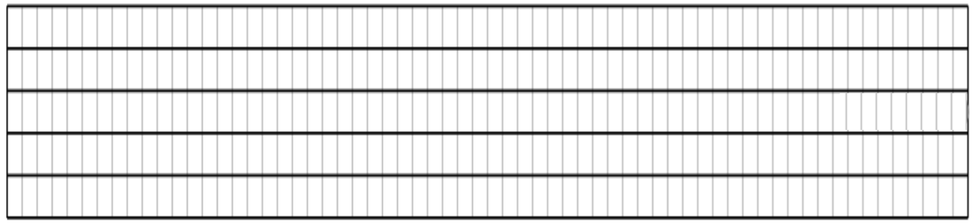
can’t control full state

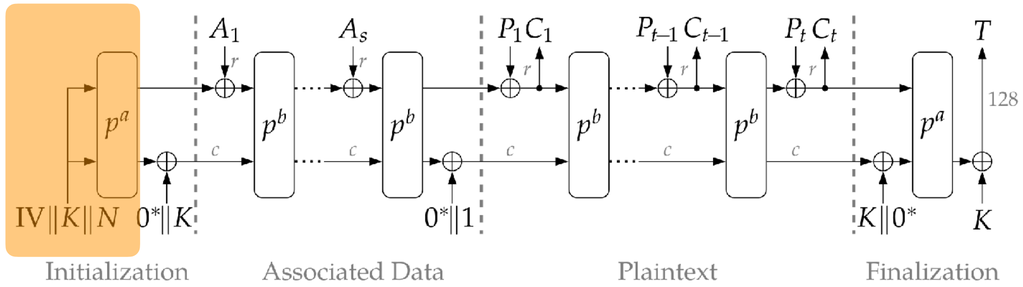
Round computation



Round computation

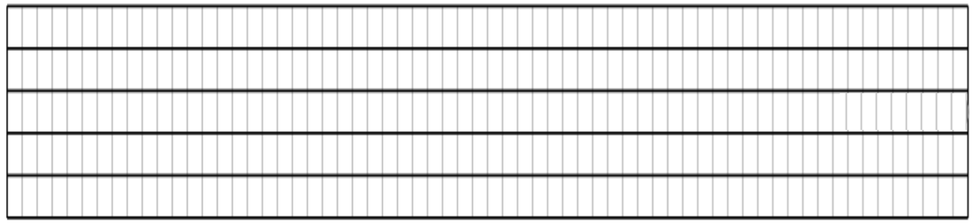
On 320-bit state = 5 x 64-bit words





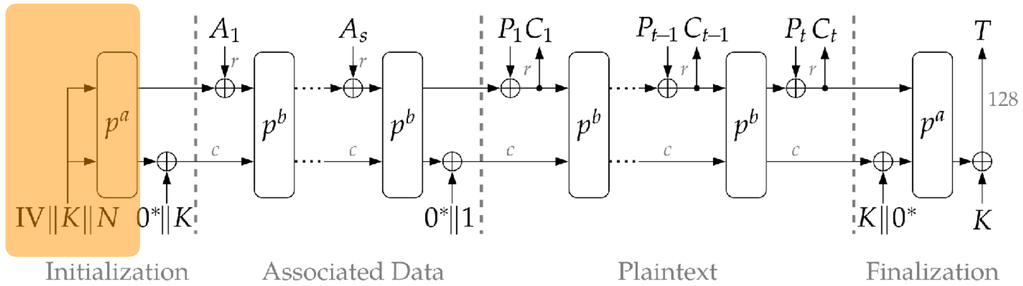
Round computation

On 320-bit state = 5 x 64-bit words



Input of the first round:

- 128-bit key : $K = (k_0, k_1)$
- 128-bit nonce : $N = (n_0, n_1)$
- 64-bit init. vector : IV

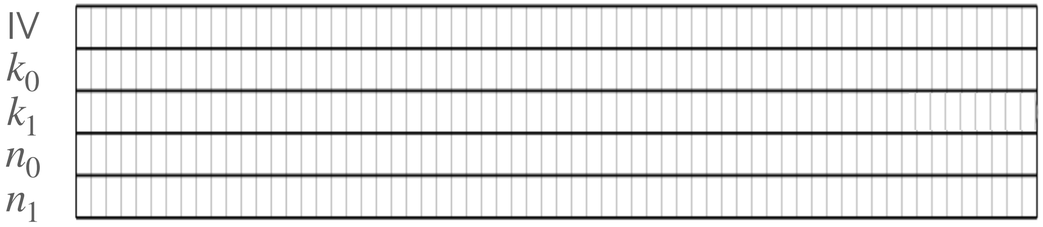


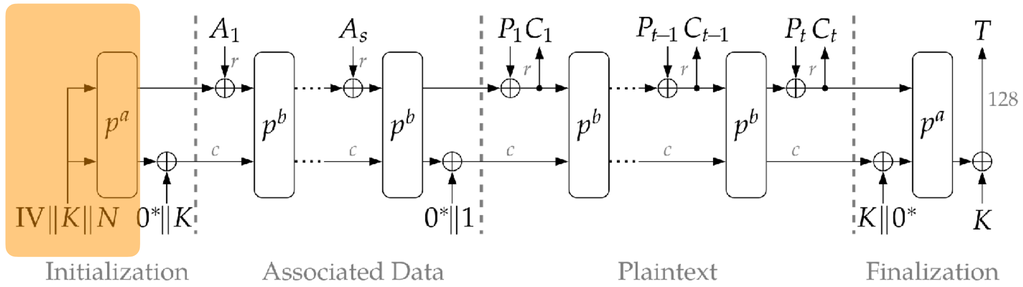
Round computation

On 320-bit state = 5 x 64-bit words

Input of the first round:

- 128-bit key : $K = (k_0, k_1)$
- 128-bit nonce : $N = (n_0, n_1)$
- 64-bit init. vector : IV



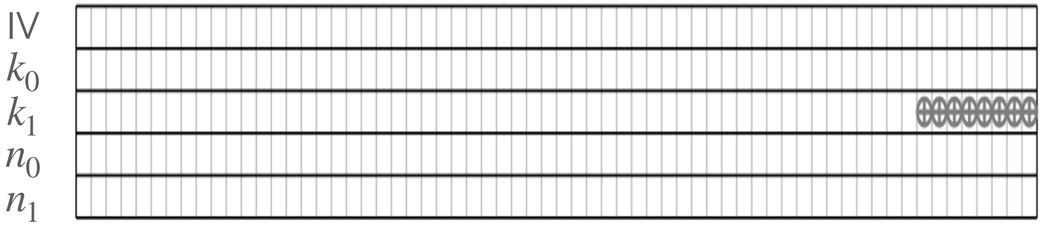


Round computation

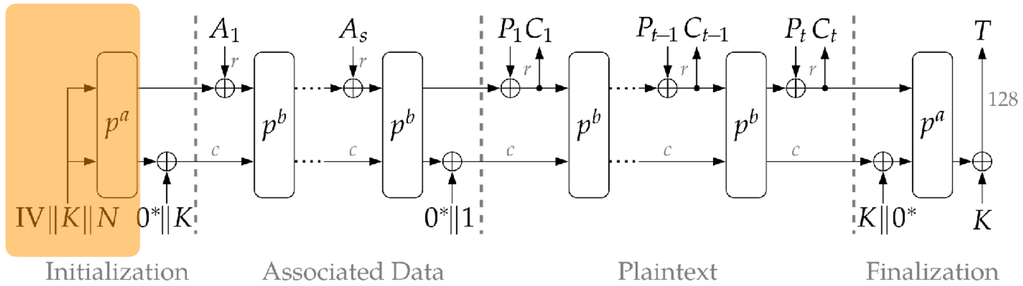
On 320-bit state = 5 x 64-bit words

Input of the first round:

- 128-bit key : $K = (k_0, k_1)$
- 128-bit nonce : $N = (n_0, n_1)$
- 64-bit init. vector : IV



(1) Round constant addition

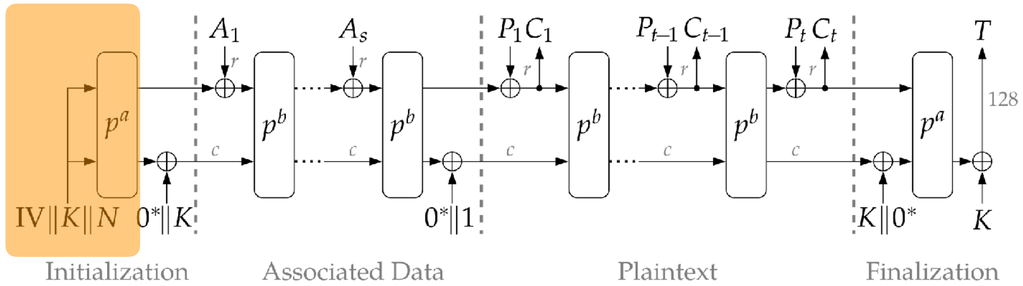


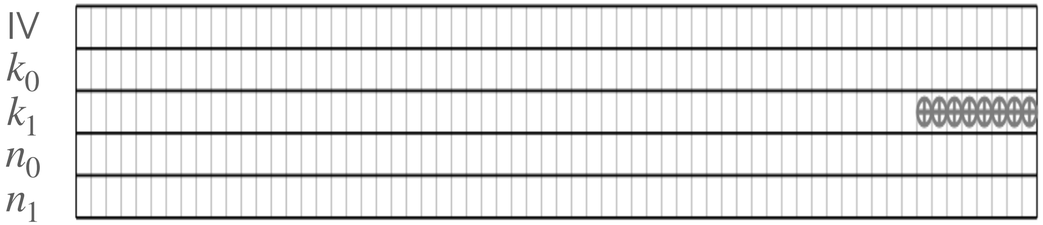
Round computation

On 320-bit state = 5 x 64-bit words

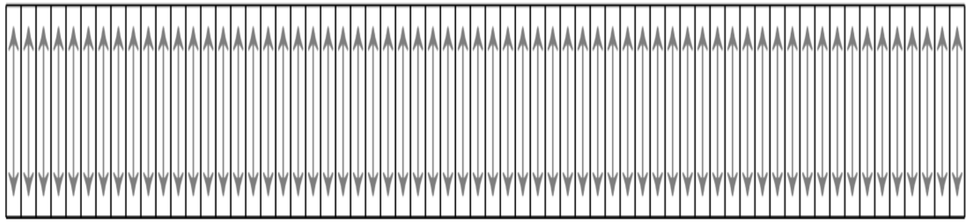
Input of the first round:

- 128-bit key : $K = (k_0, k_1)$
- 128-bit nonce : $N = (n_0, n_1)$
- 64-bit init. vector : IV





(1) Round constant addition



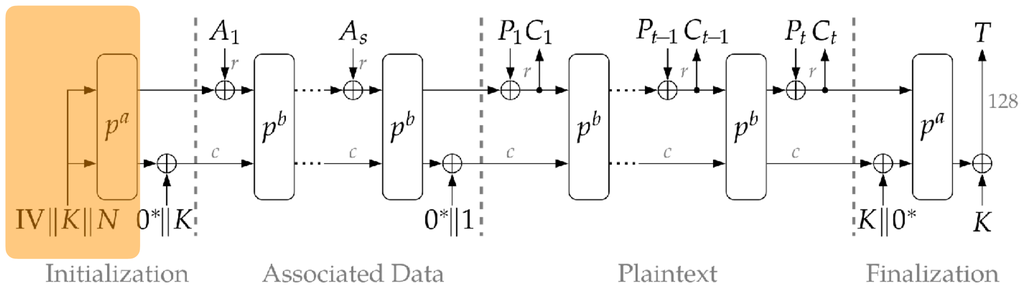
(2) Substitution *(vertical)*

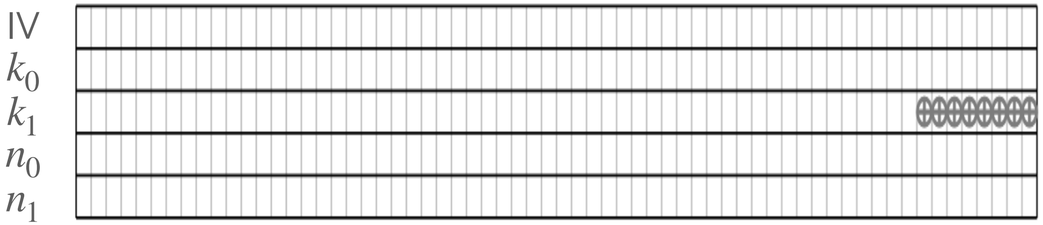
Round computation

On 320-bit state = 5 x 64-bit words

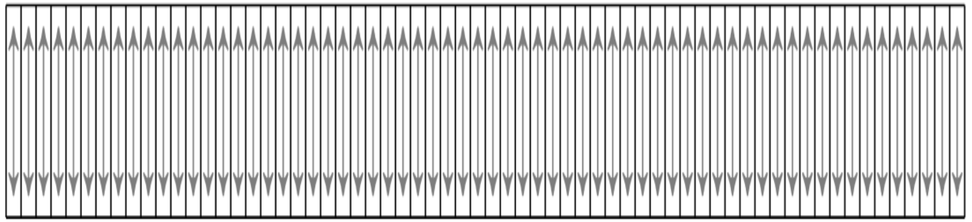
Input of the first round:

- 128-bit key : $K = (k_0, k_1)$
- 128-bit nonce : $N = (n_0, n_1)$
- 64-bit init. vector : IV

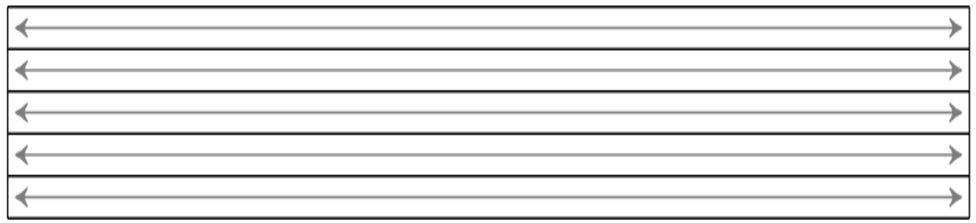




(1) Round constant addition

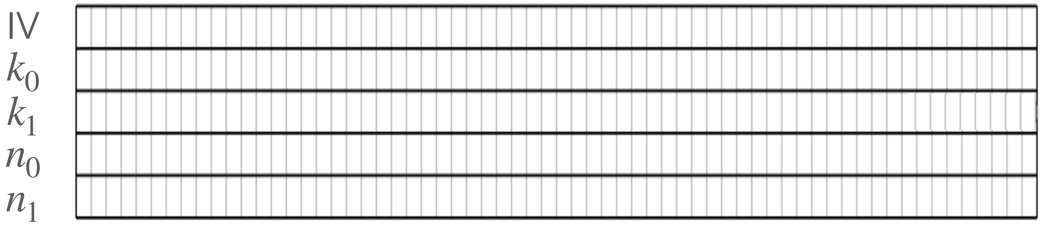


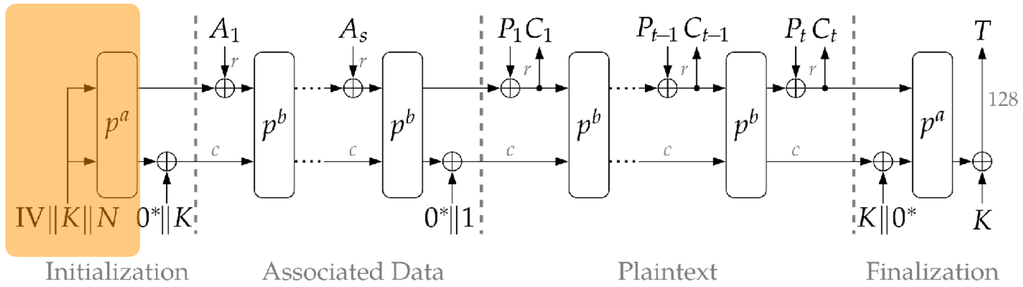
(2) Substitution *(vertical)*



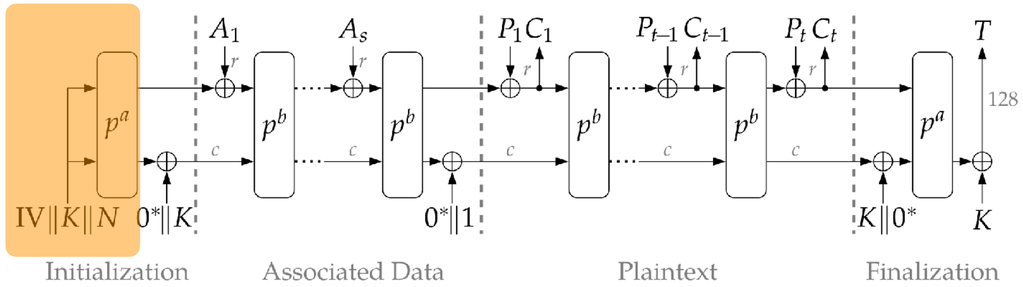
(3) Linear diffusion *(horizontal)*

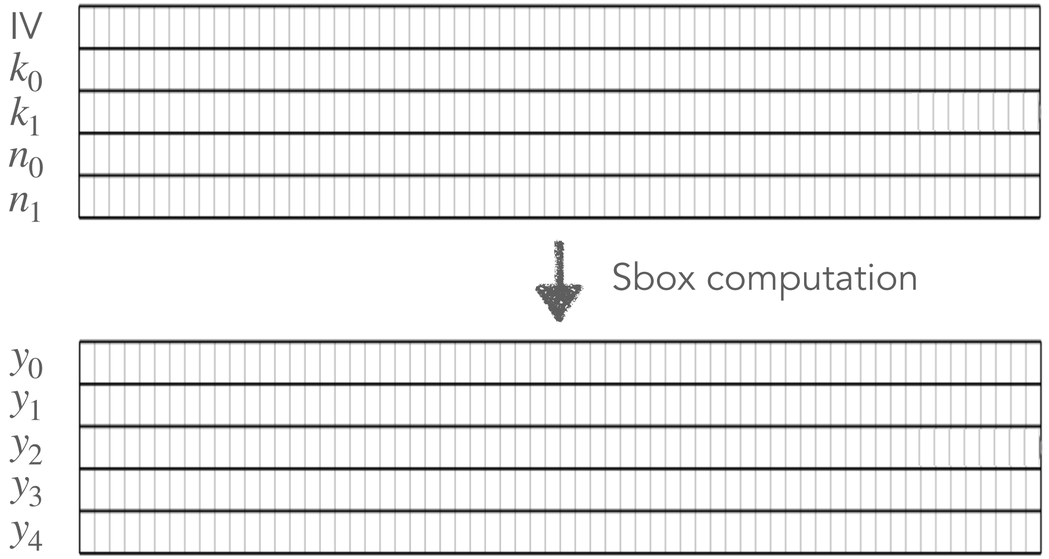
State change in first round



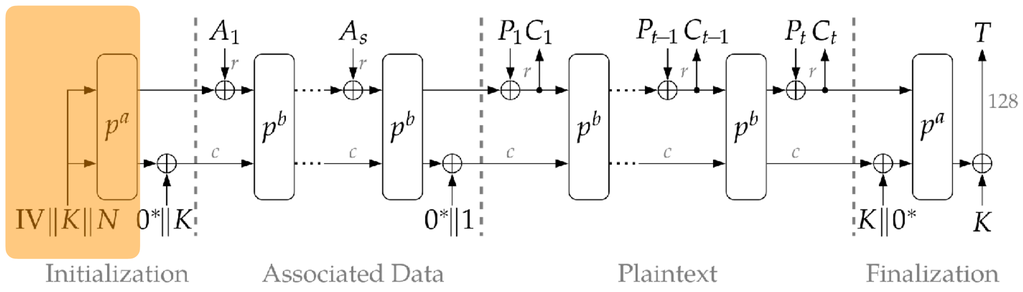


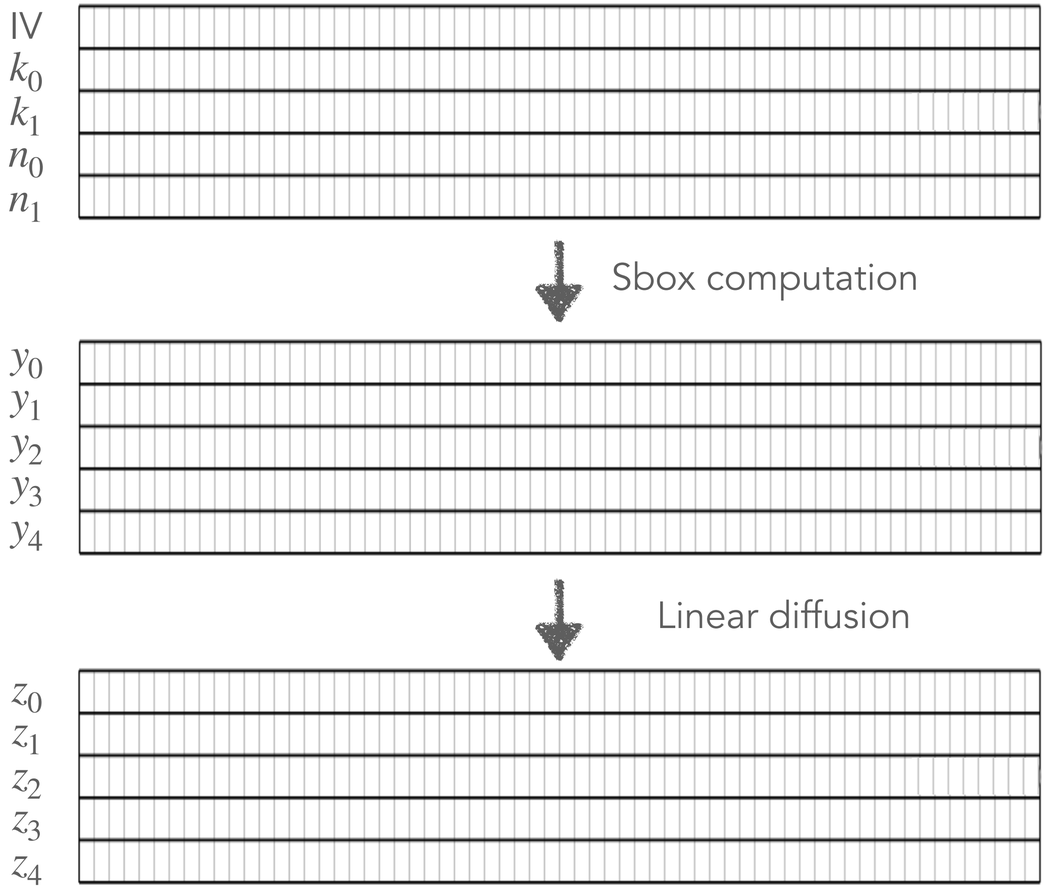
State change in first round



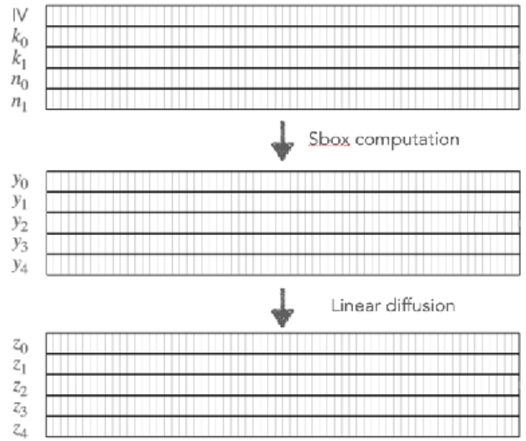


State change in first round



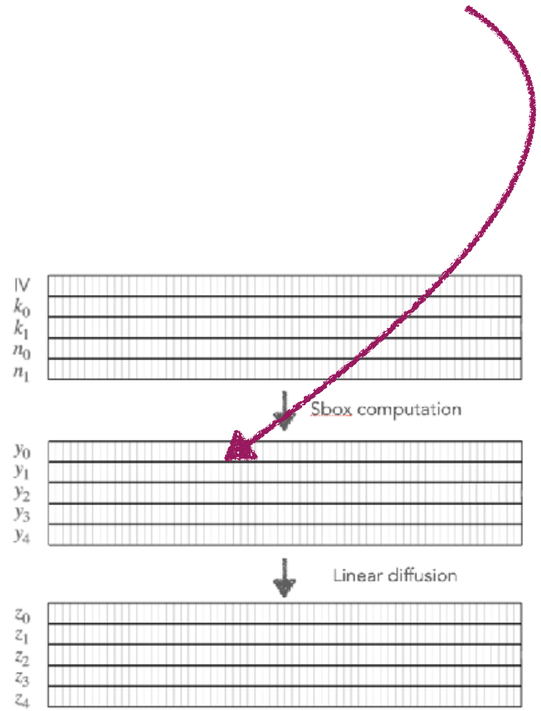


CPA attack on Ascon-AEAD



CPA attack on Ascon-AEAD

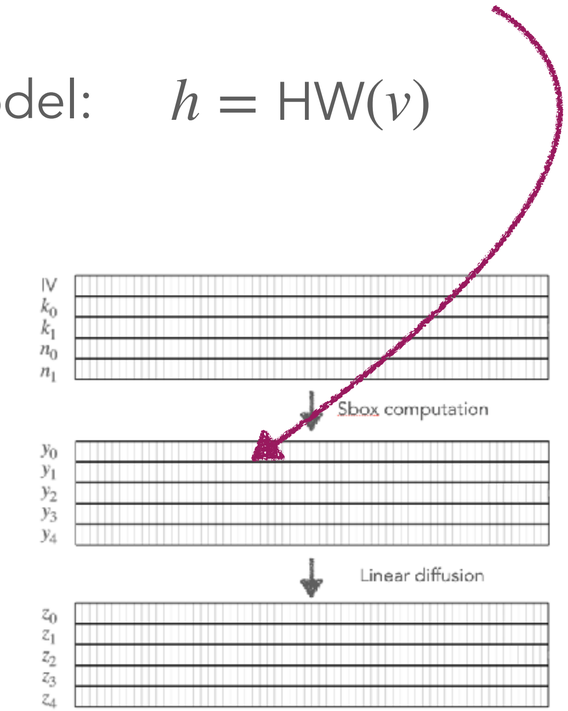
Selection function: $v = f(\text{nonce}, \text{key})$



CPA attack on Ascon-AEAD

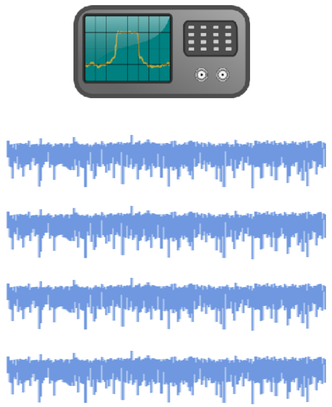
Selection function: $v = f(\text{nonce}, \text{key})$

Leakage model: $h = \text{HW}(v)$



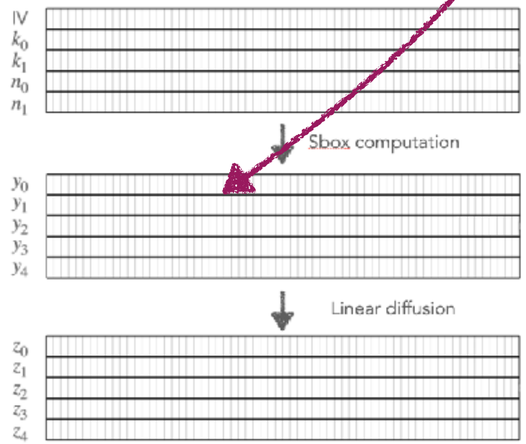
CPA attack on Ascon-AEAD

nonce_1
nonce_2
nonce_3
nonce_4



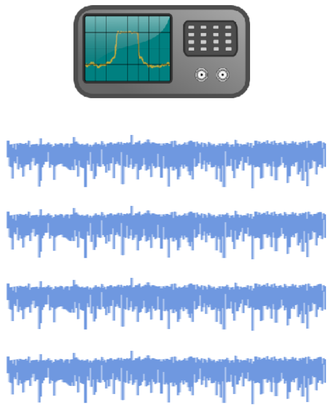
Selection function: $v = f(\text{nonce}, \text{key})$

Leakage model: $h = \text{HW}(v)$



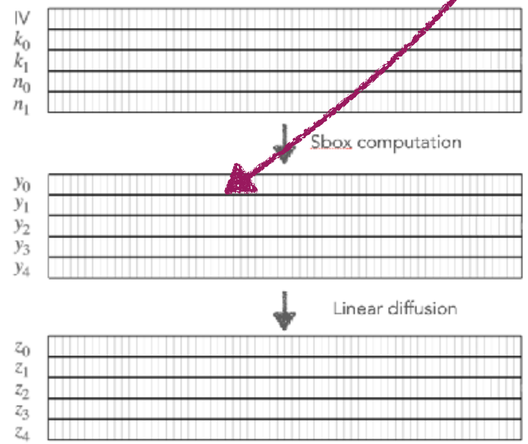
CPA attack on Ascon-AEAD

	key guesses		
	key_1	key_2	key_3
$nonce_1$			
$nonce_2$			
$nonce_3$			
$nonce_4$			



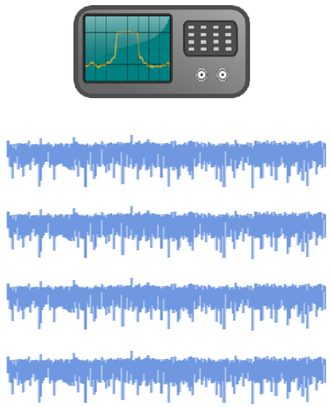
Selection function: $v = f(\text{nonce}, \text{key})$

Leakage model: $h = \text{HW}(v)$



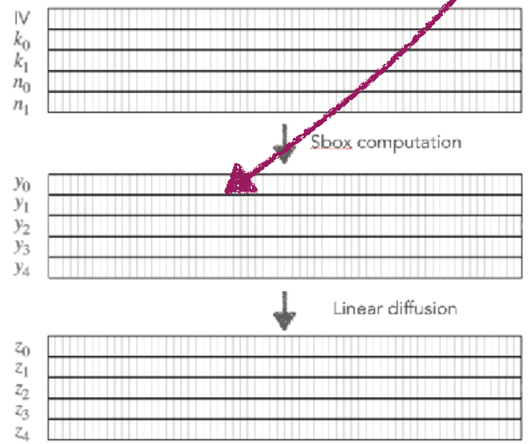
CPA attack on Ascon-AEAD

	key guesses		
	key$_1$	key$_2$	key$_3$
nonce$_1$	h_{11}		
nonce$_2$	h_{21}		
nonce$_3$	h_{31}		
nonce$_4$	h_{41}		



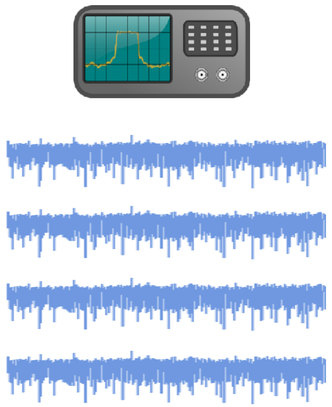
Selection function: $v = f(\text{nonce}, \text{key})$

Leakage model: $h = \text{HW}(v)$



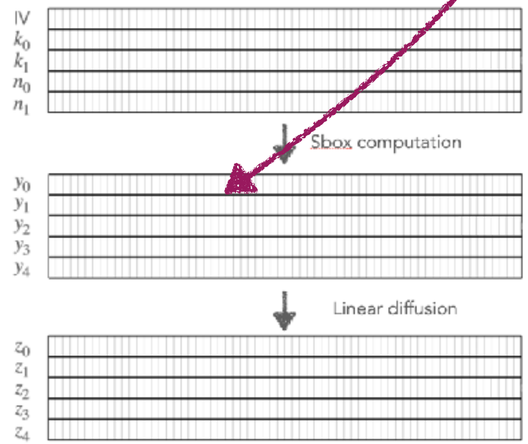
CPA attack on Ascon-AEAD

	key guesses		
	key$_1$	key$_2$	key$_3$
nonce$_1$	h_{11}	h_{12}	h_{13}
nonce$_2$	h_{21}	h_{22}	h_{23}
nonce$_3$	h_{31}	h_{32}	h_{33}
nonce$_4$	h_{41}	h_{42}	h_{43}

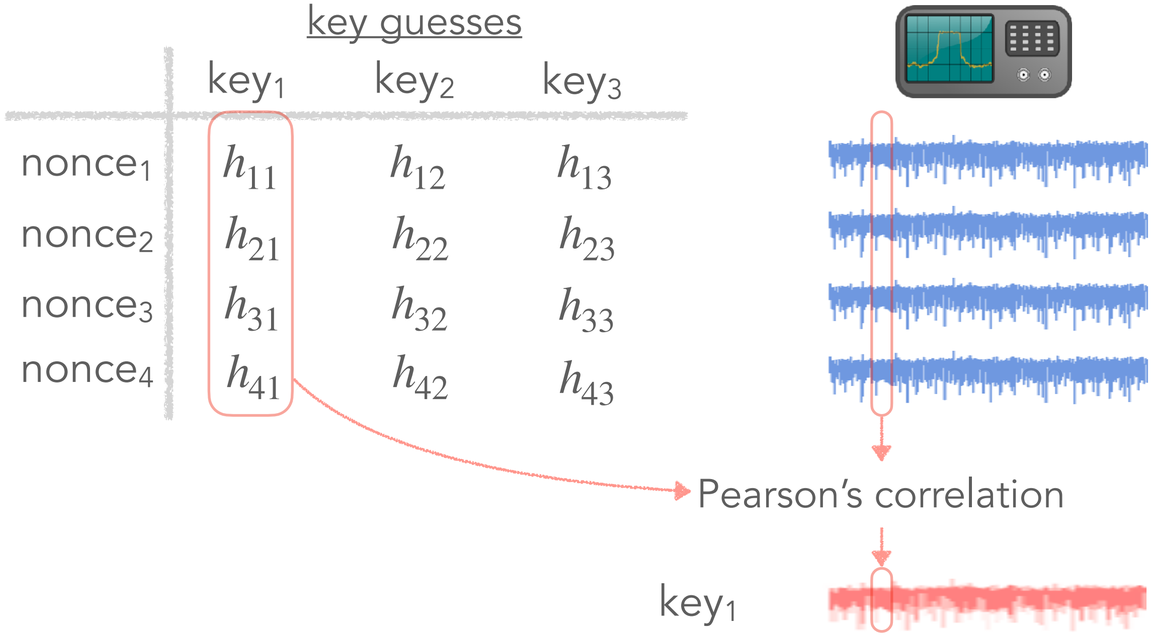


Selection function: $v = f(\text{nonce}, \text{key})$

Leakage model: $h = \text{HW}(v)$

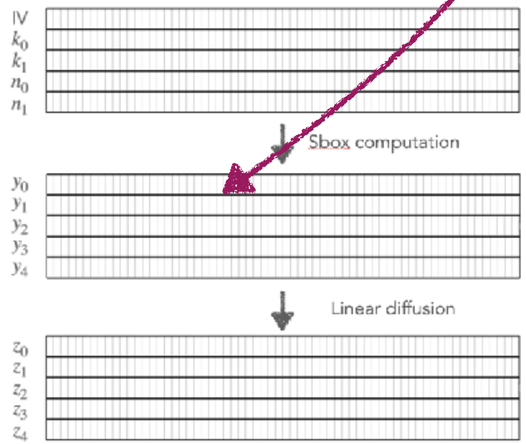


CPA attack on Ascon-AEAD

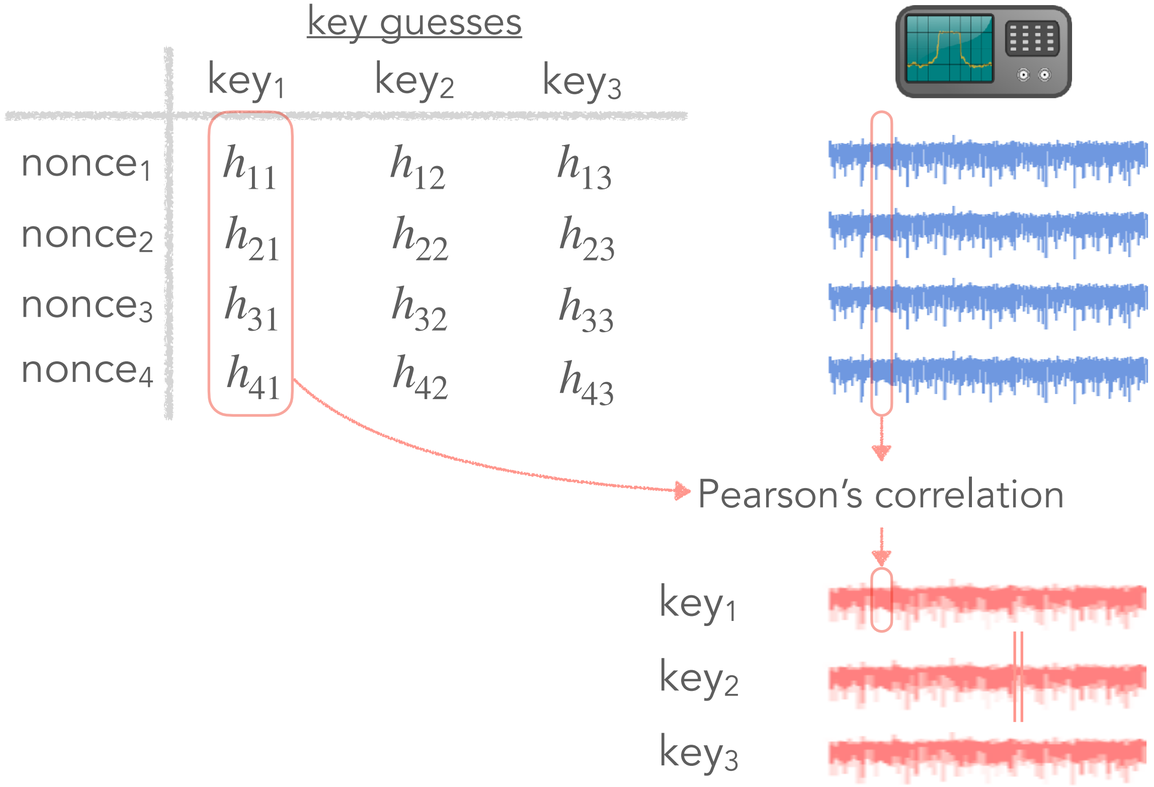


Selection function: $v = f(\text{nonce}, \text{key})$

Leakage model: $h = \text{HW}(v)$

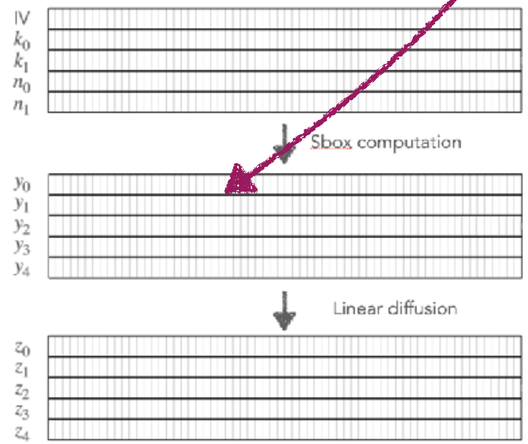


CPA attack on Ascon-AEAD

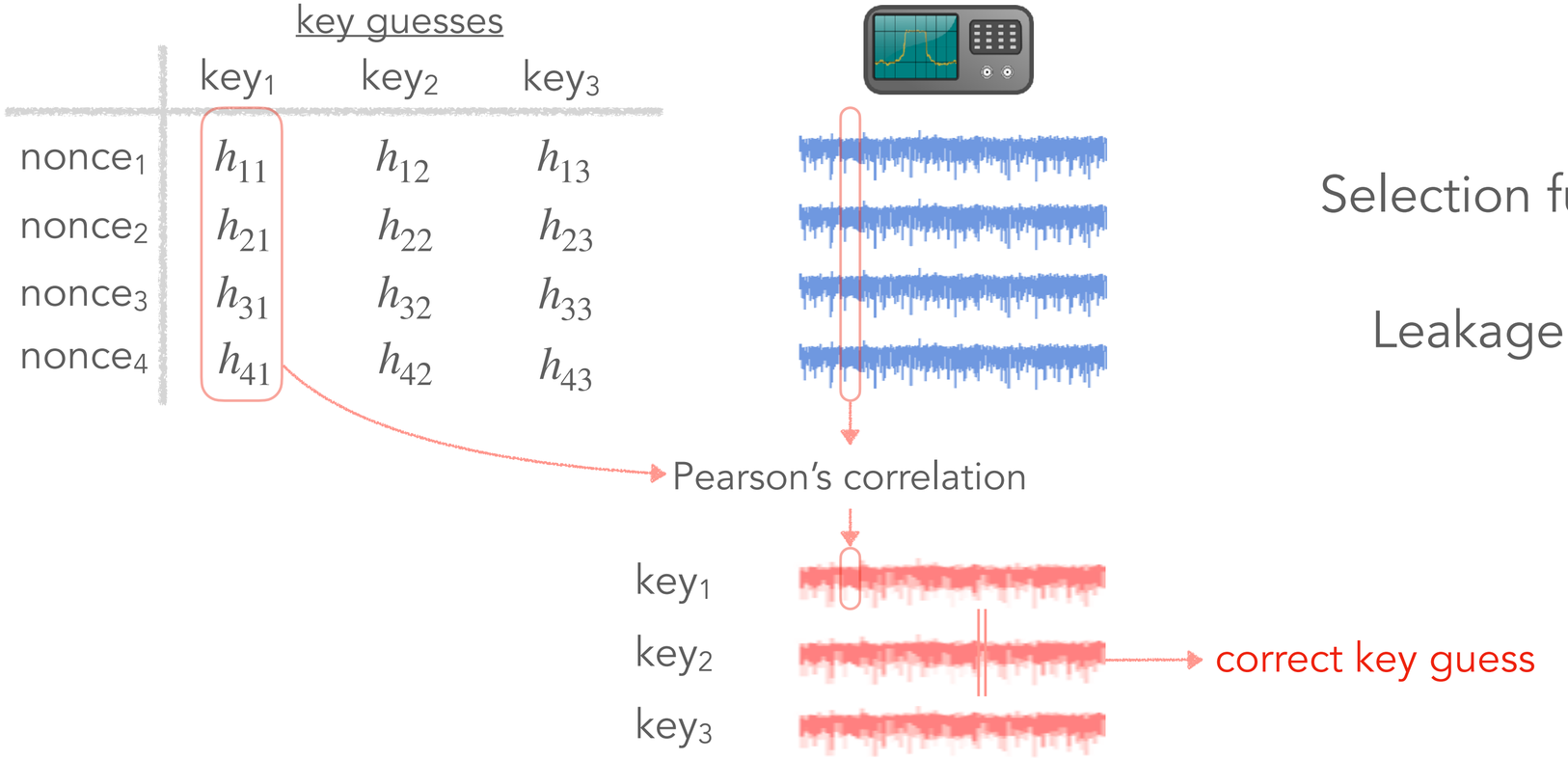


Selection function: $v = f(\text{nonce}, \text{key})$

Leakage model: $h = \text{HW}(v)$

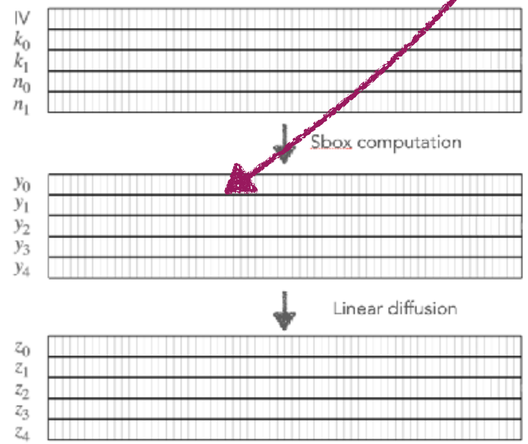


CPA attack on Ascon-AEAD

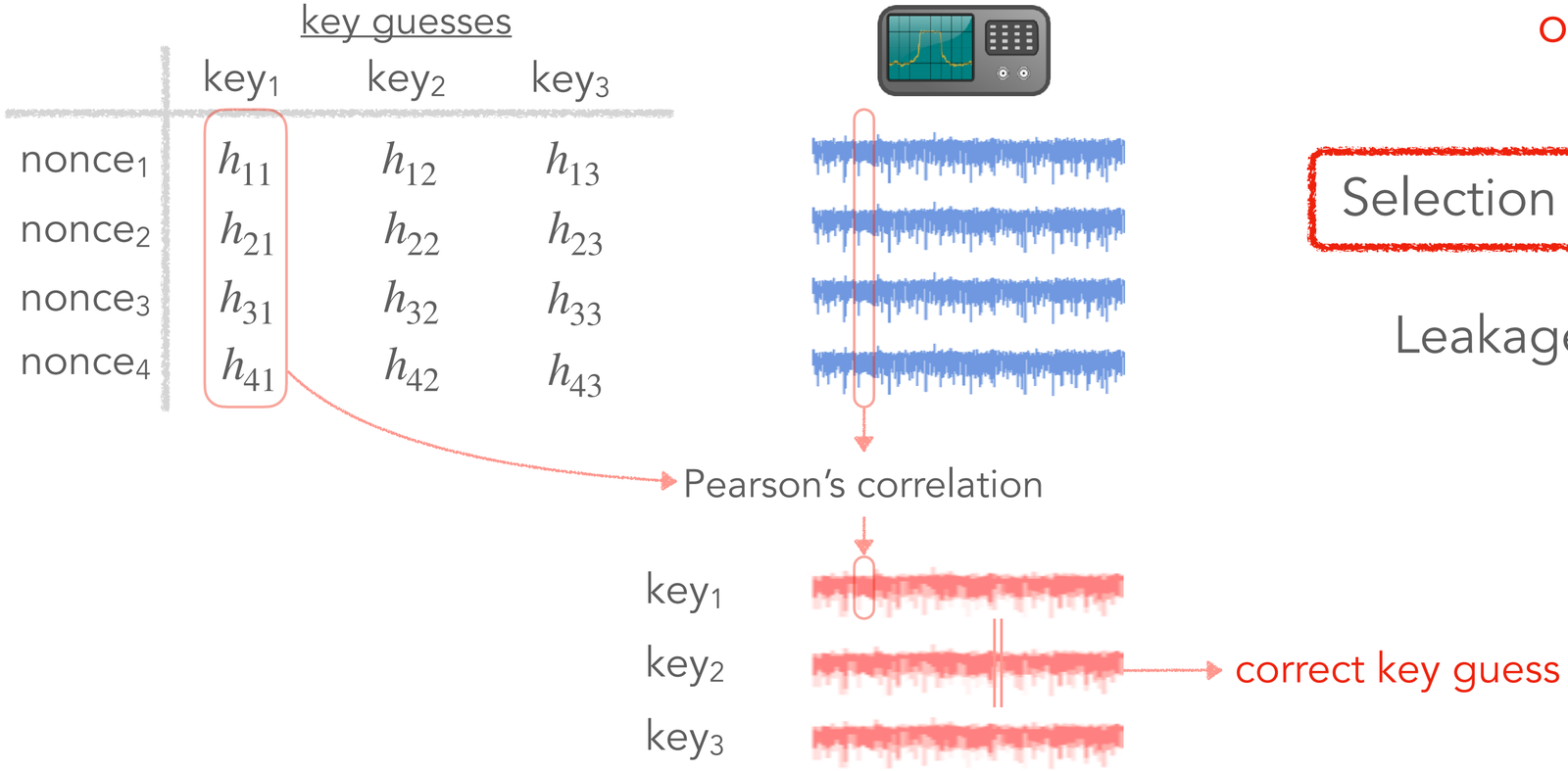


Selection function: $v = f(\text{nonce}, \text{key})$

Leakage model: $h = \text{HW}(v)$



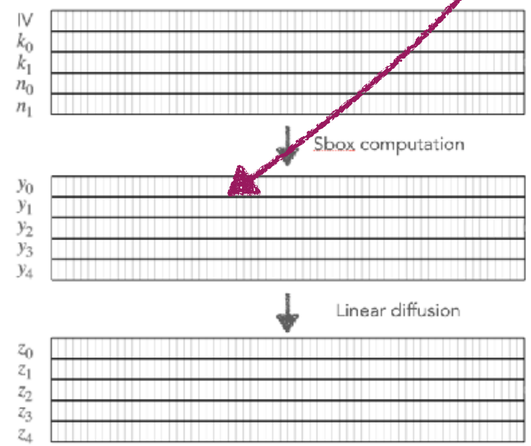
CPA attack on Ascon-AEAD



one of the core factors

Selection function: $v = f(\text{nonce}, \text{key})$

Leakage model: $h = \text{HW}(v)$



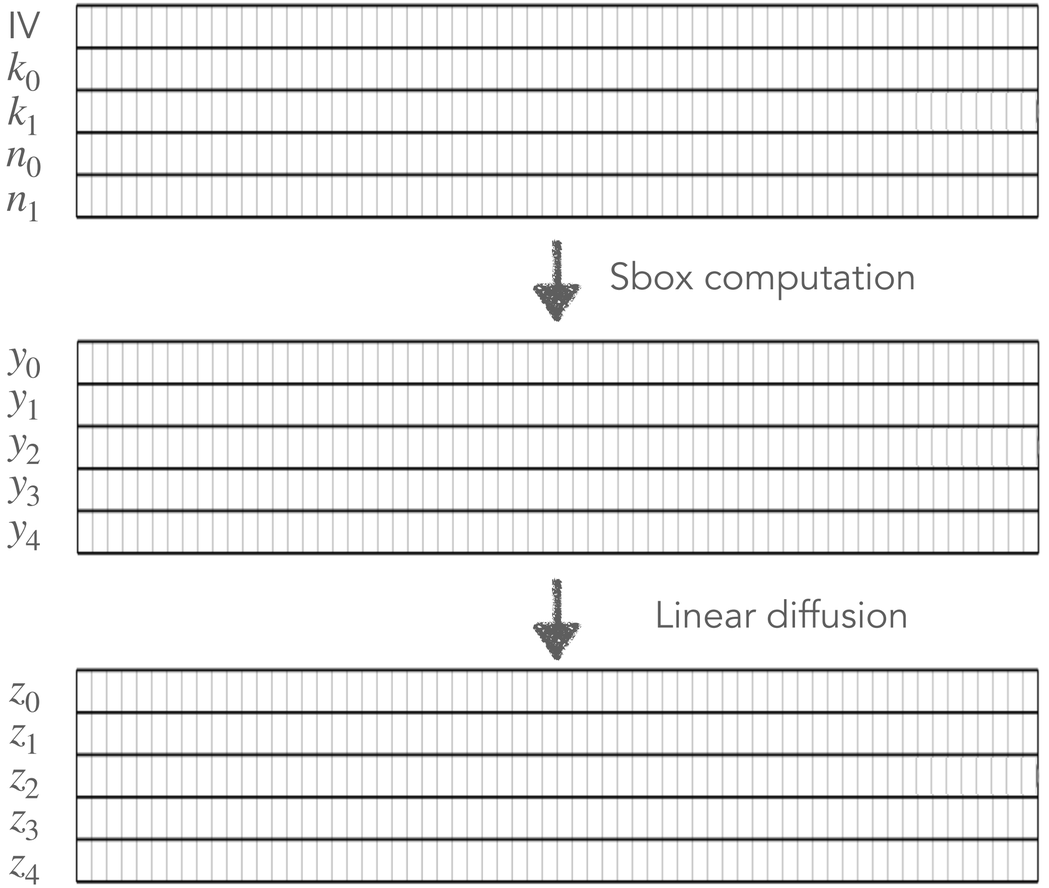
Different approaches for selection function

[RADKA20] Failed with 40,000 traces

❌

[SD17] Successful

✅



Different approaches for selection function

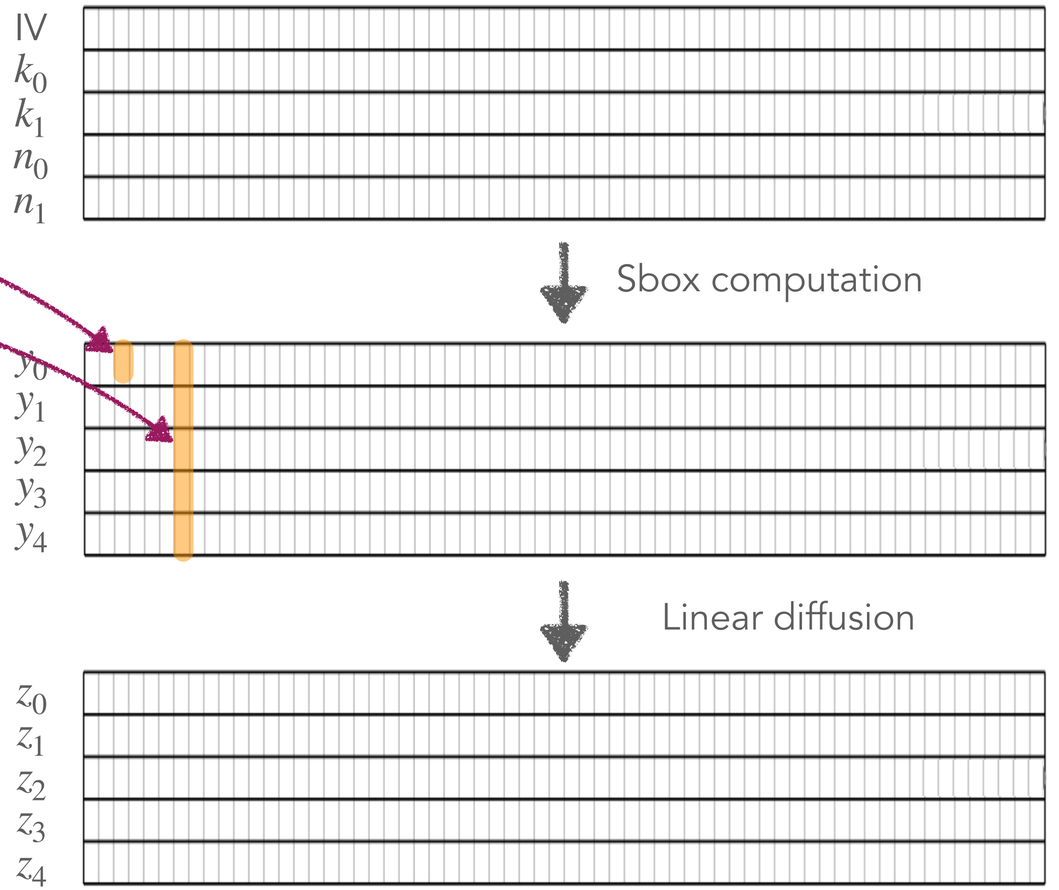
[RADKA20] Failed with 40,000 traces

❌

- Sbox output bit
- HW of Sbox output

[SD17] Successful

✅



Different approaches for selection function

[RADKA20] Failed with 40,000 traces

❌

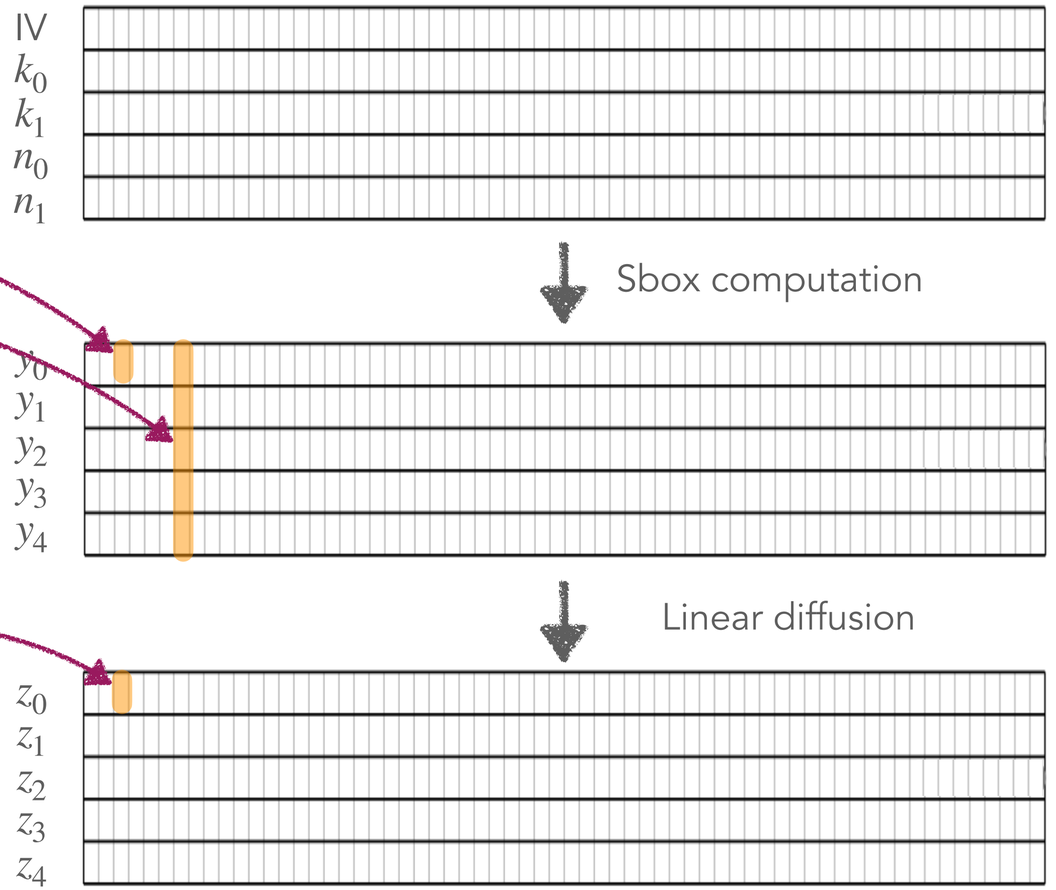
- Sbox output bit
- HW of Sbox output

[SD17] Successful

✅

- Linear layer output bit

(with the computation fine-tuned)



Different approaches for selection function

[RADKA20] Failed with 40,000 traces

❌

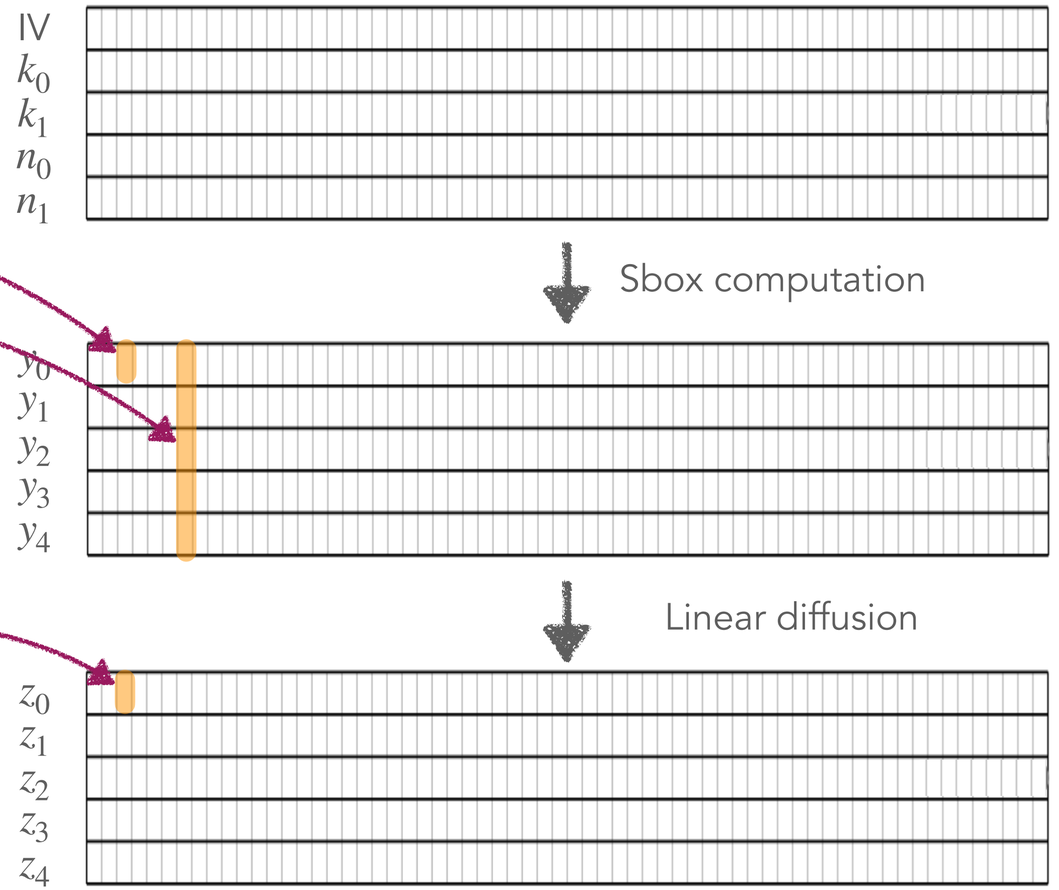
- Sbox output bit
- HW of Sbox output

[SD17] Successful

✅

- Linear layer output bit

(with the computation fine-tuned)



What are the differences?

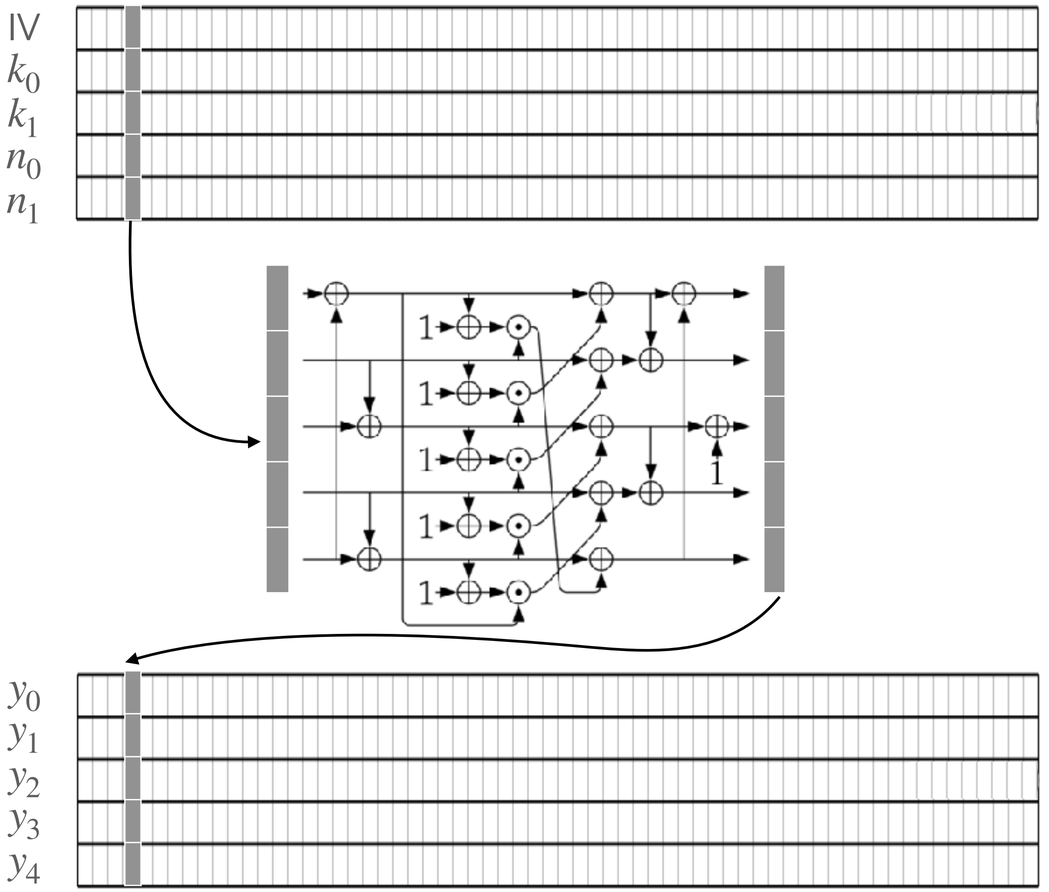
Correlation Power Analysis (CPA) attacks on Ascon-AEAD

👨‍🎓 Contribution 1: in-depth analysis of selection functions

Nguyen, Grosso, Cayrel - CASCADE 2025

👨‍🎓 Contribution 2:

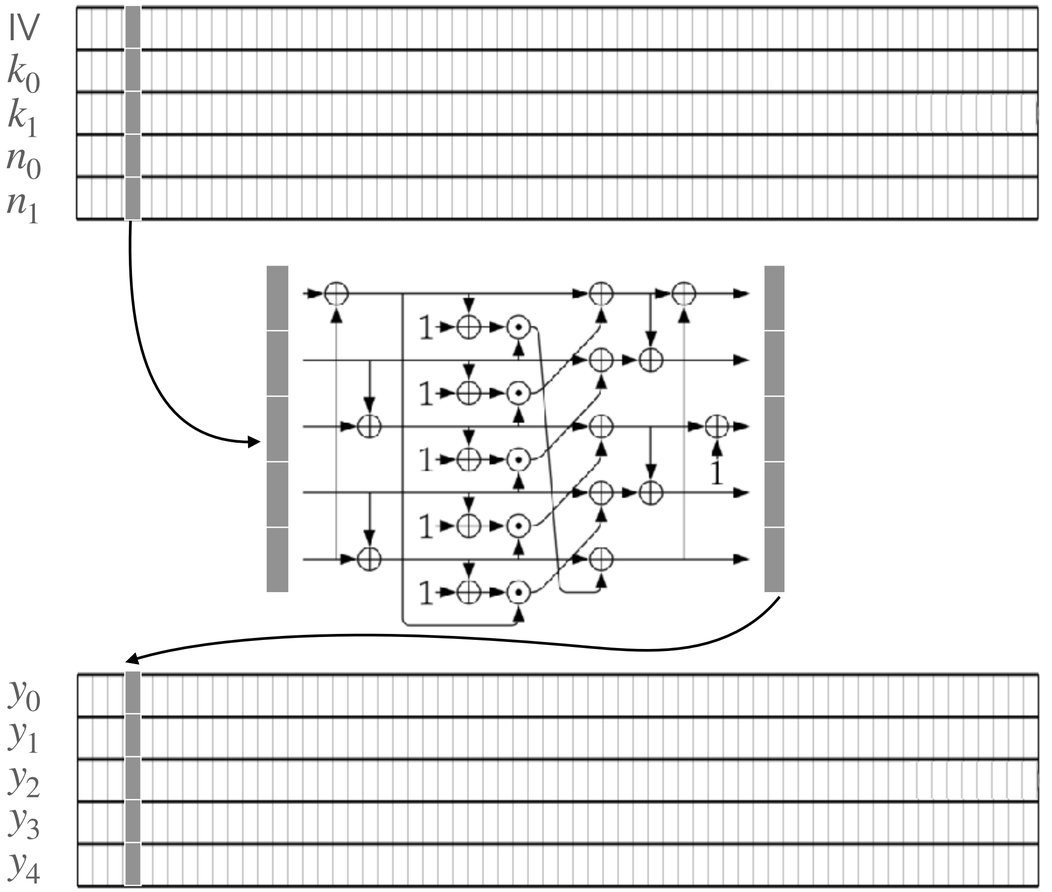
😈 Sbox output as attack point



😈 Sbox output as attack point

In an Sbox computation y_0^j:

- 2 bits of key : (k_0^j, k_1^j)
- 2 bits of nonce : (n_0^j, n_1^j)

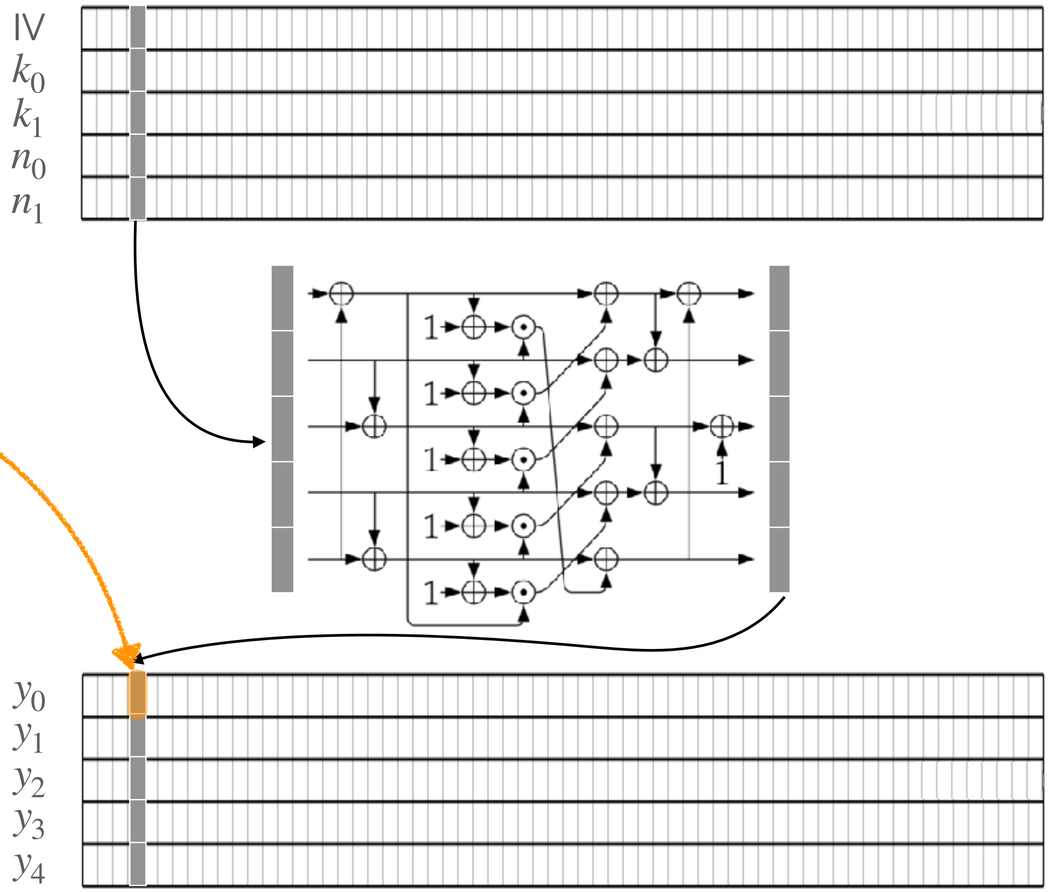


😈 Sbox output as attack point

In an Sbox computation y_0^j:

- 2 bits of key : (k_0^j, k_1^j)
- 2 bits of nonce : (n_0^j, n_1^j)

$$y_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j \oplus k_1^j k_0^j \oplus k_0^j \mathrm{IV}^j \oplus k_1^j \oplus \mathrm{IV}^j$$



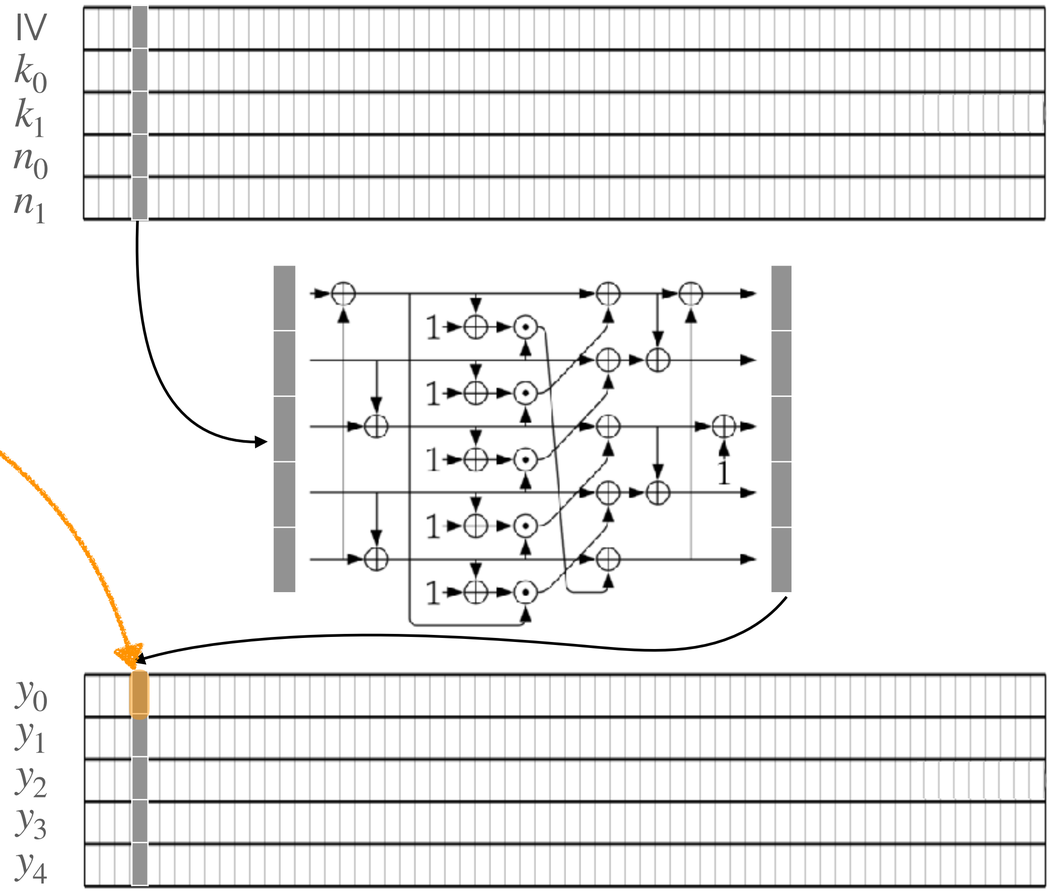
😈 Sbox output as attack point

In an Sbox computation y_0^j:

- 2 bits of key : (k_0^j, k_1^j)
- 2 bits of nonce : (n_0^j, n_1^j)

$$y_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j \oplus k_1^j k_0^j \oplus k_0^j \mathrm{IV}^j \oplus k_1^j \oplus \mathrm{IV}^j$$

	(k_0^j, k_1^j)			
(n_0^j, n_1^j)	(0,0)	(0,1)	(1,0)	(1,1)
(0,0)	0	1	1	1
(0,1)	0	1	0	0
(1,0)	1	0	0	0
(1,1)	1	0	1	1
Correlation	-1		1	



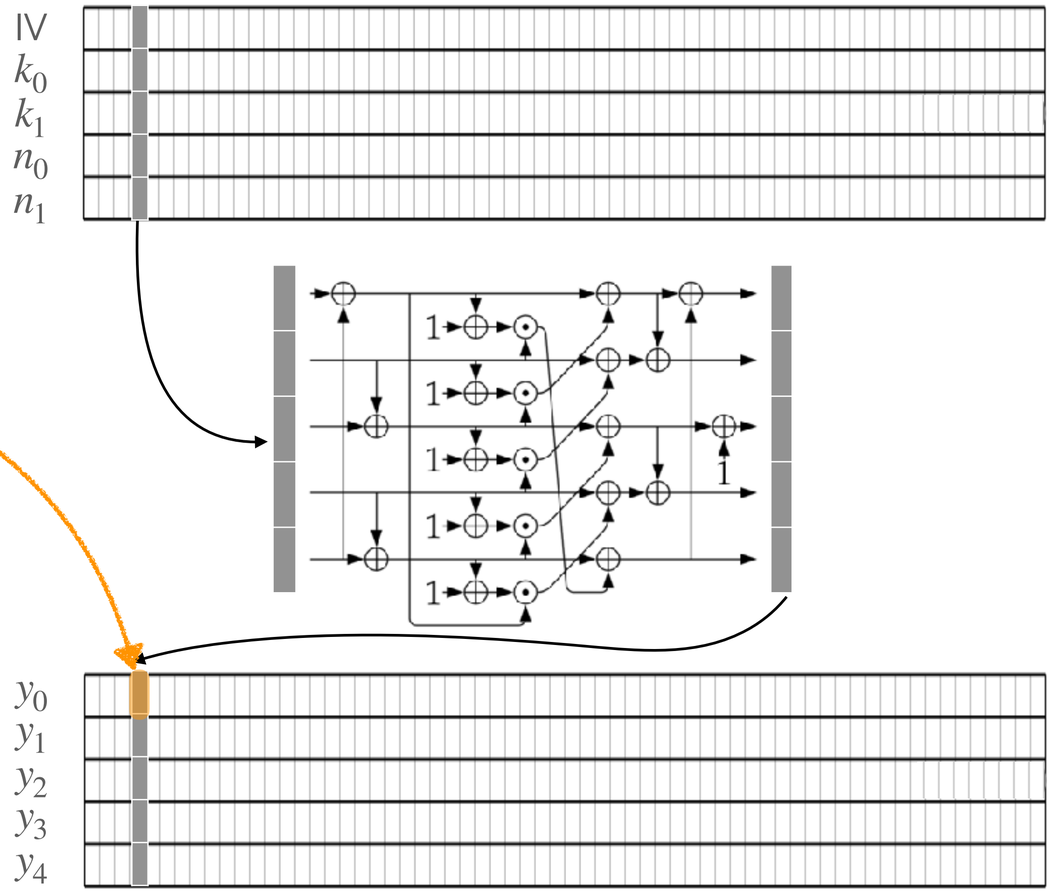
😈 Sbox output as attack point

In an Sbox computation y_0^j:

- 2 bits of key : (k_0^j, k_1^j)
- 2 bits of nonce : (n_0^j, n_1^j)

$$y_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j \oplus k_1^j k_0^j \oplus k_0^j \mathrm{IV}^j \oplus k_1^j \oplus \mathrm{IV}^j$$

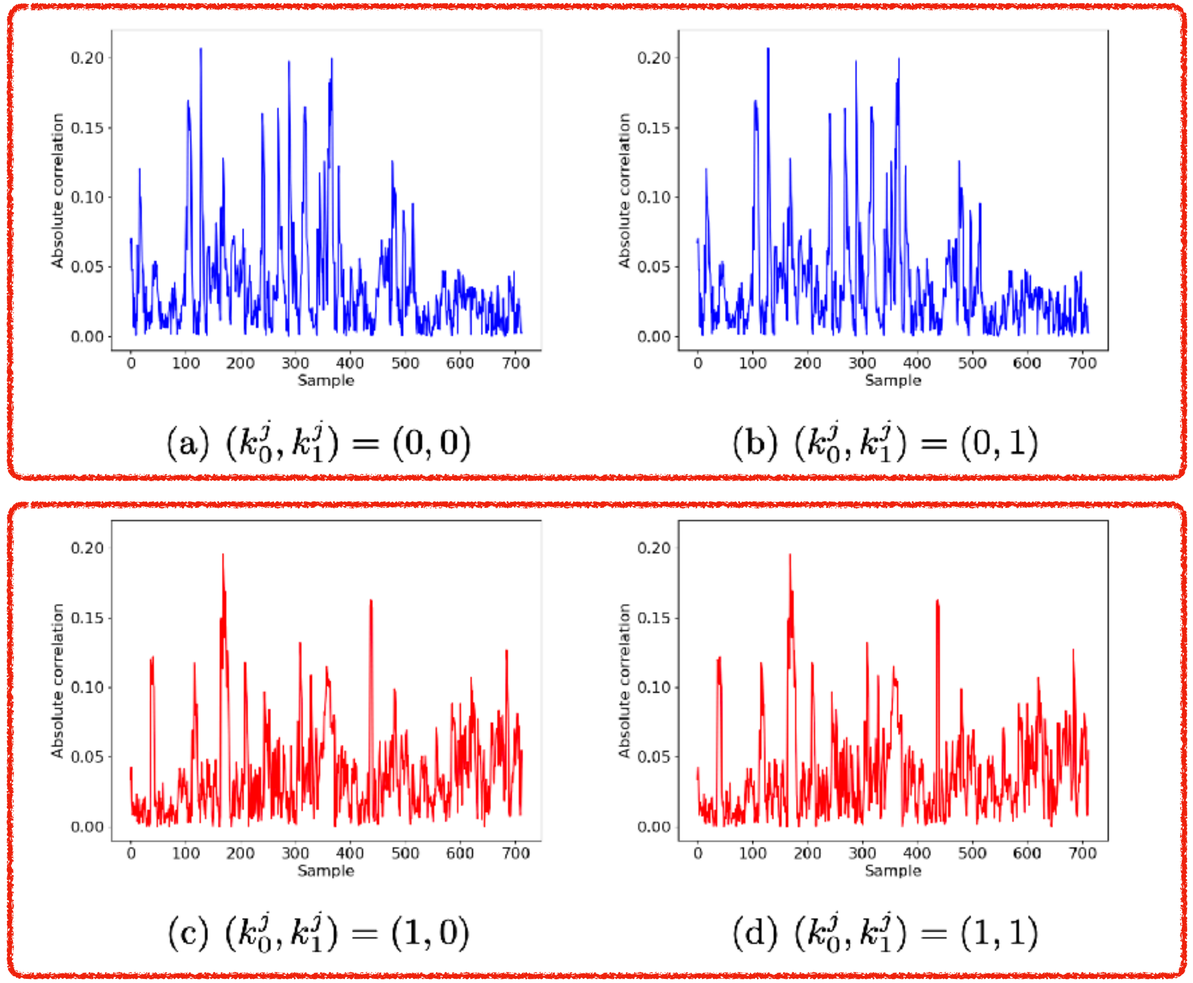
	(k_0^j, k_1^j)			
(n_0^j, n_1^j)	(0,0)	(0,1)	(1,0)	(1,1)
(0,0)	0	1	1	1
(0,1)	0	1	0	0
(1,0)	1	0	0	0
(1,1)	1	0	1	1
Correlation	-1		1	



→ Same correlation with traces → Cannot obtain unique (correct) key ❌

😈 Sbox output as attack point

Correlation traces for all key candidates

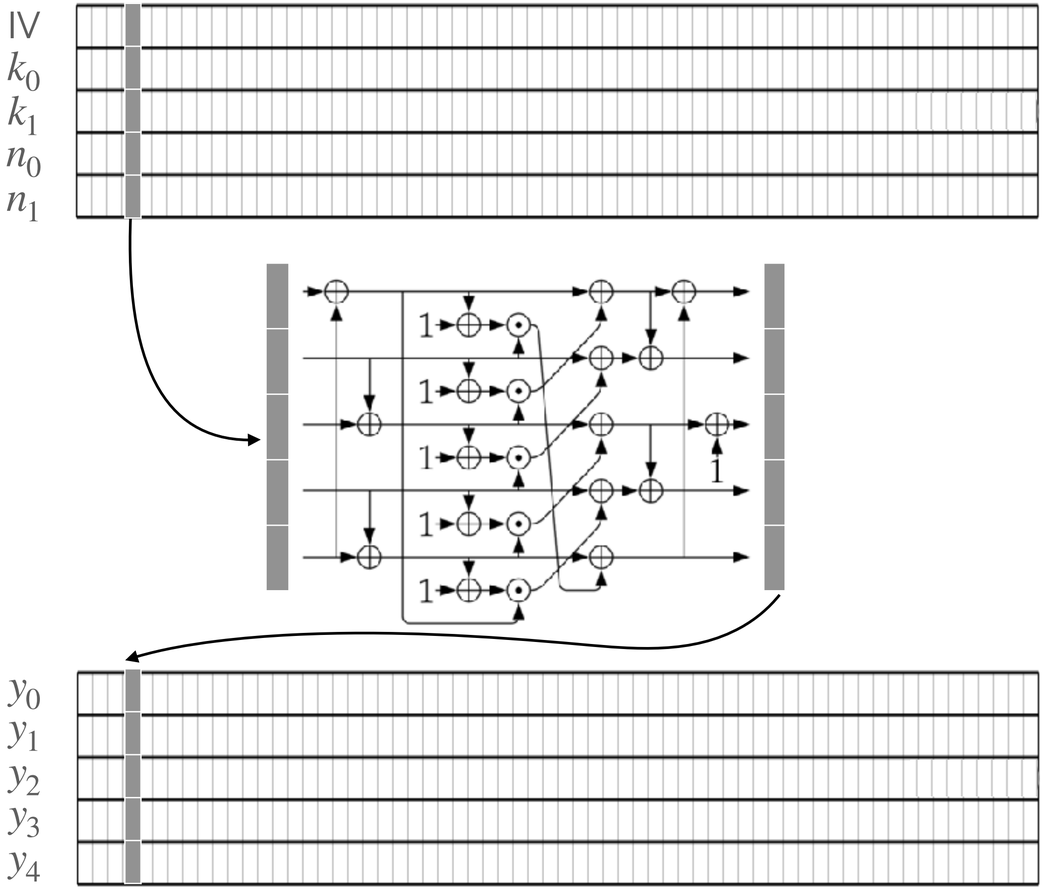


😈 Sbox output as attack point

Correlations of distributions associated to all key pairs

(k_0^j, k_1^j)	(0,0)	(0,1)	(1,0)	(1,1)
(0,0)				
(0,1)				
(1,0)				
(1,1)				

y_0^j

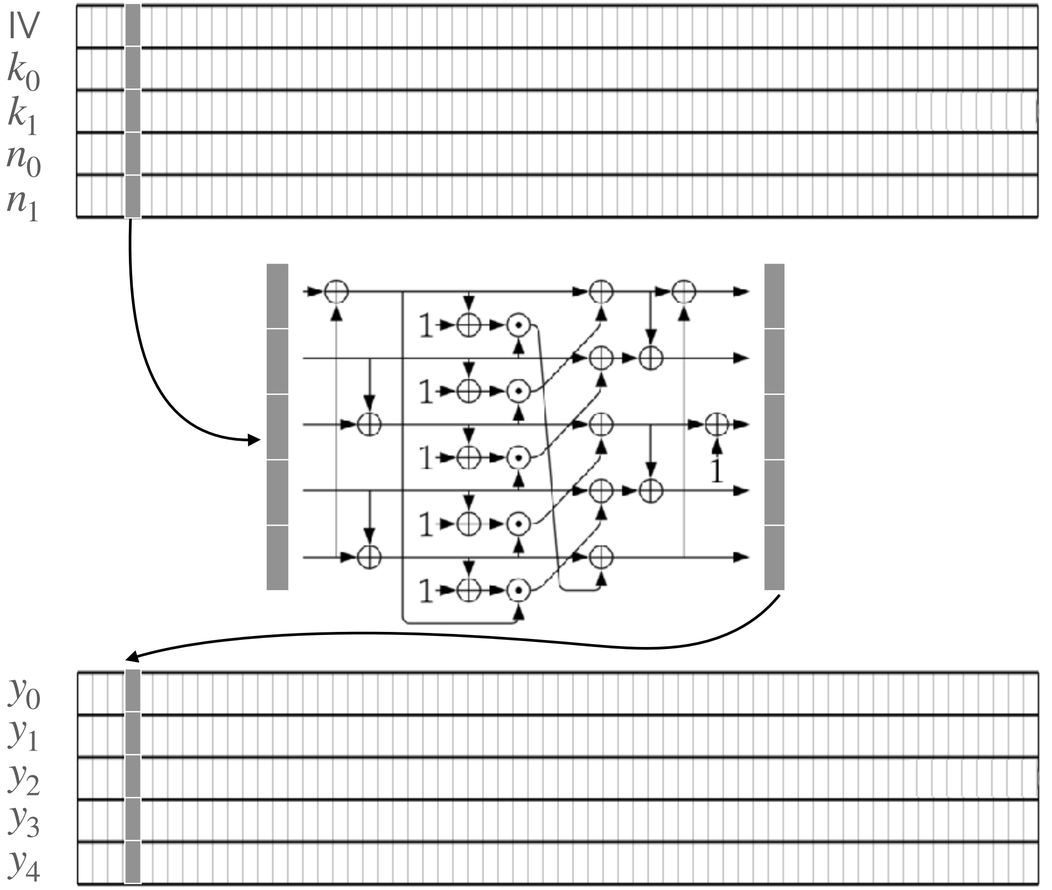


😈 Sbox output as attack point

Correlations of distributions associated to all key pairs

(k_0^j, k_1^j)	(0,0)	(0,1)	(1,0)	(1,1)
(0,0)	1			
(0,1)		1		
(1,0)			1	
(1,1)				1

y_0^j

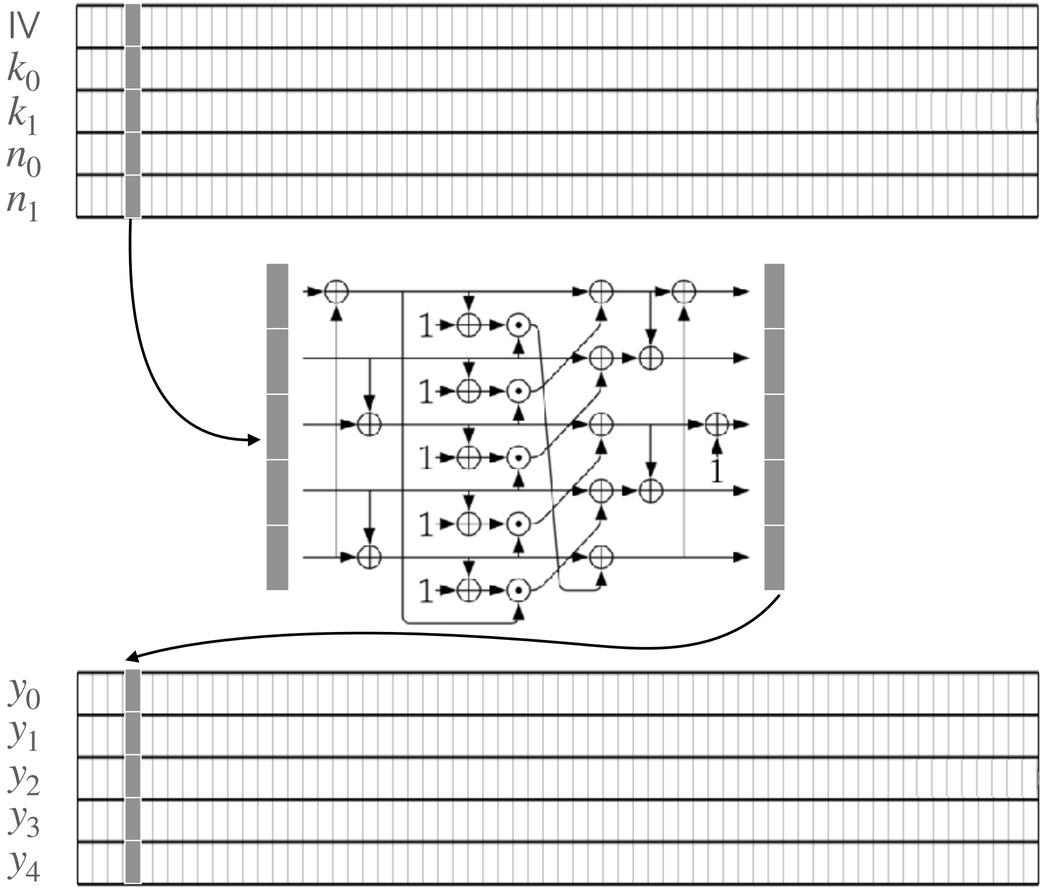


😈 Sbox output as attack point

Correlations of distributions associated to all key pairs

(k_0^j, k_1^j)	(0,0)	(0,1)	(1,0)	(1,1)
(0,0)	1	1	-	-
(0,1)	1	1	-	-
(1,0)	-	-	1	1
(1,1)	-	-	1	1

y_0^j



😈 Sbox output as attack point

Correlations of distributions associated to all key pairs

(k_0^j, k_1^j)	(0,0)	(0,1)	(1,0)	(1,1)
(0,0)	1	1	-	-
(0,1)	1	1	-	-
(1,0)	-	-	1	1
(1,1)	-	-	1	1

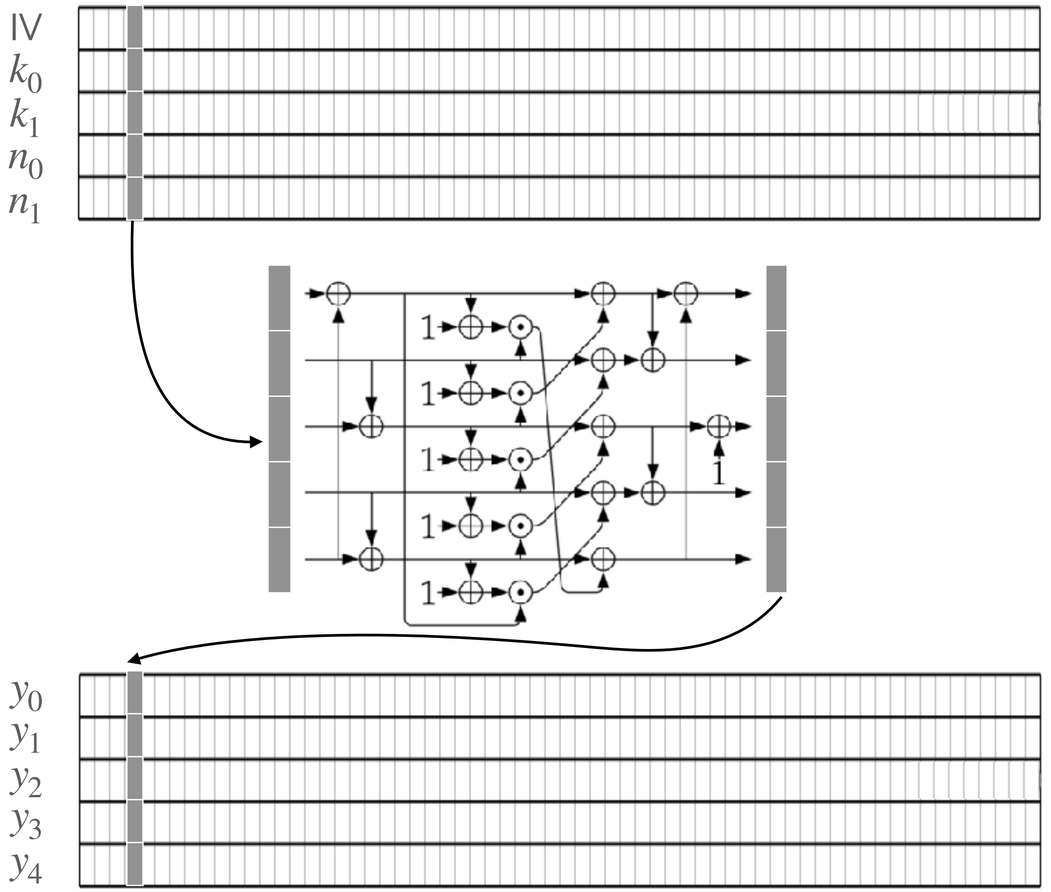
y_0^j

(k_0^j, k_1^j)	(0,0)	(0,1)	(1,0)	(1,1)
(0,0)	1	-	-	1
(0,1)	-	1	1	-
(1,0)	-	1	1	-
(1,1)	1	-	-	1

y_1^j

(k_0^j, k_1^j)	(0,0)	(0,1)	(1,0)	(1,1)
(0,0)	1	1	1	1
(0,1)	1	1	1	1
(1,0)	1	1	1	1
(1,1)	1	1	1	1

y_2^j and y_3^j



😈 Sbox output as attack point

Correlations of distributions associated to all key pairs

(k_0^j, k_1^j)	(0,0)	(0,1)	(1,0)	(1,1)
(0,0)	1	1	-	-
(0,1)	1	1	-	-
(1,0)	-	-	1	1
(1,1)	-	-	1	1

y_0^j

(k_0^j, k_1^j)	(0,0)	(0,1)	(1,0)	(1,1)
(0,0)	1	-	-	1
(0,1)	-	1	1	-
(1,0)	-	1	1	-
(1,1)	1	-	-	1

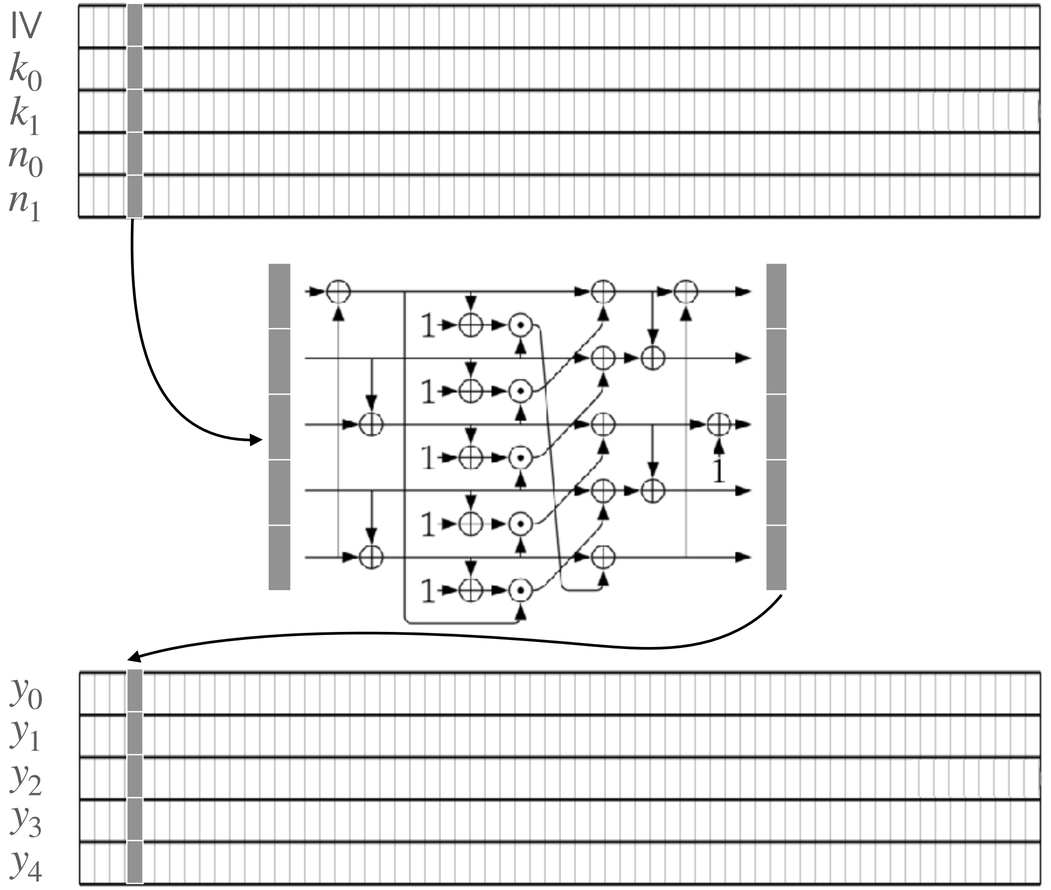
y_1^j

(k_0^j, k_1^j)	(0,0)	(0,1)	(1,0)	(1,1)
(0,0)	1	1	1	1
(0,1)	1	1	1	1
(1,0)	1	1	1	1
(1,1)	1	1	1	1

y_2^j and y_3^j

k_0^j	0	1
0	1	-
1	-	1

y_4^j



😈 Sbox output as attack point

Correlations of distributions associated to all key pairs

(k_0^j, k_1^j)	(0,0)	(0,1)	(1,0)	(1,1)
(0,0)	1	1	-	-
(0,1)	1	1	-	-
(1,0)	-	-	1	1
(1,1)	-	-	1	1

y_0^j

(k_0^j, k_1^j)	(0,0)	(0,1)	(1,0)	(1,1)
(0,0)	1	-	-	1
(0,1)	-	1	1	-
(1,0)	-	1	1	-
(1,1)	1	-	-	1

y_1^j

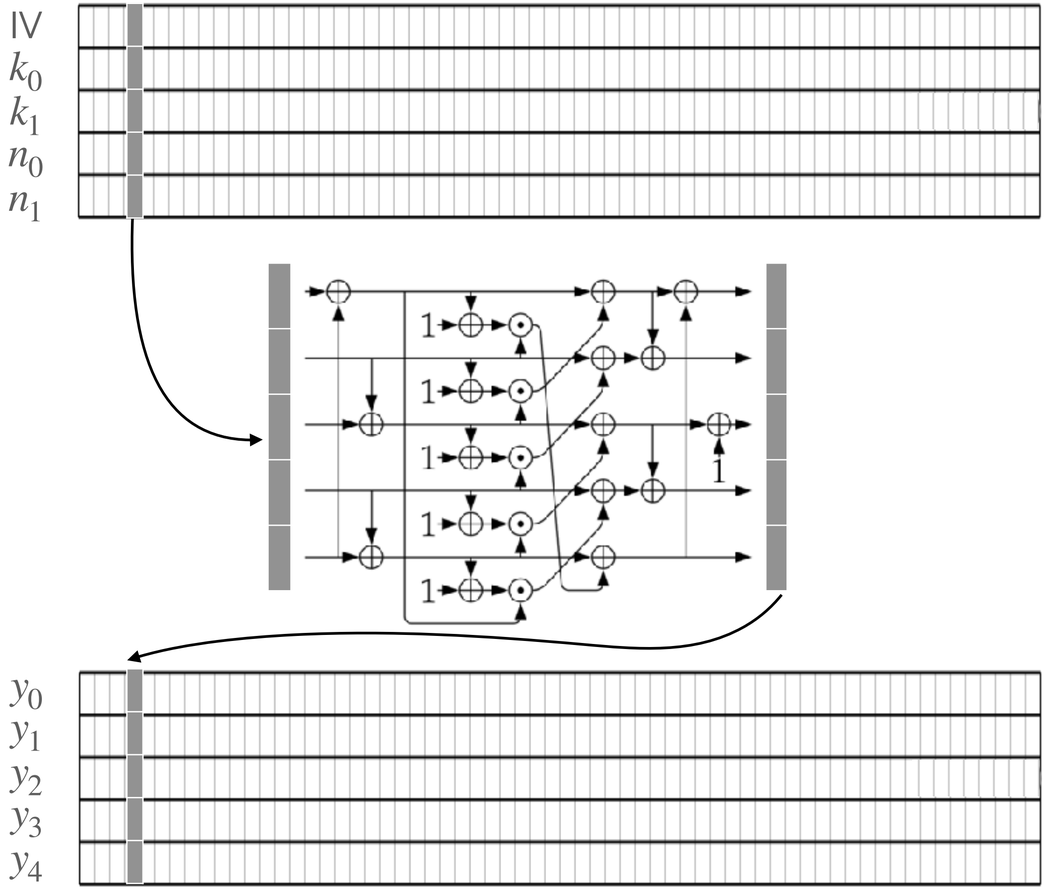
(k_0^j, k_1^j)	(0,0)	(0,1)	(1,0)	(1,1)
(0,0)	1	1	1	1
(0,1)	1	1	1	1
(1,0)	1	1	1	1
(1,1)	1	1	1	1

y_2^j and y_3^j

k_0^j	0	1
0	1	-
1	-	1

y_4^j

y_4^j seems good, but we'll see later



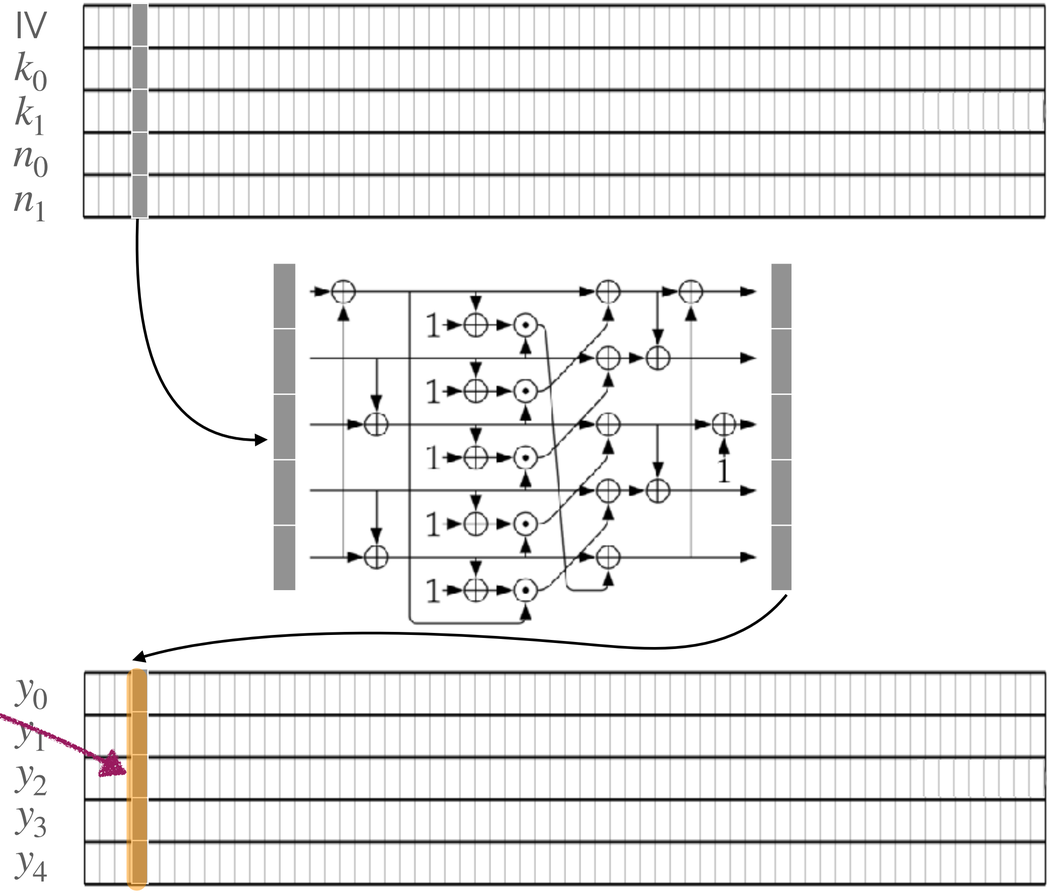
😈 Sbox output as attack point

Correlations of distributions associated to all key pairs

(k_0^j, k_1^j)	(0,0)	(0,1)	(1,0)	(1,1)
(0,0)	1.00			
(0,1)		1.00		
(1,0)			1.00	
(1,1)				1.00

HW of Sbox output

(registers store columns, not the intended design)



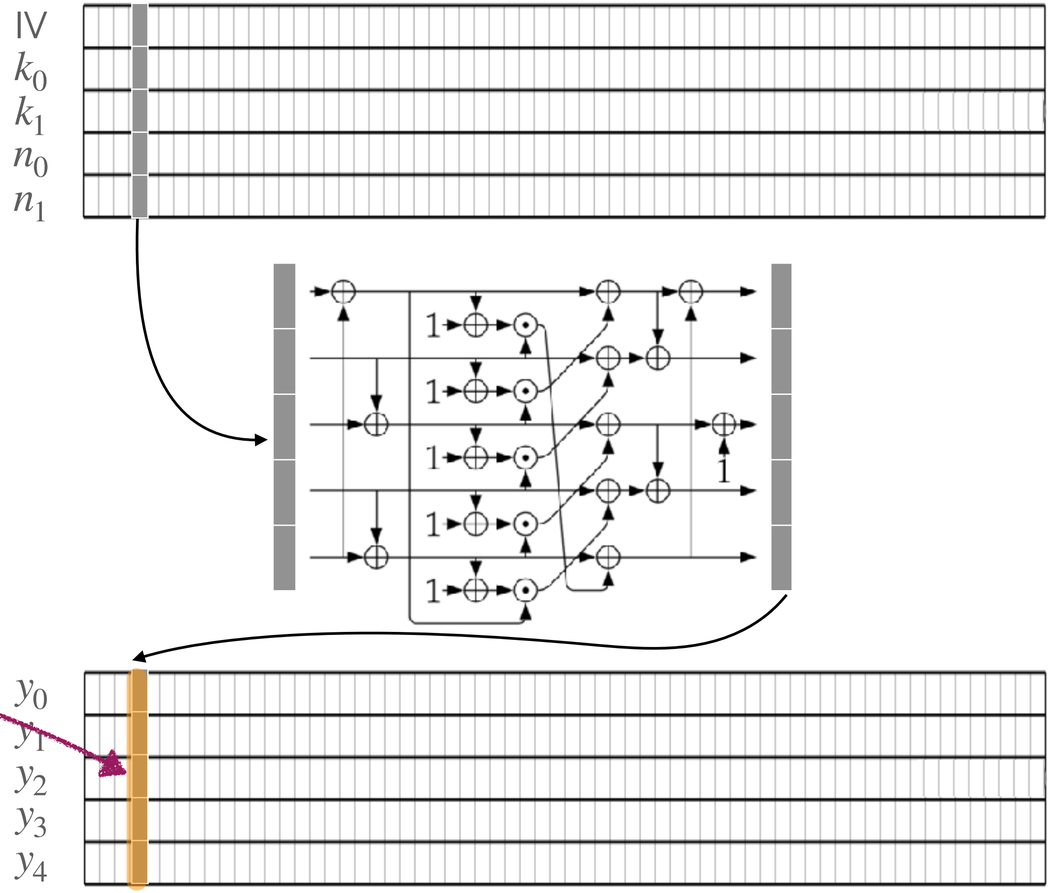
😈 Sbox output as attack point

Correlations of distributions associated to all key pairs

(k_0^j, k_1^j)	(0,0)	(0,1)	(1,0)	(1,1)
(0,0)	1.00	0.15	0.89	0.87
(0,1)	0.15	1.00	0.48	0.09
(1,0)	0.89	0.48	1.00	0.90
(1,1)	0.87	0.09	0.90	1.00

HW of Sbox output

(registers store columns, not the intended design)



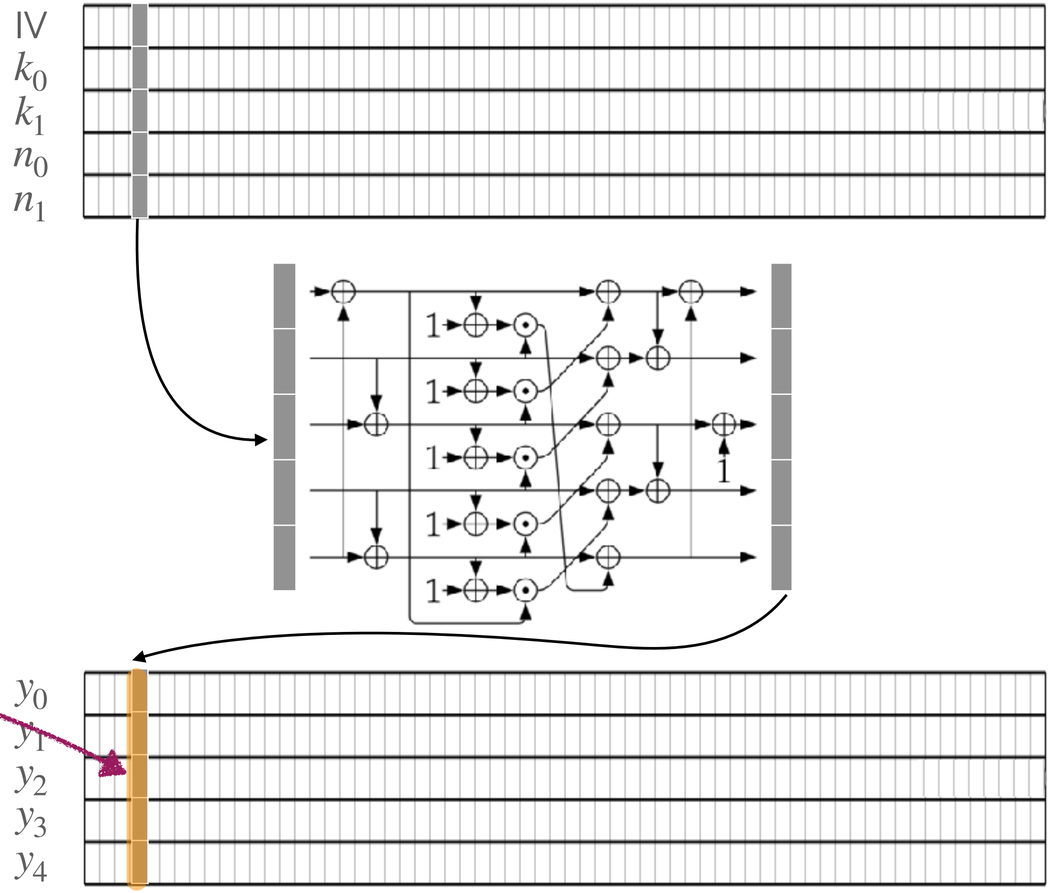
😈 Sbox output as attack point

Correlations of distributions associated to all key pairs

(k_0^j, k_1^j)	(0,0)	(0,1)	(1,0)	(1,1)
(0,0)	1.00	0.15	0.89	0.87
(0,1)	0.15	1.00	0.48	0.09
(1,0)	0.89	0.48	1.00	0.90
(1,1)	0.87	0.09	0.90	1.00

HW of Sbox output

(registers store columns, not the intended design)



😈 Sbox output as attack point

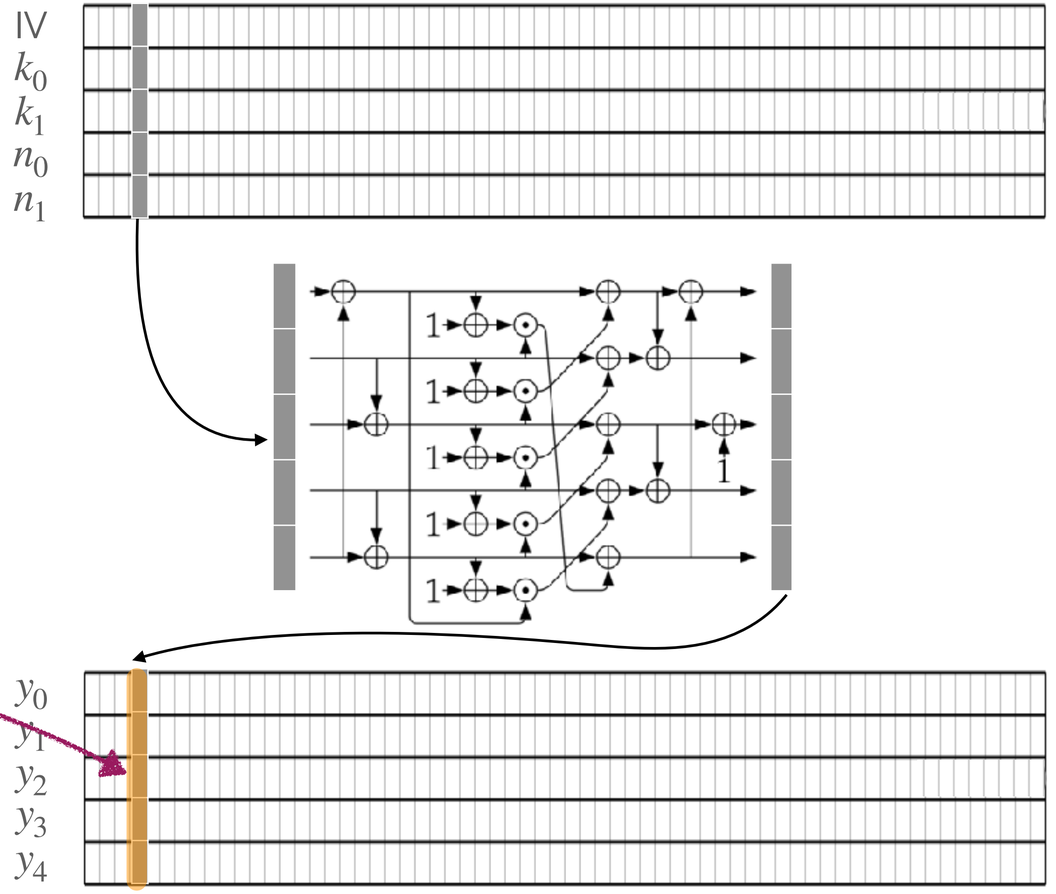
Correlations of distributions associated to all key pairs

(k_0^j, k_1^j)	(0,0)	(0,1)	(1,0)	(1,1)
(0,0)	1.00	0.15	0.89	0.87
(0,1)	0.15	1.00	0.48	0.09
(1,0)	0.89	0.48	1.00	0.90
(1,1)	0.87	0.09	0.90	1.00

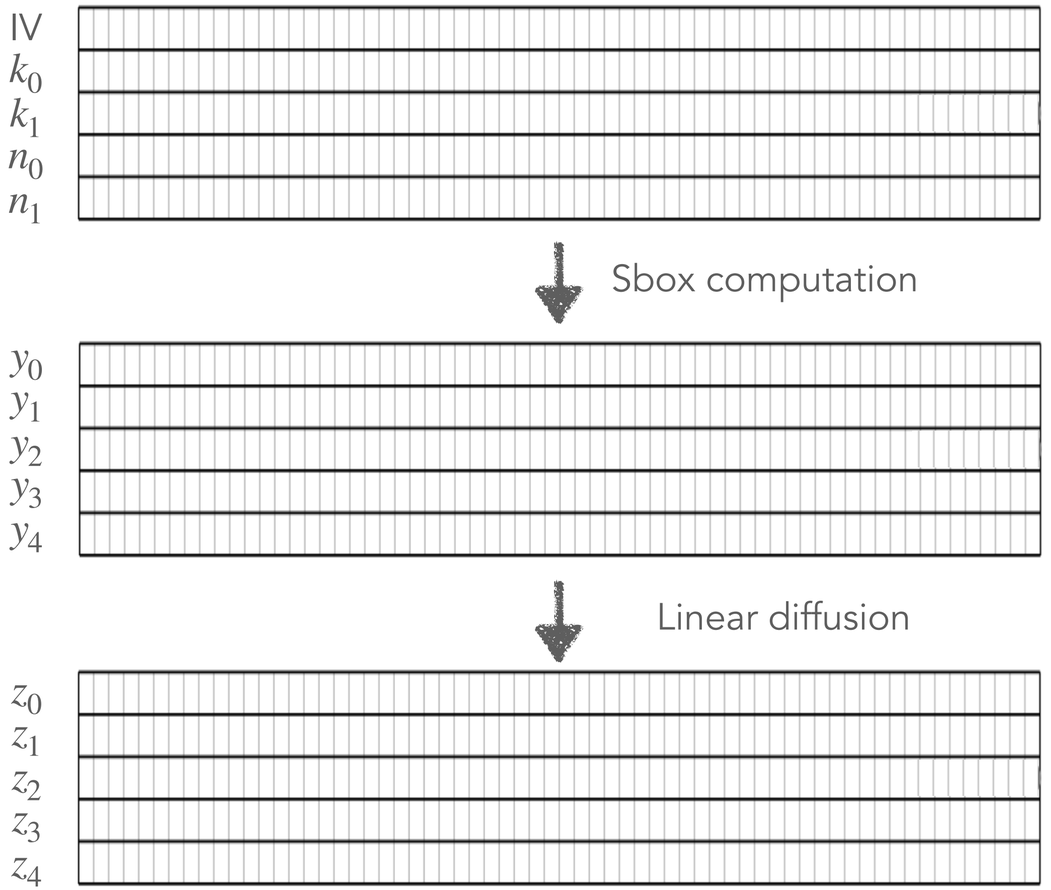
HW of Sbox output

(registers store columns, not the intended design)

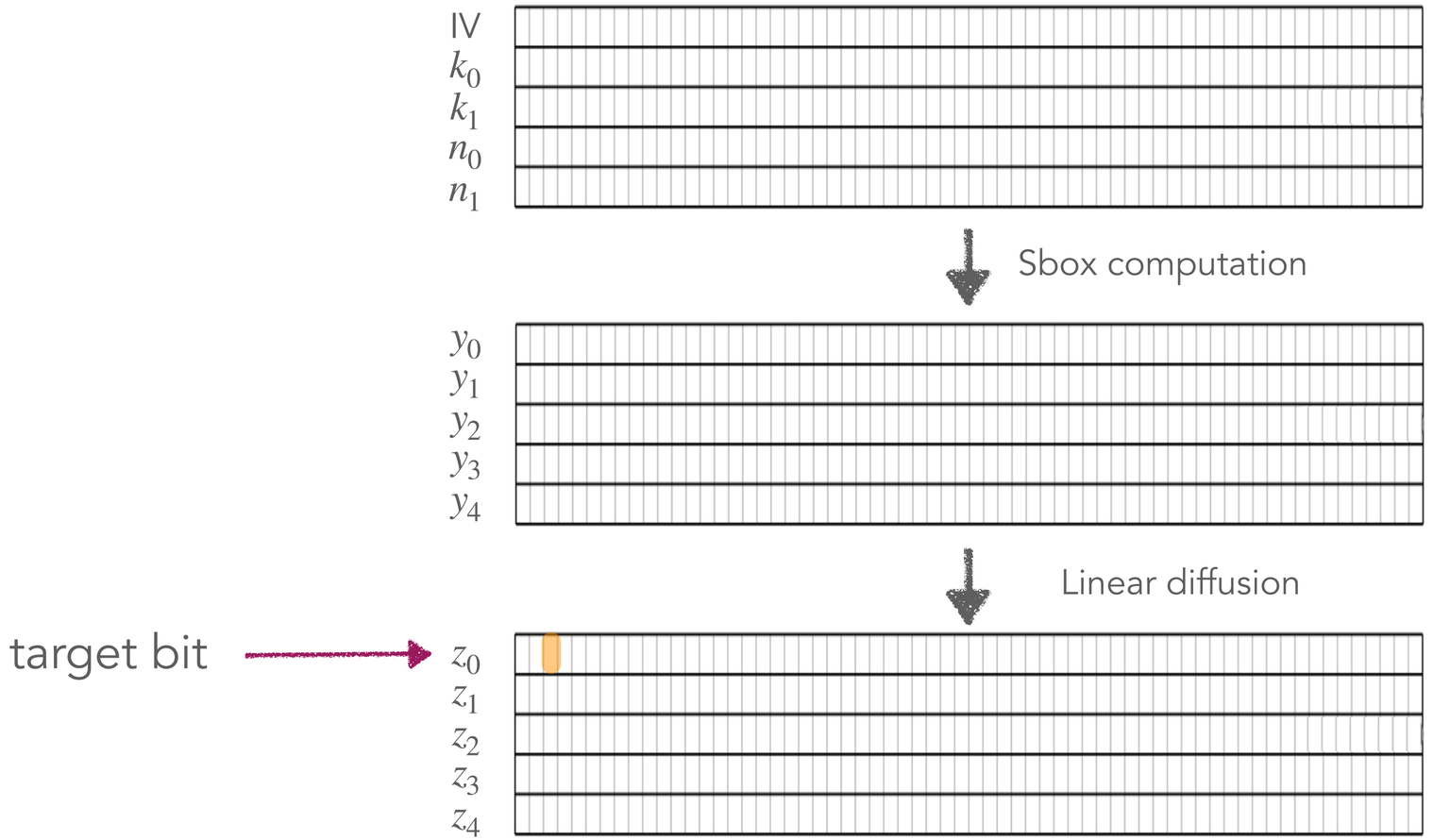
❌ Not effective for CPA attacks



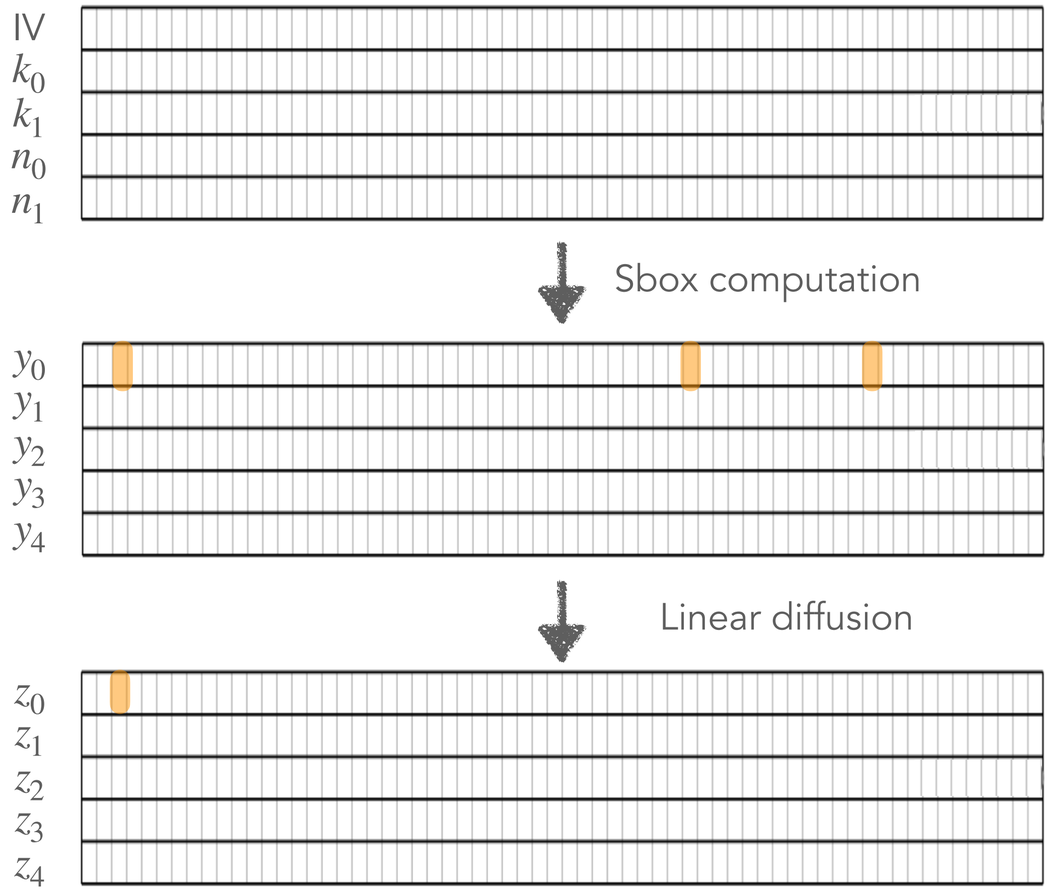
😈 Linear layer output as attack point



😈 Linear layer output as attack point



😈 Linear layer output as attack point

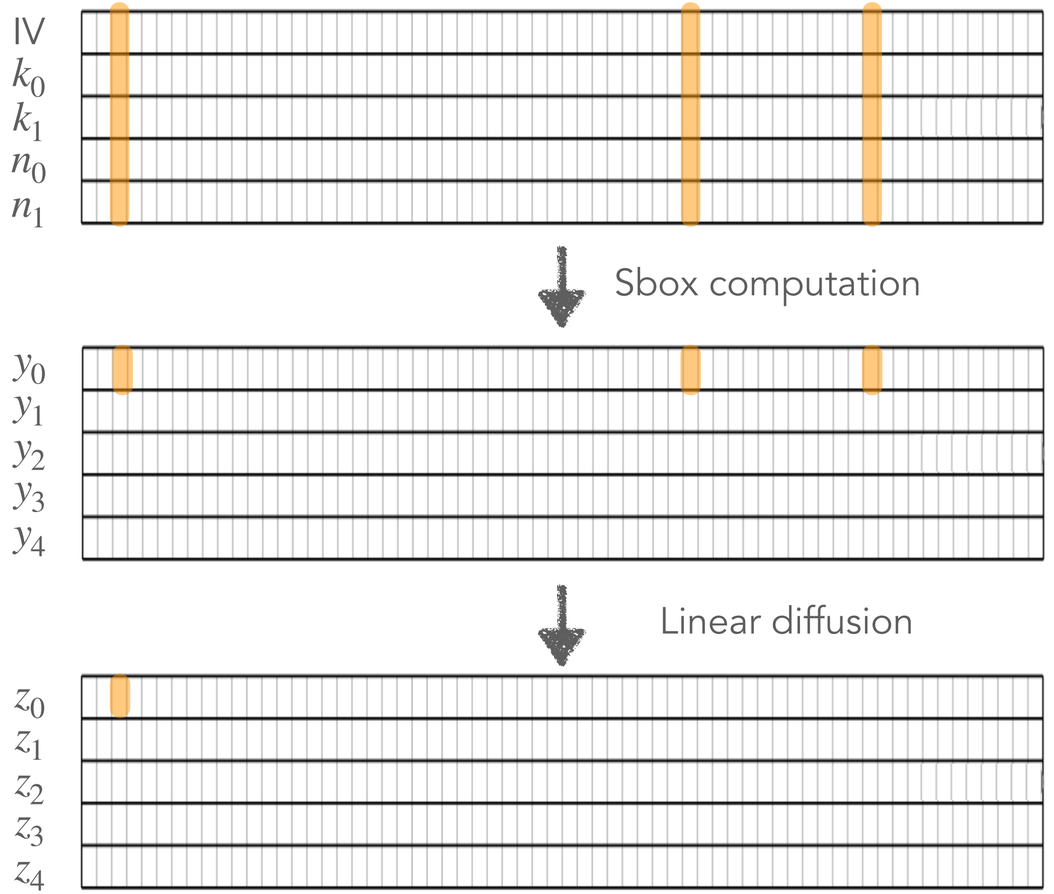


target bit

$$z_0^j = y_0^j \oplus y_0^{j+19} \oplus y_0^{j+28}$$

😈 Linear layer output as attack point

$$y_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j \oplus k_1^j k_0^j \oplus k_0^j \mathrm{IV}^j \oplus k_1^j \oplus \mathrm{IV}^j$$

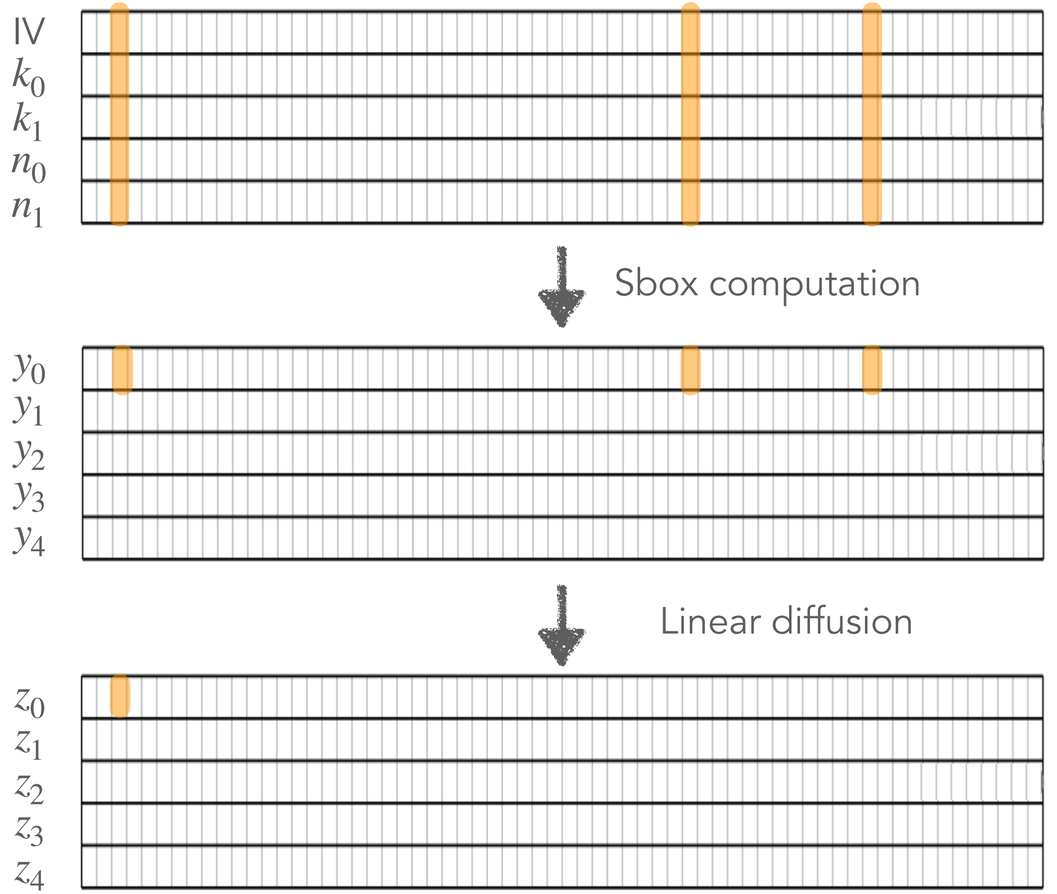


target bit

$$z_0^j = y_0^j \oplus y_0^{j+19} \oplus y_0^{j+28}$$

😈 Linear layer output as attack point

$$y_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j \oplus k_1^j k_0^j \oplus k_0^j \mathrm{IV}^j \oplus k_1^j \oplus \mathrm{IV}^j$$



target bit

$$z_0^j = y_0^j \oplus y_0^{j+19} \oplus y_0^{j+28}$$

😈 Linear layer output as attack point

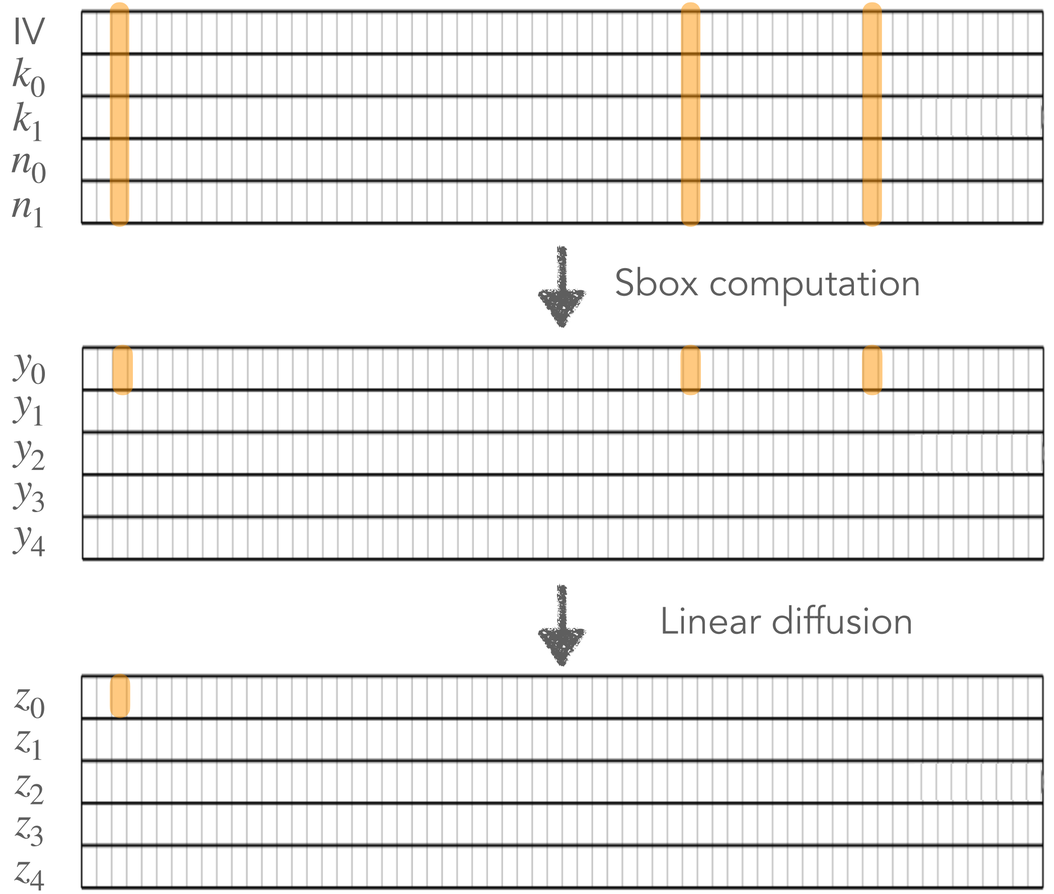
$$y_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j \oplus k_1^j k_0^j \oplus k_0^j \mathrm{IV}^j \oplus k_1^j \oplus \mathrm{IV}^j$$

[SD17]: constant part contributes a constant amount to power consumption

$$\tilde{y}_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j$$

target bit

$$z_0^j = y_0^j \oplus y_0^{j+19} \oplus y_0^{j+28}$$



😈 Linear layer output as attack point

$$y_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j \oplus k_1^j k_0^j \oplus k_0^j \mathrm{IV}^j \oplus k_1^j \oplus \mathrm{IV}^j$$

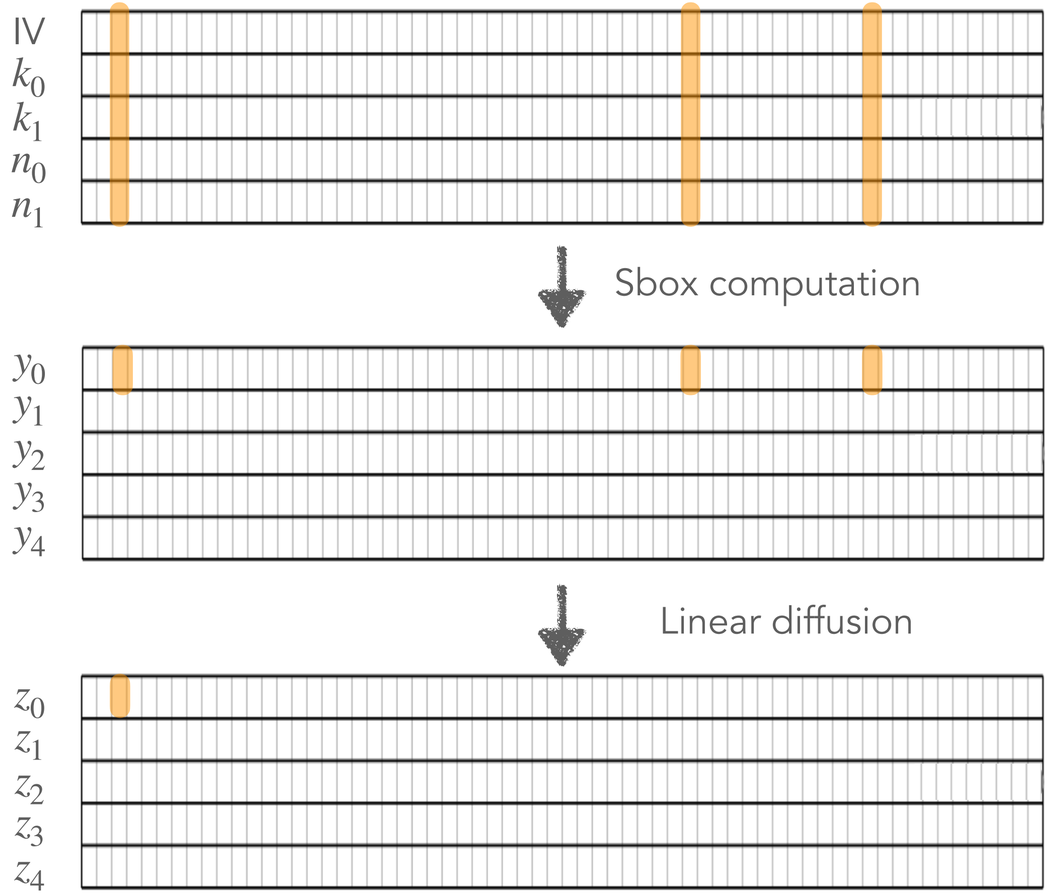
[SD17]: constant part contributes a constant amount to power consumption

$$\tilde{y}_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j$$

	k_0^j	
(n_0^j, n_1^j)	0	1
(0,0)	0	1
(0,1)	0	0
(1,0)	1	0
(1,1)	1	1
Correlation	0	

target bit

$$z_0^j = y_0^j \oplus y_0^{j+19} \oplus y_0^{j+28}$$

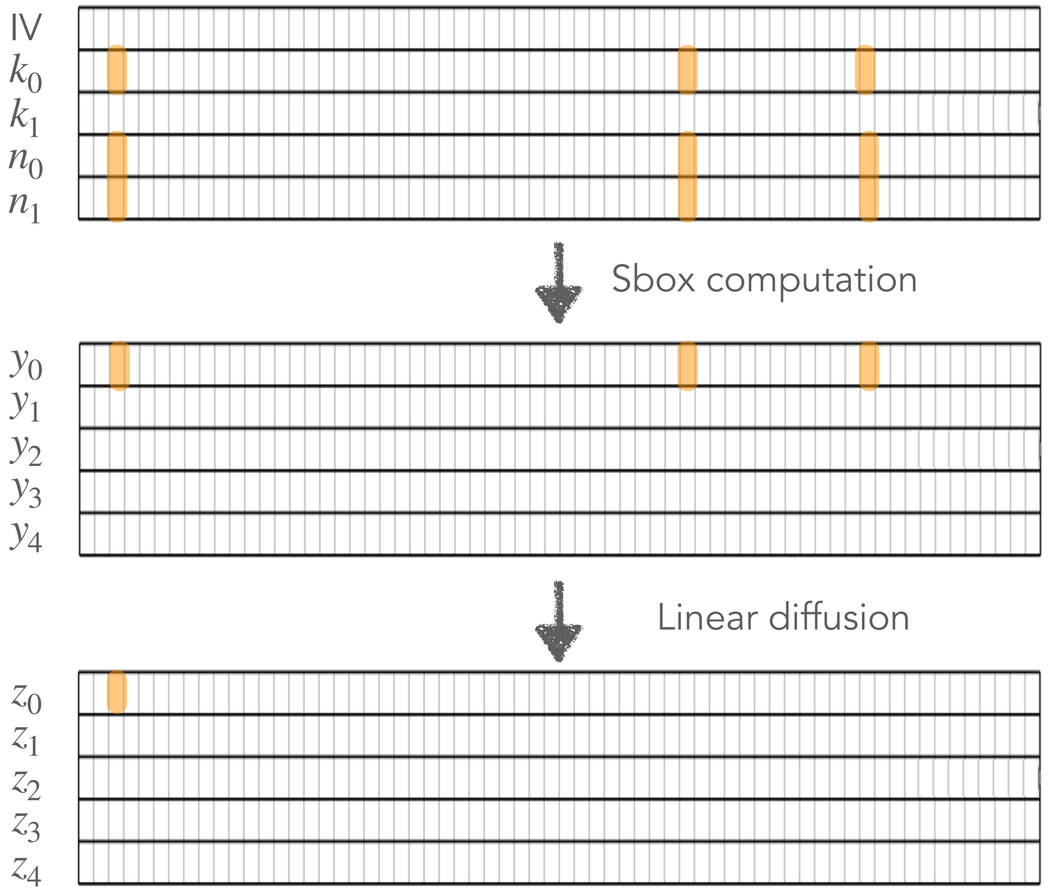


😈 Linear layer output as attack point

$$y_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j \oplus k_1^j k_0^j \oplus k_0^j \mathrm{IV}^j \oplus k_1^j \oplus \mathrm{IV}^j$$

[SD17]: constant part contributes a constant amount to power consumption

$$\tilde{y}_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j$$



😈 Linear layer output as attack point

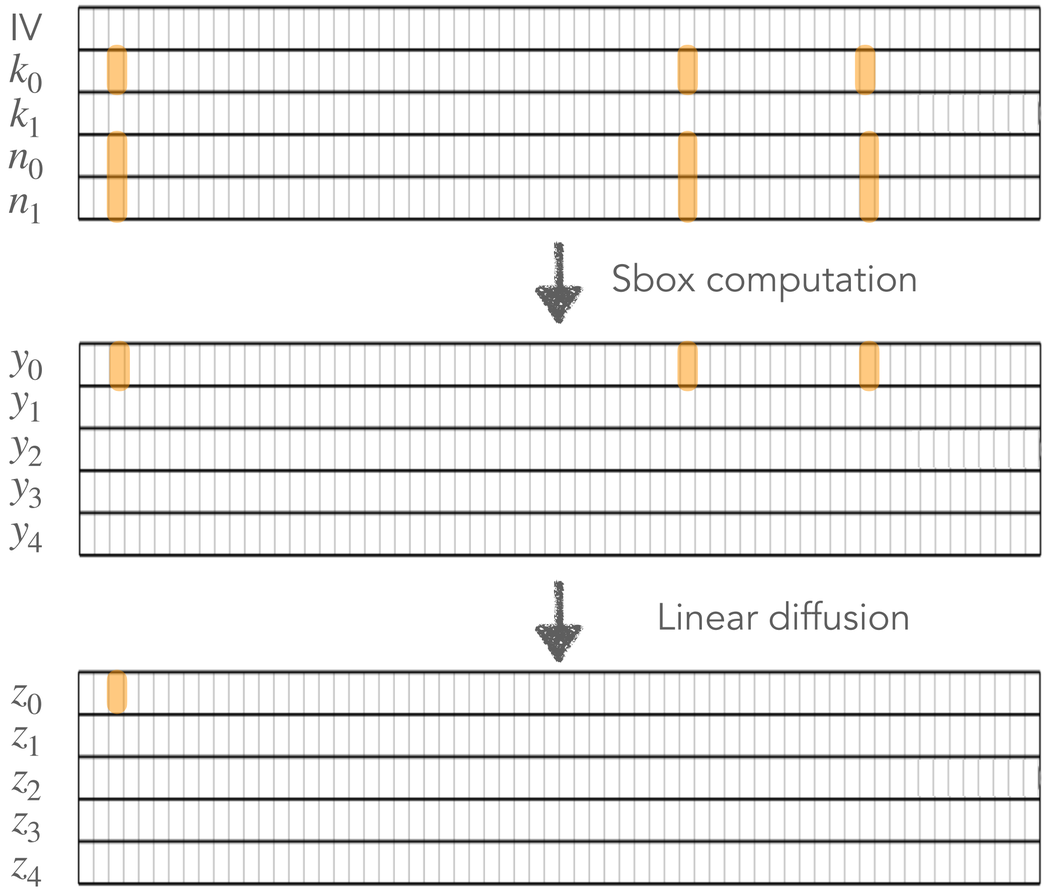
$$y_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j \oplus k_1^j k_0^j \oplus k_0^j \mathrm{IV}^j \oplus k_1^j \oplus \mathrm{IV}^j$$

[SD17]: constant part contributes a constant amount to power consumption

$$\tilde{y}_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j$$

Similarly:

$$\tilde{y}_0^{j+19} = k_0^{j+19}(n_1^{j+19} \oplus 1) \oplus n_0^{j+19}$$
$$\tilde{y}_0^{j+28} = k_0^{j+28}(n_1^{j+28} \oplus 1) \oplus n_0^{j+28}$$



😈 Linear layer output as attack point

$$y_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j \oplus k_1^j k_0^j \oplus k_0^j \mathrm{IV}^j \oplus k_1^j \oplus \mathrm{IV}^j$$

[SD17]: constant part contributes a constant amount to power consumption

$$\tilde{y}_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j$$

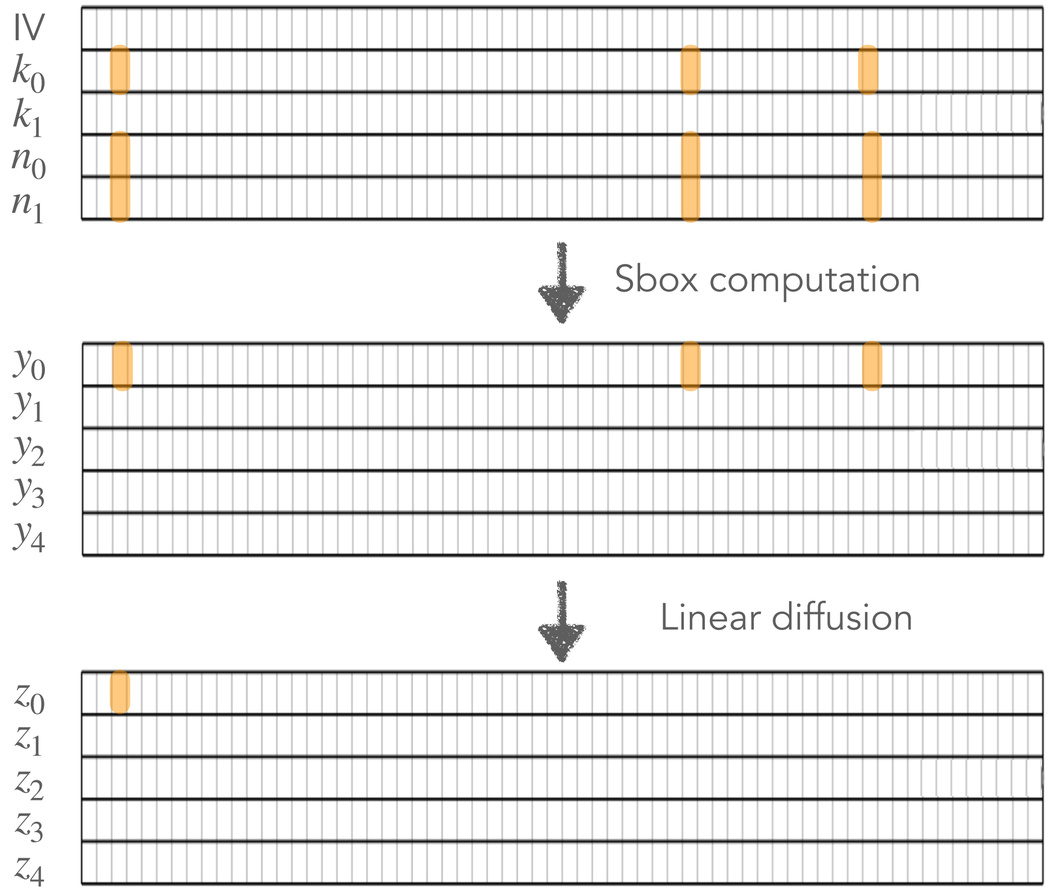
Similarly:

$$\tilde{y}_0^{j+19} = k_0^{j+19}(n_1^{j+19} \oplus 1) \oplus n_0^{j+19}$$
$$\tilde{y}_0^{j+28} = k_0^{j+28}(n_1^{j+28} \oplus 1) \oplus n_0^{j+28}$$

Linear computation:

$$\tilde{z}_0^j = \tilde{y}_0^j \oplus \tilde{y}_0^{j+19} \oplus \tilde{y}_0^{j+28}$$

target bit



😈 Linear layer output as attack point

$$y_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j \oplus k_1^j k_0^j \oplus k_0^j \mathrm{IV}^j \oplus k_1^j \oplus \mathrm{IV}^j$$

[SD17]: constant part contributes a constant amount to power consumption

$$\tilde{y}_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j$$

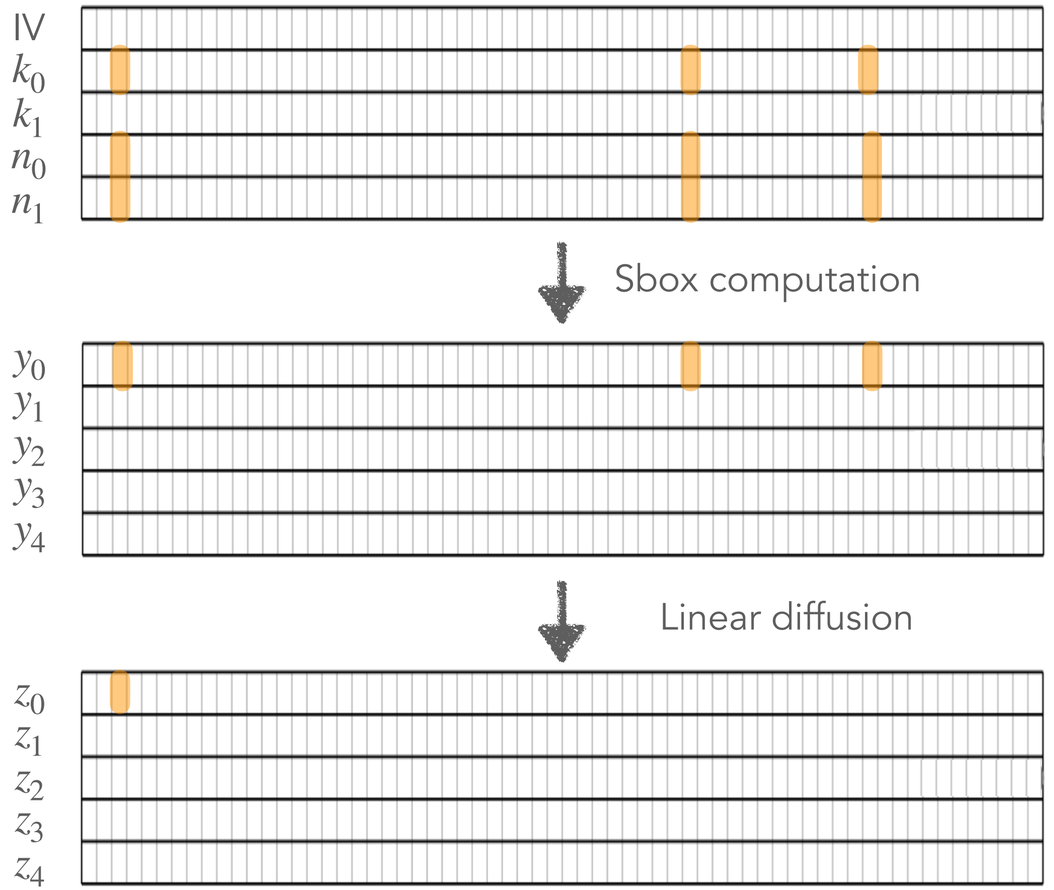
Similarly:

$$\tilde{y}_0^{j+19} = k_0^{j+19}(n_1^{j+19} \oplus 1) \oplus n_0^{j+19}$$
$$\tilde{y}_0^{j+28} = k_0^{j+28}(n_1^{j+28} \oplus 1) \oplus n_0^{j+28}$$

Linear computation:

$$\tilde{z}_0^j = \tilde{y}_0^j \oplus \tilde{y}_0^{j+19} \oplus \tilde{y}_0^{j+28}$$
$$\tilde{z}_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j$$
$$\oplus\, k_0^{j+19}(n_1^{j+19} \oplus 1) \oplus n_0^{j+19}$$
$$\oplus\, k_0^{j+28}(n_1^{j+28} \oplus 1) \oplus n_0^{j+28}$$

target bit →



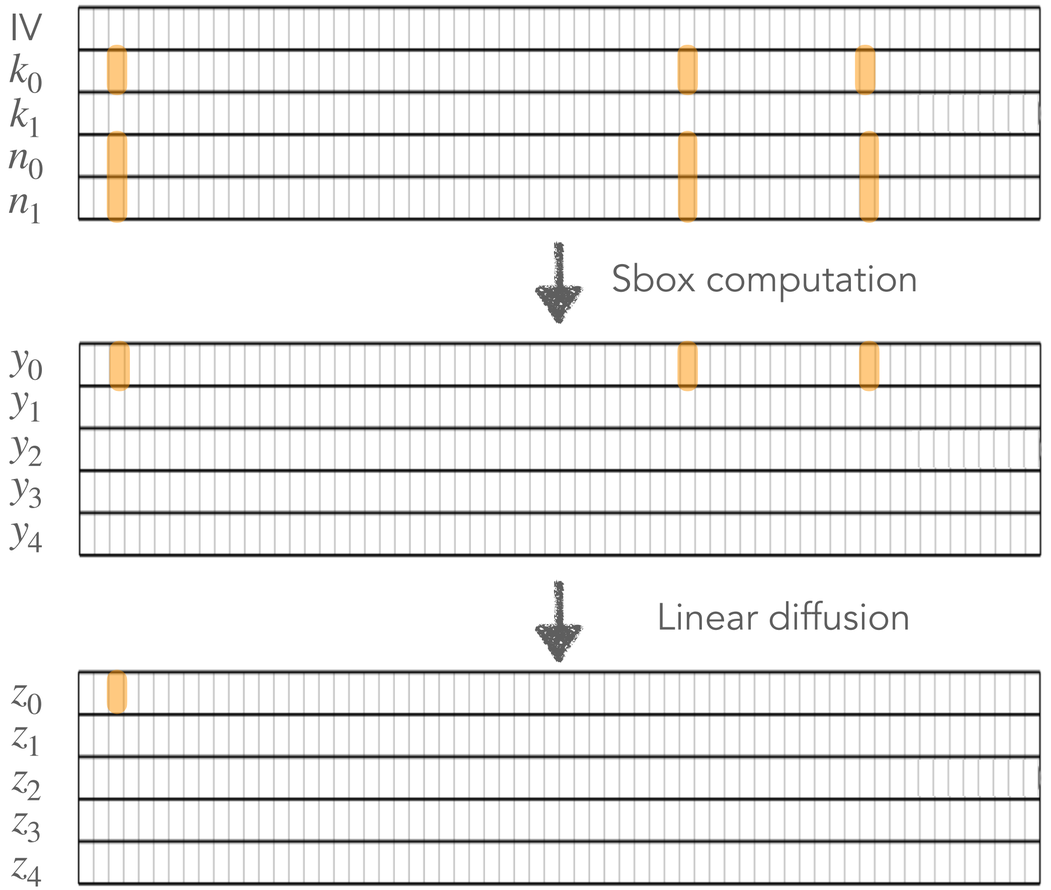
😈 Linear layer output as attack point

Linear computation:

$$\tilde{z}_0^j = \tilde{y}_0^j \oplus \tilde{y}_0^{j+19} \oplus \tilde{y}_0^{j+28}$$

$$\tilde{z}_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j \oplus k_0^{j+19}(n_1^{j+19} \oplus 1) \oplus n_0^{j+19} \oplus k_0^{j+28}(n_1^{j+28} \oplus 1) \oplus n_0^{j+28}$$

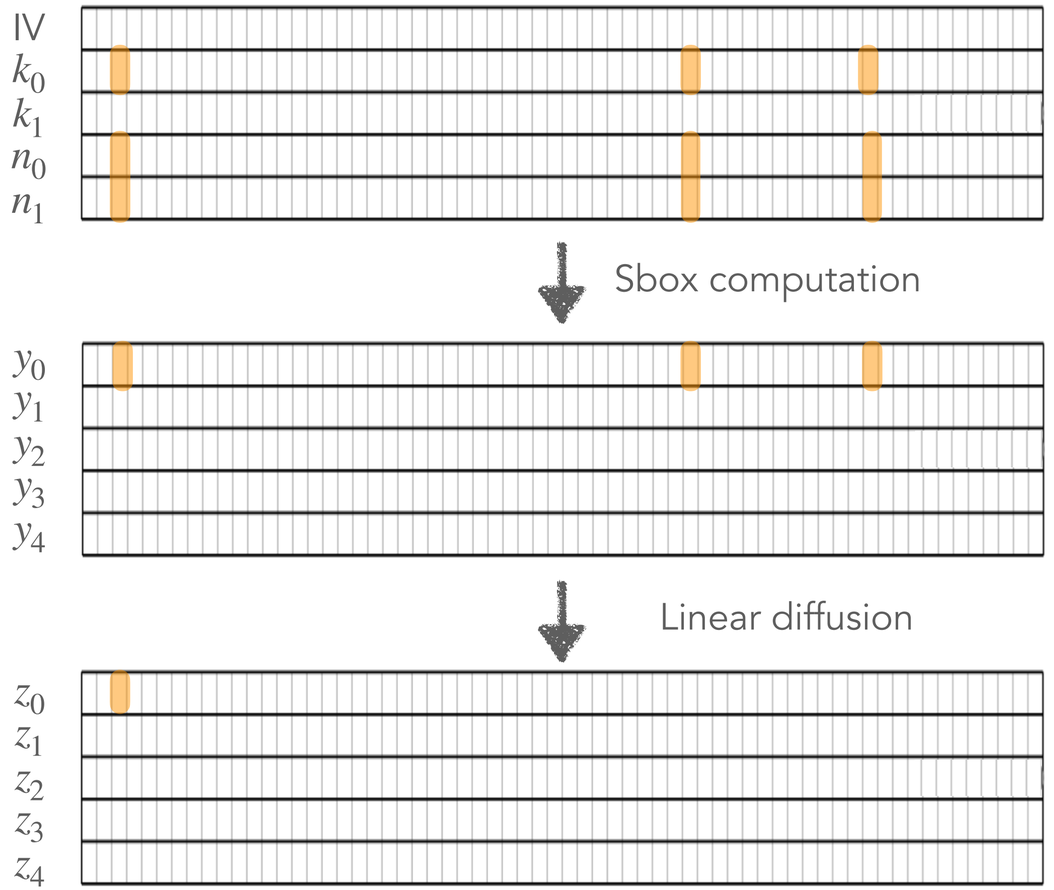
target bit →



😈 Linear layer output as attack point

Correlations of distributions associated to all key pairs

$(k_0^j, k_0^{j+19}, k_0^{j+28})$	(0,0,0)	(0,0,1)	(0,1,0)	(0,1,1)	(1,0,0)	(1,0,1)	(1,1,0)	(1,1,1)
(0,0,0)	1	-	-	-	-	-	-	-
(0,0,1)	-	1	-	-	-	-	-	-
(0,1,0)	-	-	1	-	-	-	-	-
(0,1,1)	-	-	-	1	-	-	-	-
(1,0,0)	-	-	-	-	1	-	-	-
(1,0,1)	-	-	-	-	-	1	-	-
(1,1,0)	-	-	-	-	-	-	1	-
(1,1,1)	-	-	-	-	-	-	-	1



Linear computation:

$$\tilde{z}_0^j = \tilde{y}_0^j \oplus \tilde{y}_0^{j+19} \oplus \tilde{y}_0^{j+28}$$

$$\tilde{z}_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j \oplus k_0^{j+19}(n_1^{j+19} \oplus 1) \oplus n_0^{j+19} \oplus k_0^{j+28}(n_1^{j+28} \oplus 1) \oplus n_0^{j+28}$$

😈 Linear layer output as attack point

Correlations of distributions associated to all key pairs

$(k_0^j, k_0^{j+19}, k_0^{j+28})$	(0,0,0)	(0,0,1)	(0,1,0)	(0,1,1)	(1,0,0)	(1,0,1)	(1,1,0)	(1,1,1)
(0,0,0)	1	-	-	-	-	-	-	-
(0,0,1)	-	1	-	-	-	-	-	-
(0,1,0)	-	-	1	-	-	-	-	-
(0,1,1)	-	-	-	1	-	-	-	-
(1,0,0)	-	-	-	-	1	-	-	-
(1,0,1)	-	-	-	-	-	1	-	-
(1,1,0)	-	-	-	-	-	-	1	-
(1,1,1)	-	-	-	-	-	-	-	1

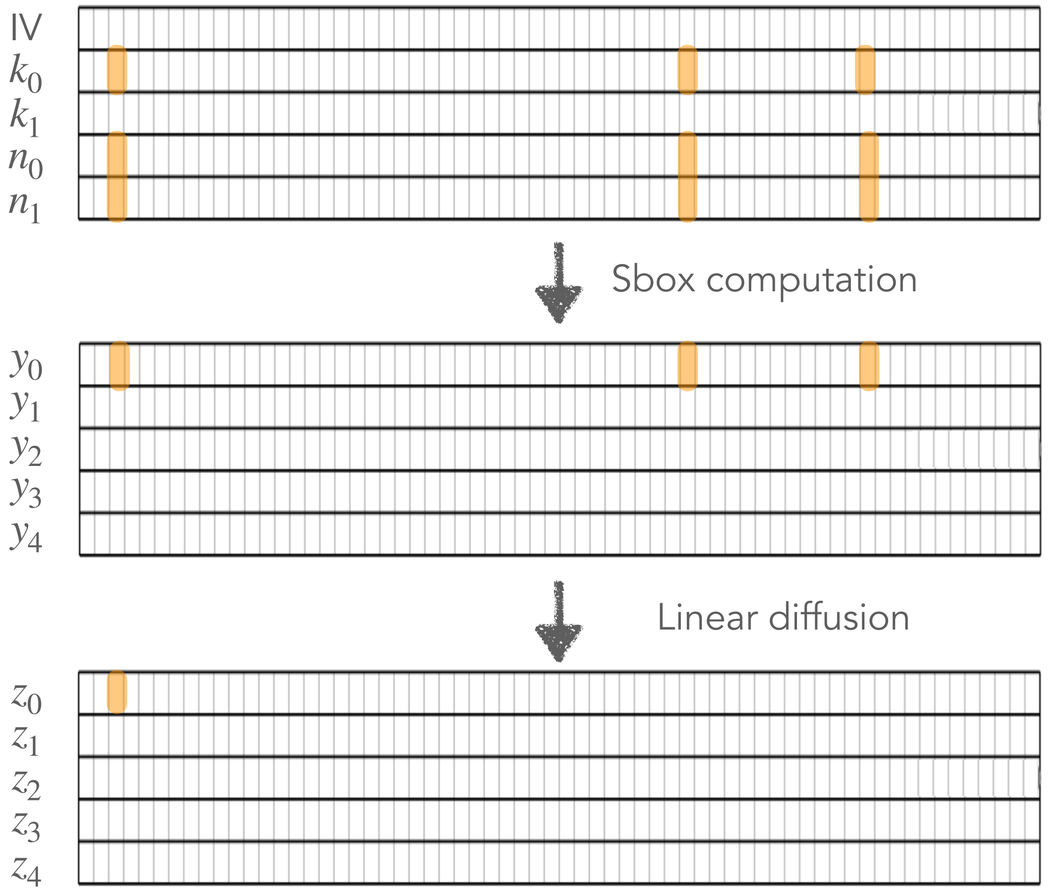
✅ Good for CPA attacks

Linear computation:

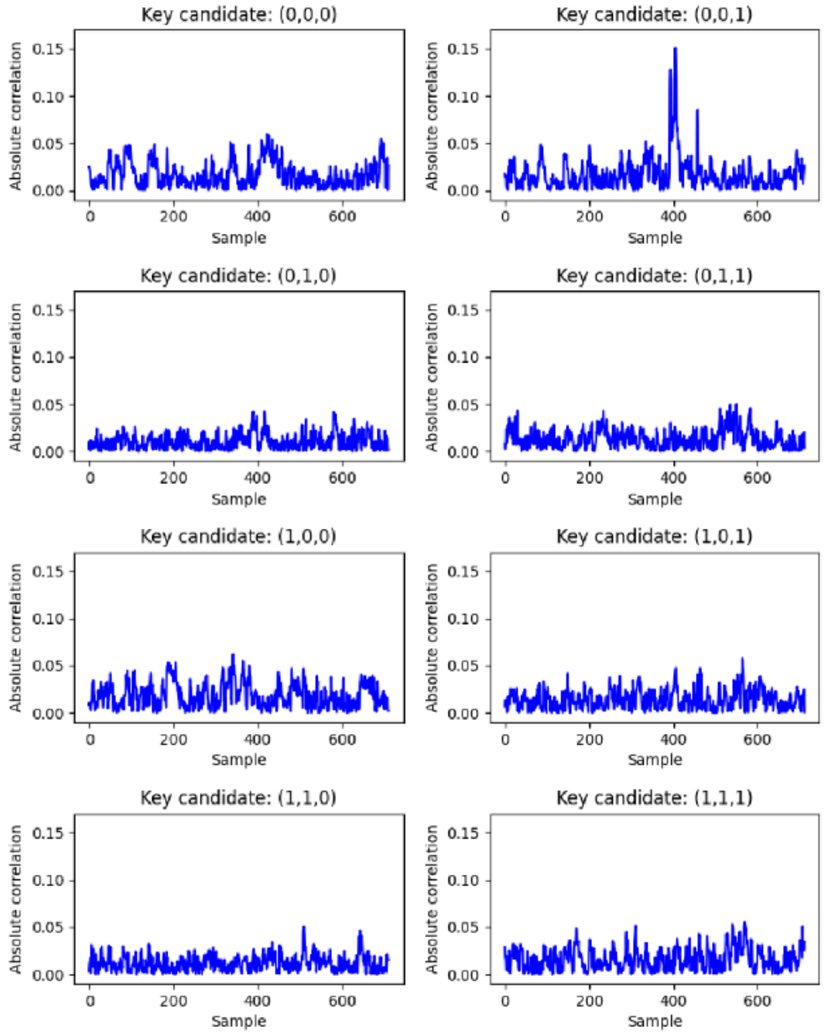
$$\tilde{z}_0^j = \tilde{y}_0^j \oplus \tilde{y}_0^{j+19} \oplus \tilde{y}_0^{j+28}$$

$$\tilde{z}_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j \oplus k_0^{j+19}(n_1^{j+19} \oplus 1) \oplus n_0^{j+19} \oplus k_0^{j+28}(n_1^{j+28} \oplus 1) \oplus n_0^{j+28}$$

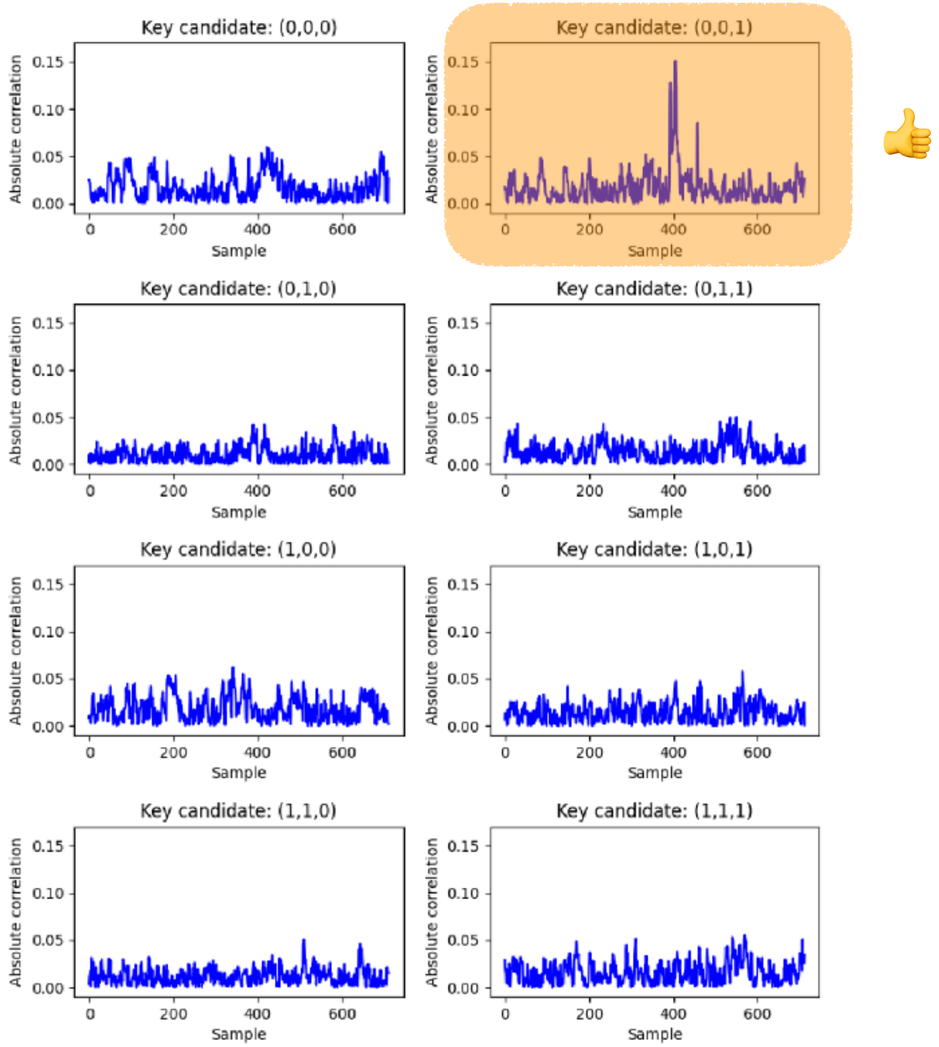
target bit



😈 Linear layer output as attack point

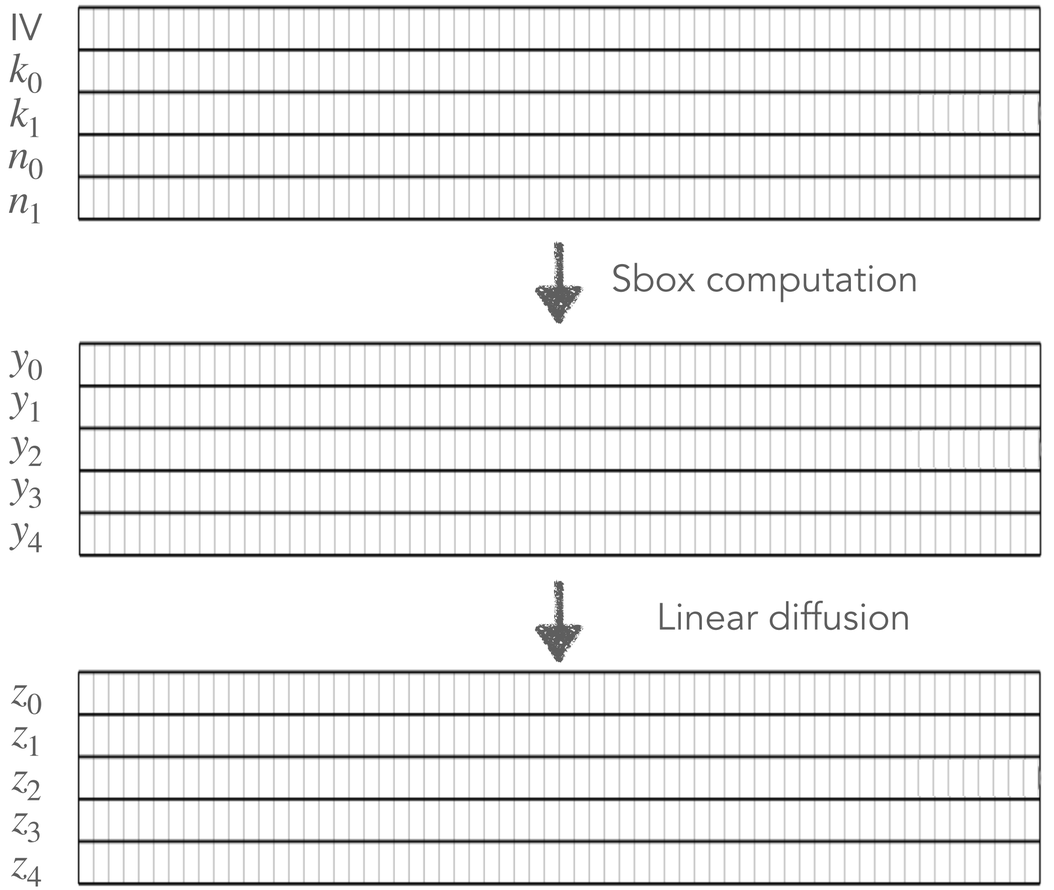


😈 Linear layer output as attack point

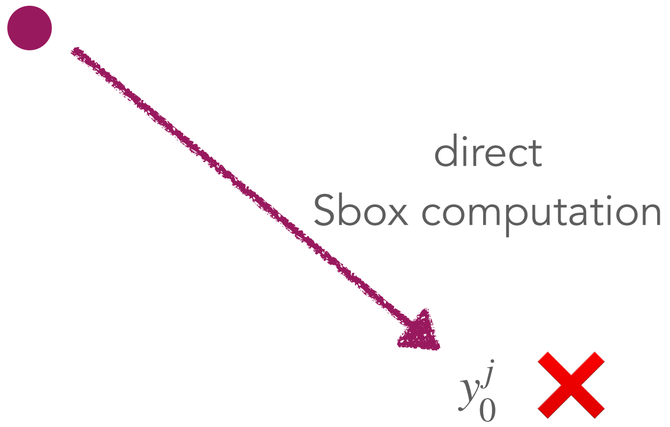


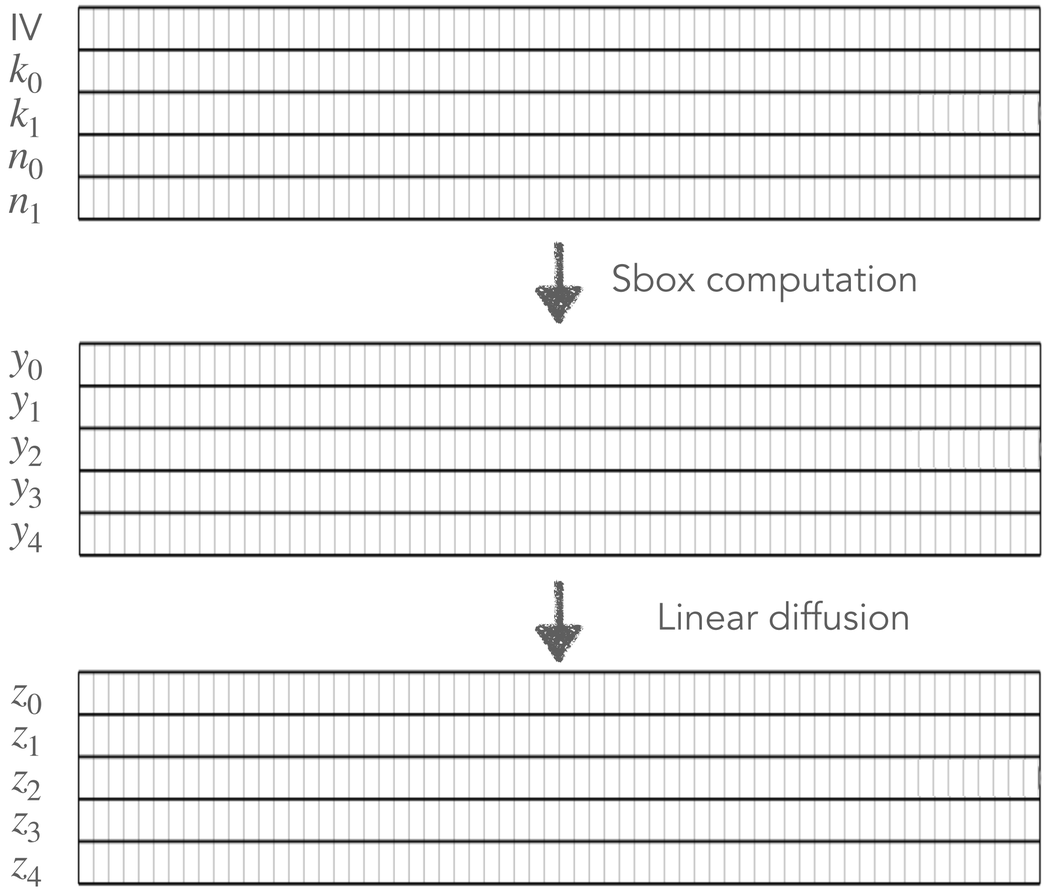
👍

Map of selection functions

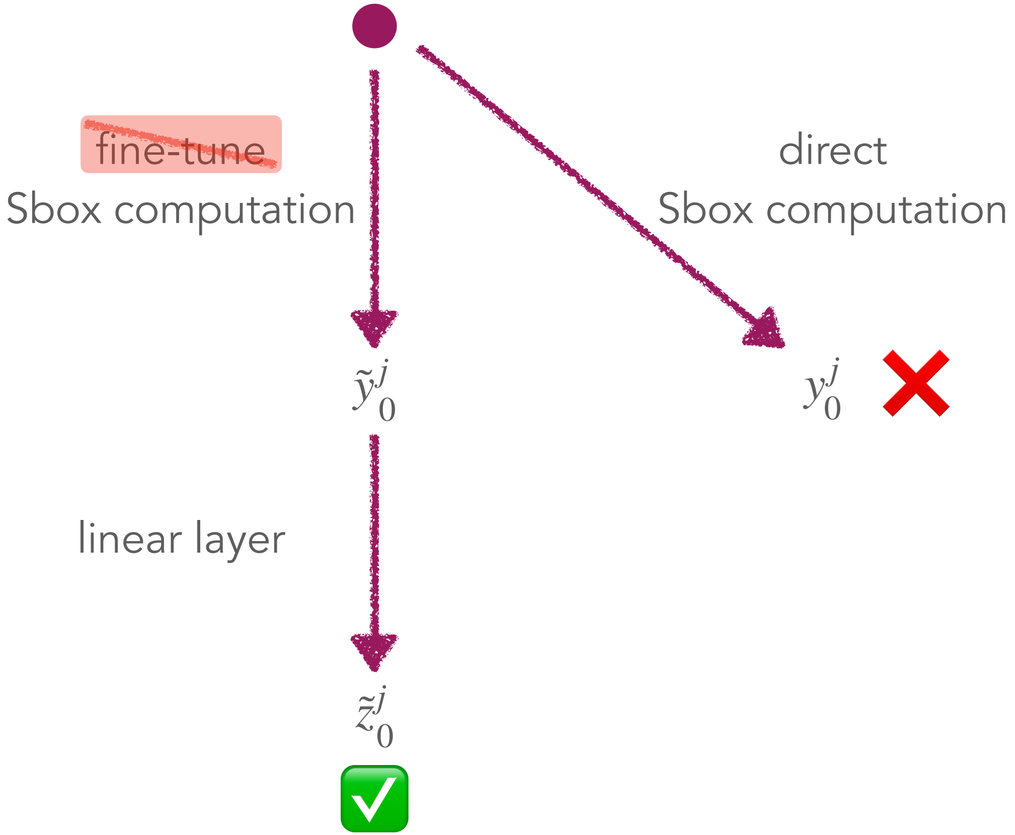


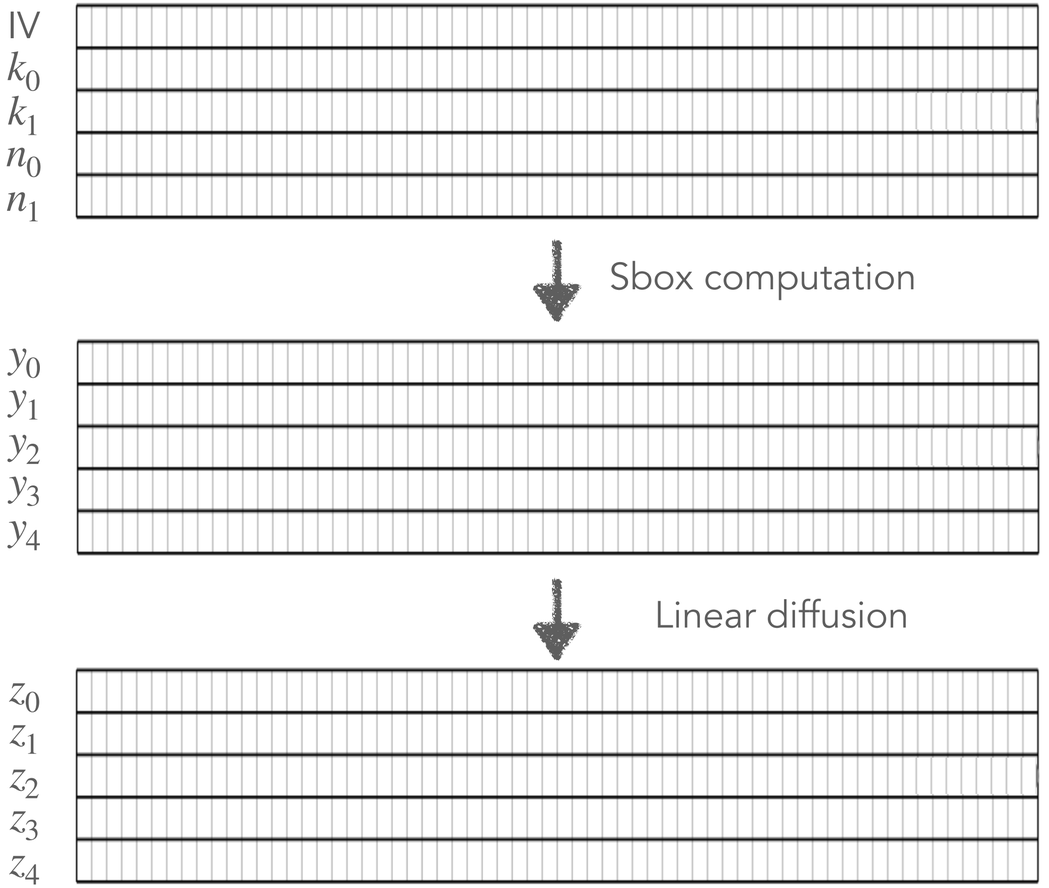
Map of selection functions



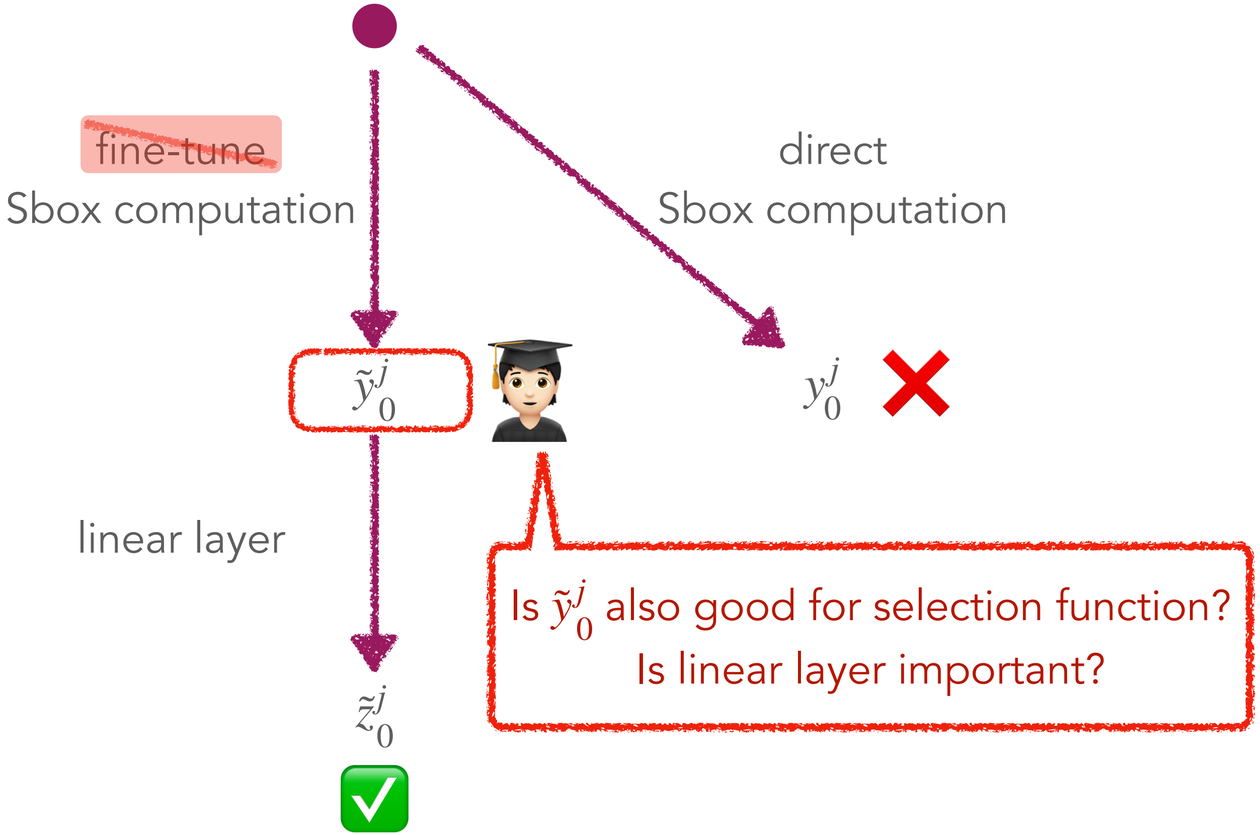


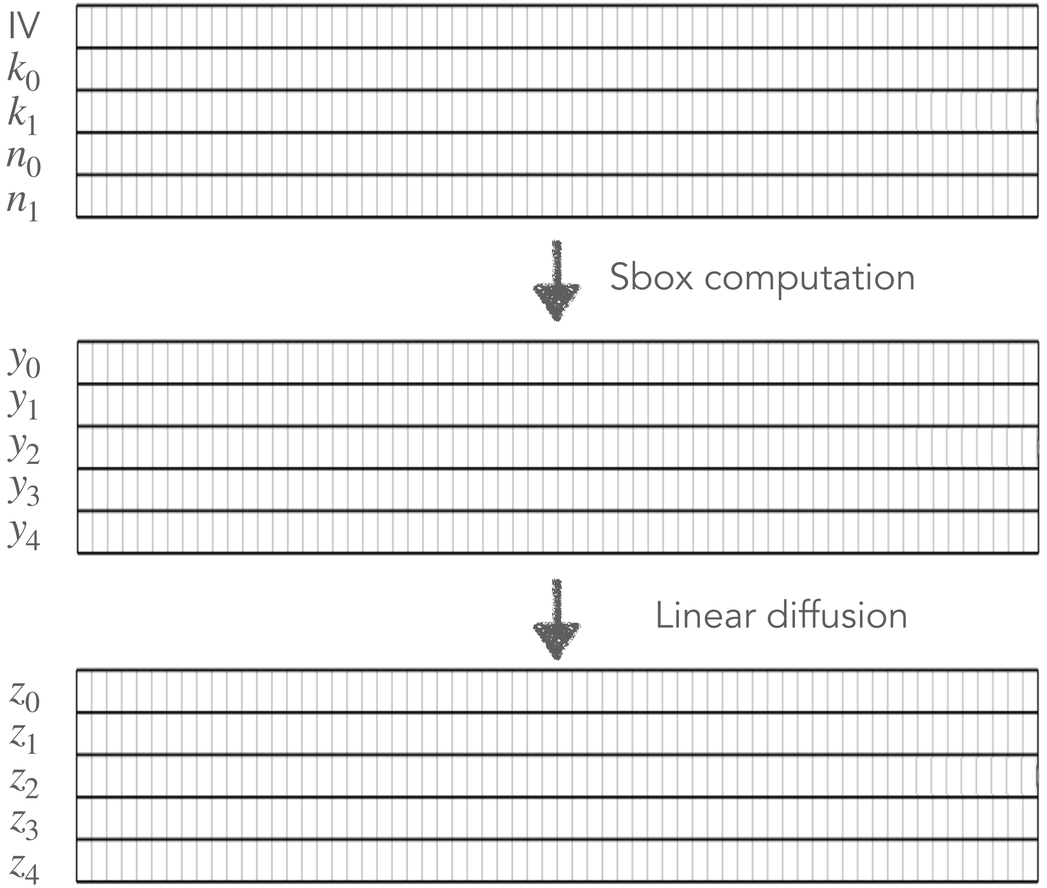
Map of selection functions





Map of selection functions



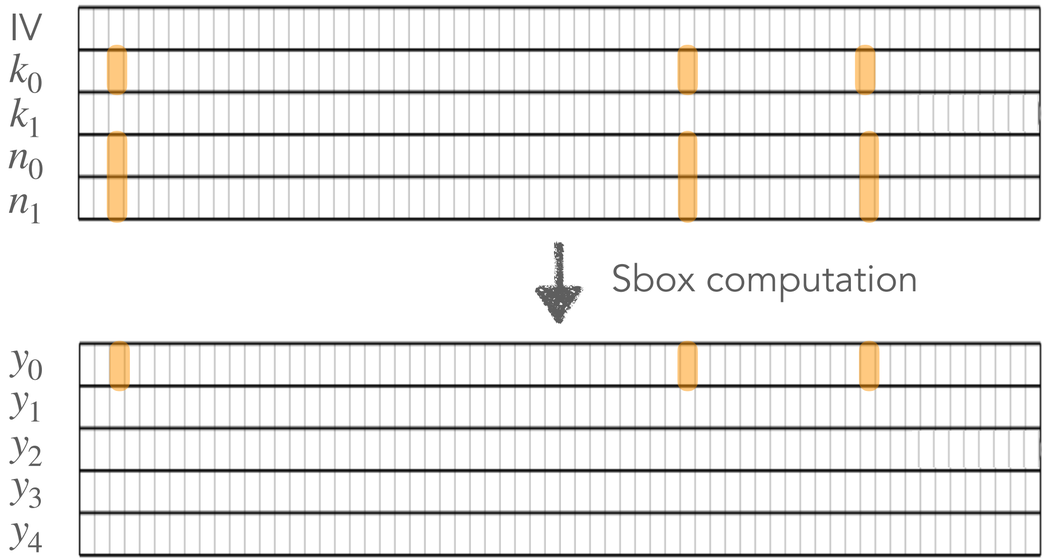


😈 $\tilde{y}_0^j$ as attack point ?

$$y_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j \oplus k_1^j k_0^j \oplus k_0^j \mathrm{IV}^j \oplus k_1^j \oplus \mathrm{IV}^j$$

[SD17]: constant part contributes a constant amount to power consumption

$$\tilde{y}_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j$$

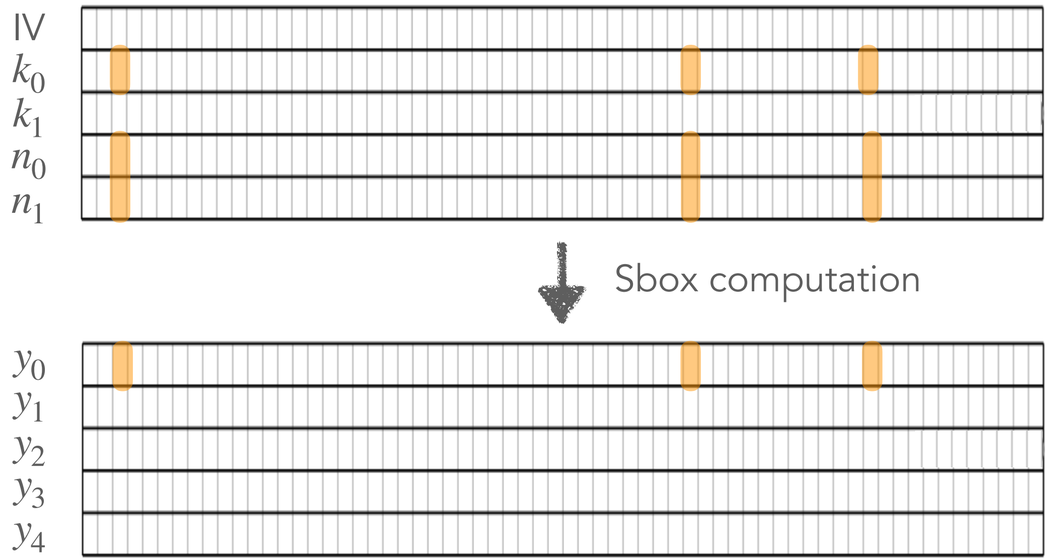


😈 $\tilde{y}_0^j$ as attack point ?

$$y_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j \oplus k_1^j k_0^j \oplus k_0^j \mathrm{IV}^j \oplus k_1^j \oplus \mathrm{IV}^j$$

[SD17]: constant part contributes a constant amount to power consumption

$$\tilde{y}_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j$$



Evaluate the equation:

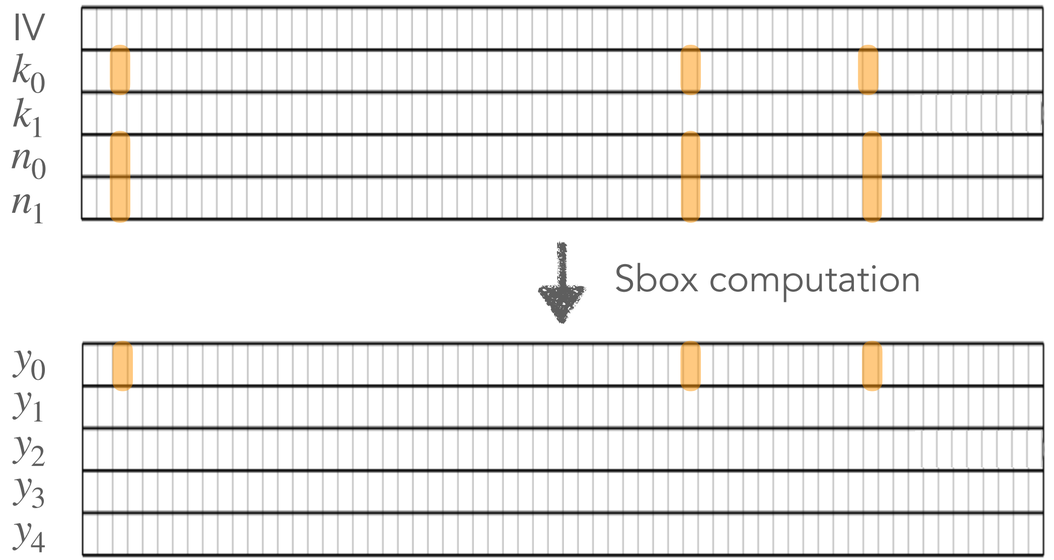
$$\tilde{y}_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j = \begin{cases} n_0^j & \text{if } k_0^j = 0, \\ n_0^j \oplus n_1^j \oplus 1 & \text{if } k_0^j = 1. \end{cases}$$

😈 $\tilde{y}_0^j$ as attack point ?

$$y_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j \oplus k_1^j k_0^j \oplus k_0^j \mathrm{IV}^j \oplus k_1^j \oplus \mathrm{IV}^j$$

[SD17]: constant part contributes a constant amount to power consumption

$$\tilde{y}_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j$$



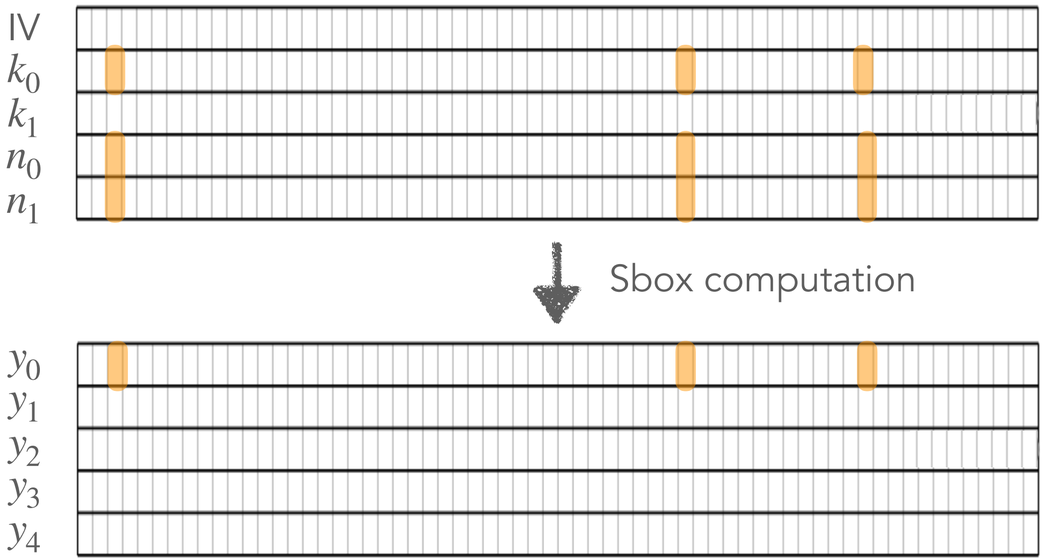
Evaluate the equation:

$$\tilde{y}_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j = \begin{cases} n_0^j & \text{if } k_0^j = 0, \\ n_0^j \oplus n_1^j \oplus 1 & \text{if } k_0^j = 1. \end{cases}$$

→ Correlated with activity of n_0

→ Correlated with activity of $n_0 \oplus n_1$

😈 $\tilde{y}_0^j$ as attack point ?



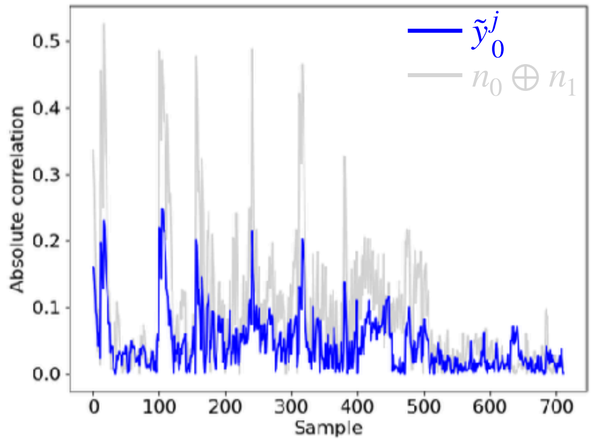
Evaluate the equation:

$$\tilde{y}_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j = \begin{cases} n_0^j & \text{if } k_0^j = 0, \\ n_0^j \oplus n_1^j \oplus 1 & \text{if } k_0^j = 1. \end{cases}$$

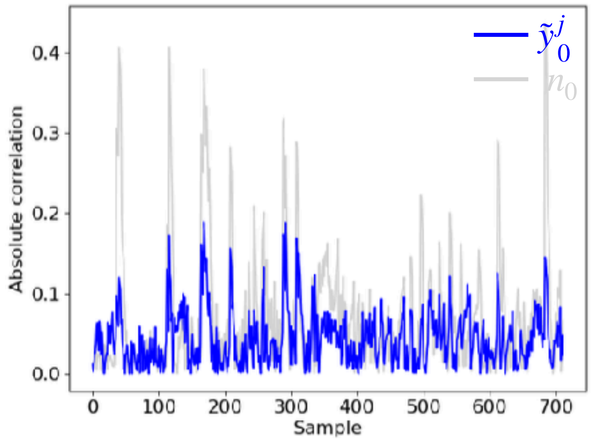
→ Correlated with activity of n_0

→ Correlated with activity of $n_0 \oplus n_1$

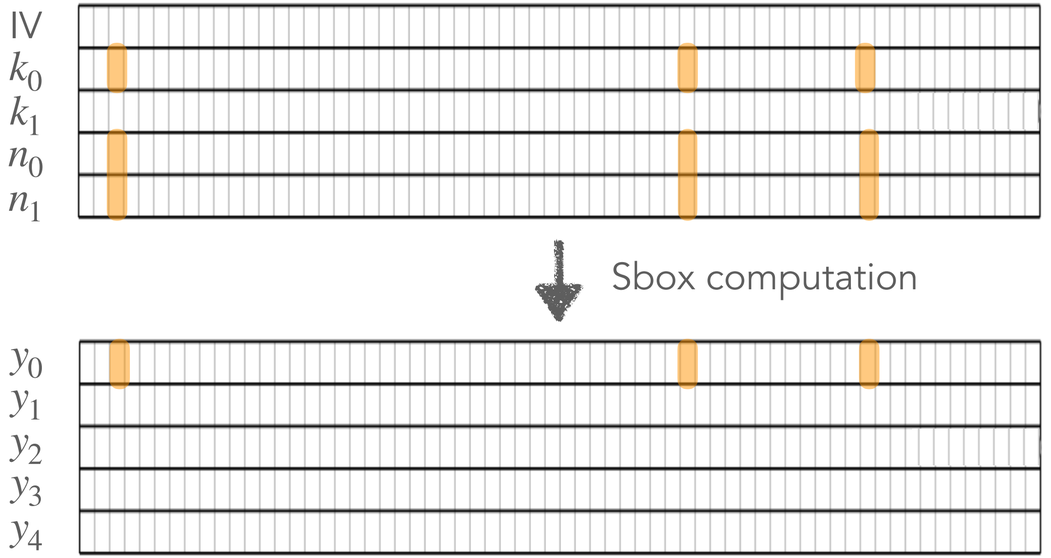
😈 $\tilde{y}_0^j$ as attack point ?



(a) $k_0^j = 0$



(b) $k_0^j = 1$



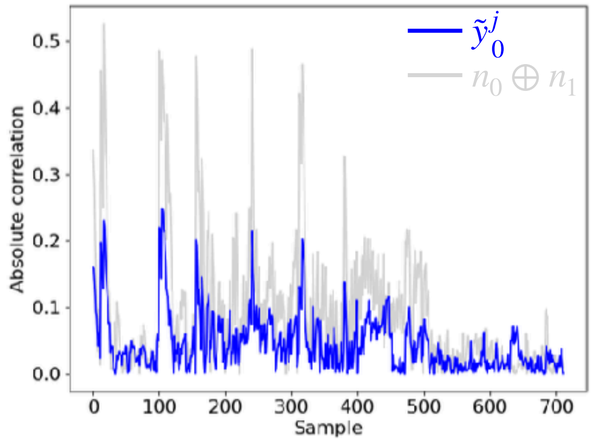
Evaluate the equation:

$$\tilde{y}_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j = \begin{cases} n_0^j & \text{if } k_0^j = 0, \\ n_0^j \oplus n_1^j \oplus 1 & \text{if } k_0^j = 1. \end{cases}$$

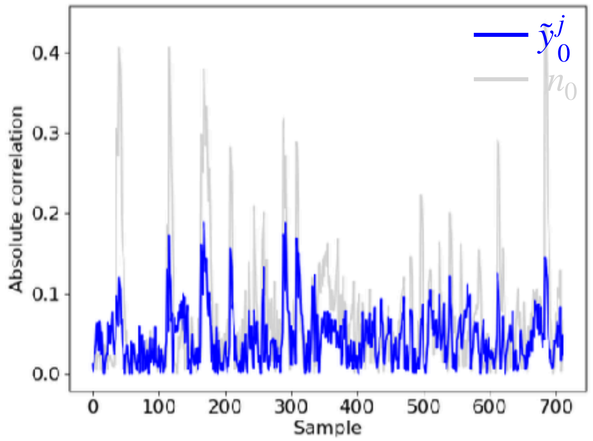
→ Correlated with activity of n_0

→ Correlated with activity of $n_0 \oplus n_1$

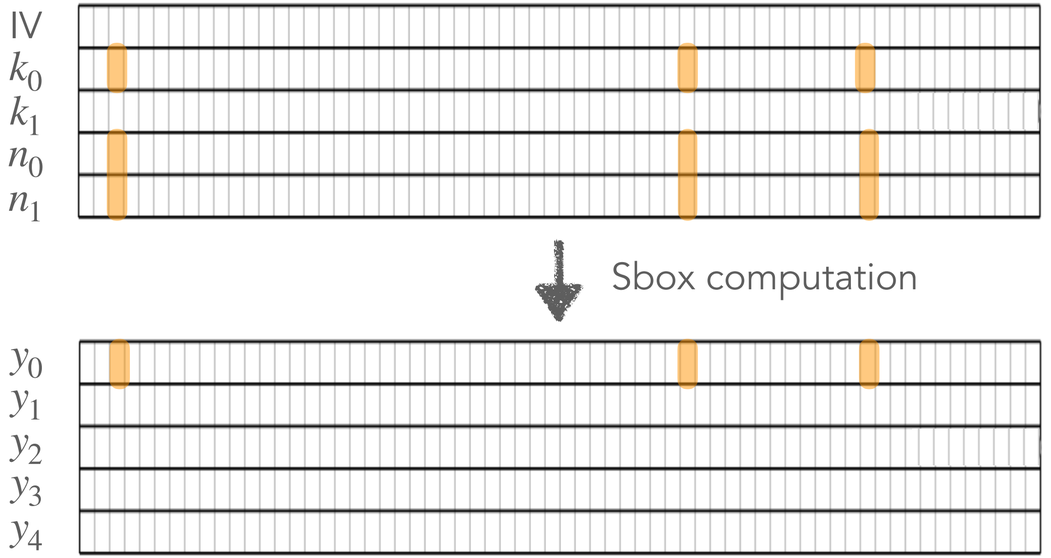
😈 $\tilde{y}_0^j$ as attack point ?



(a) $k_0^j = 0$



(b) $k_0^j = 1$



Evaluate the equation:

$$\tilde{y}_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j = \begin{cases} n_0^j & \text{if } k_0^j = 0, \\ n_0^j \oplus n_1^j \oplus 1 & \text{if } k_0^j = 1. \end{cases}$$

→ Correlated with activity of n_0

→ Correlated with activity of $n_0 \oplus n_1$

❌ Not effective for CPA attacks

Correlation Power Analysis (CPA) attacks on Ascon-AEAD

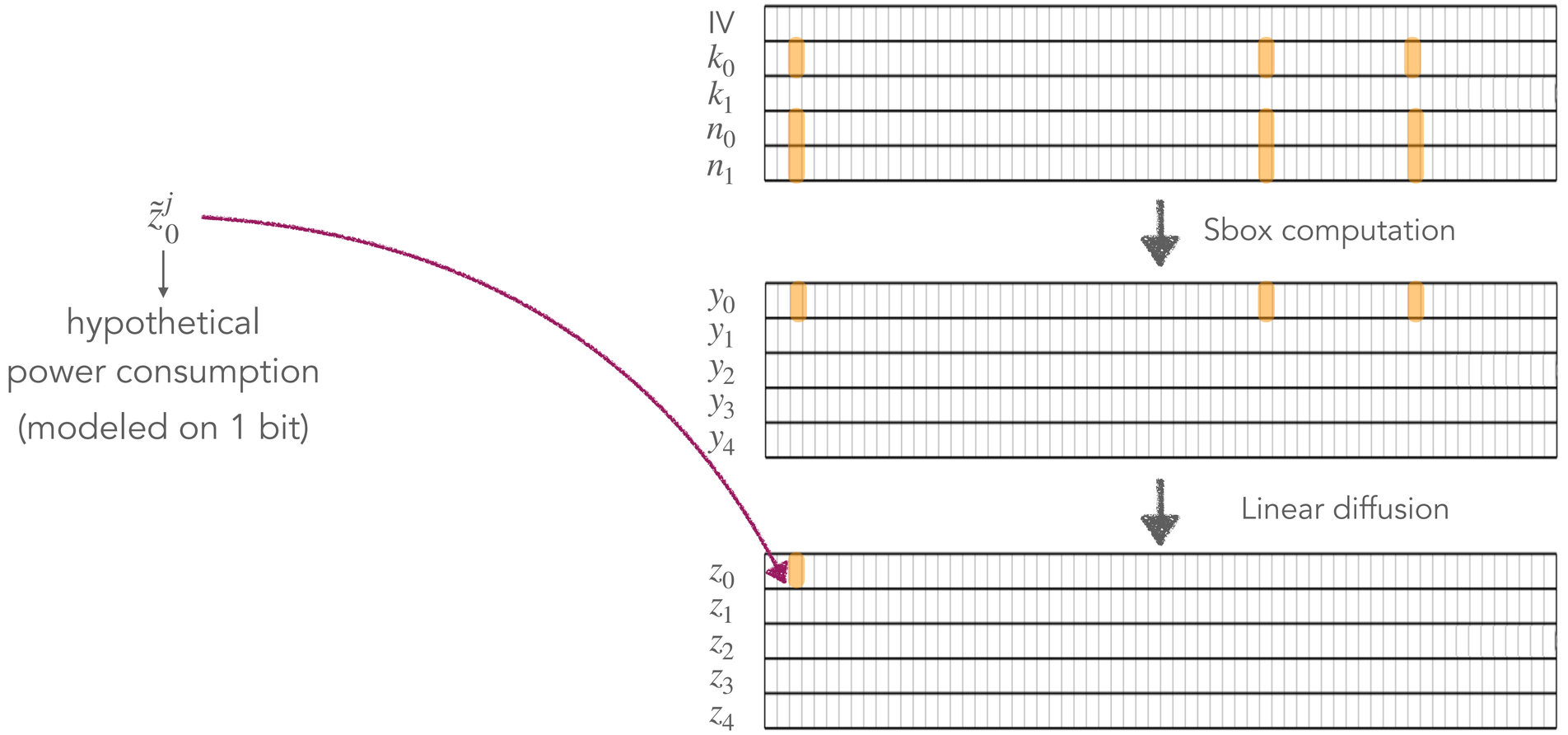
👨‍🎓 Contribution 1: in-depth analysis of selection functions

Nguyen, Grosso, Cayrel - CASCADE 2025

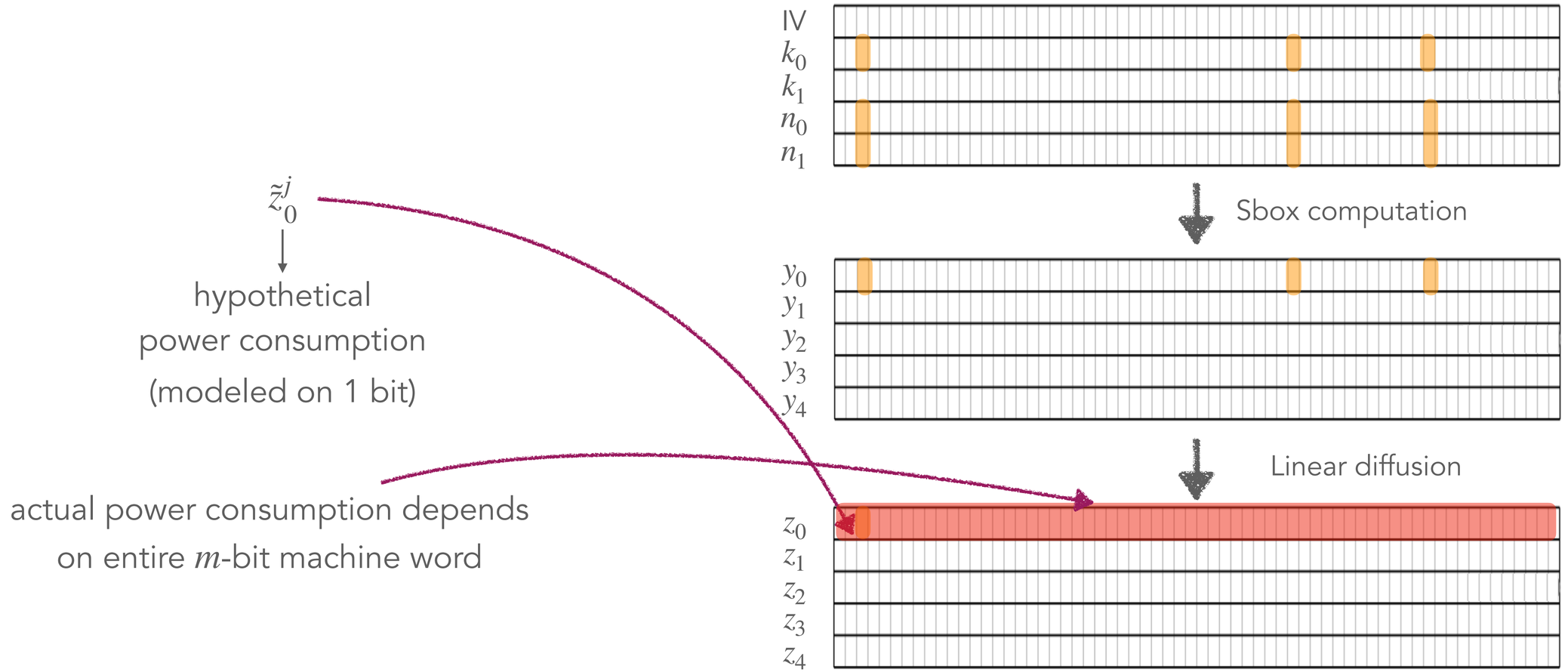
👨‍🎓 Contribution 2: extension of selection function to multiple bits

Nguyen, Grosso, Cayrel - SECRYPT 2025

Observation



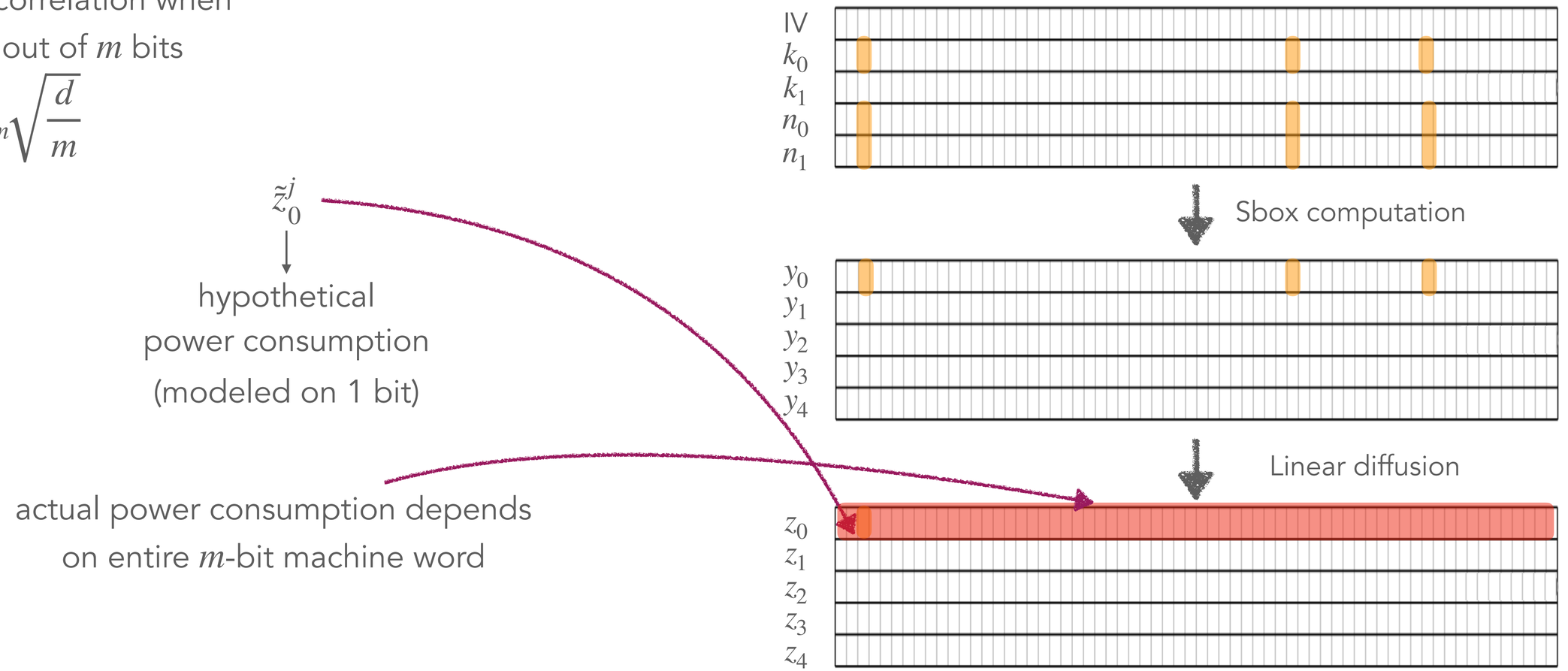
Observation



Observation

[BCO04]: partial correlation when considering d out of m bits

$$\rho_d = \rho_m \sqrt{\frac{d}{m}}$$



Observation

[BCO04]: partial correlation when considering d out of m bits

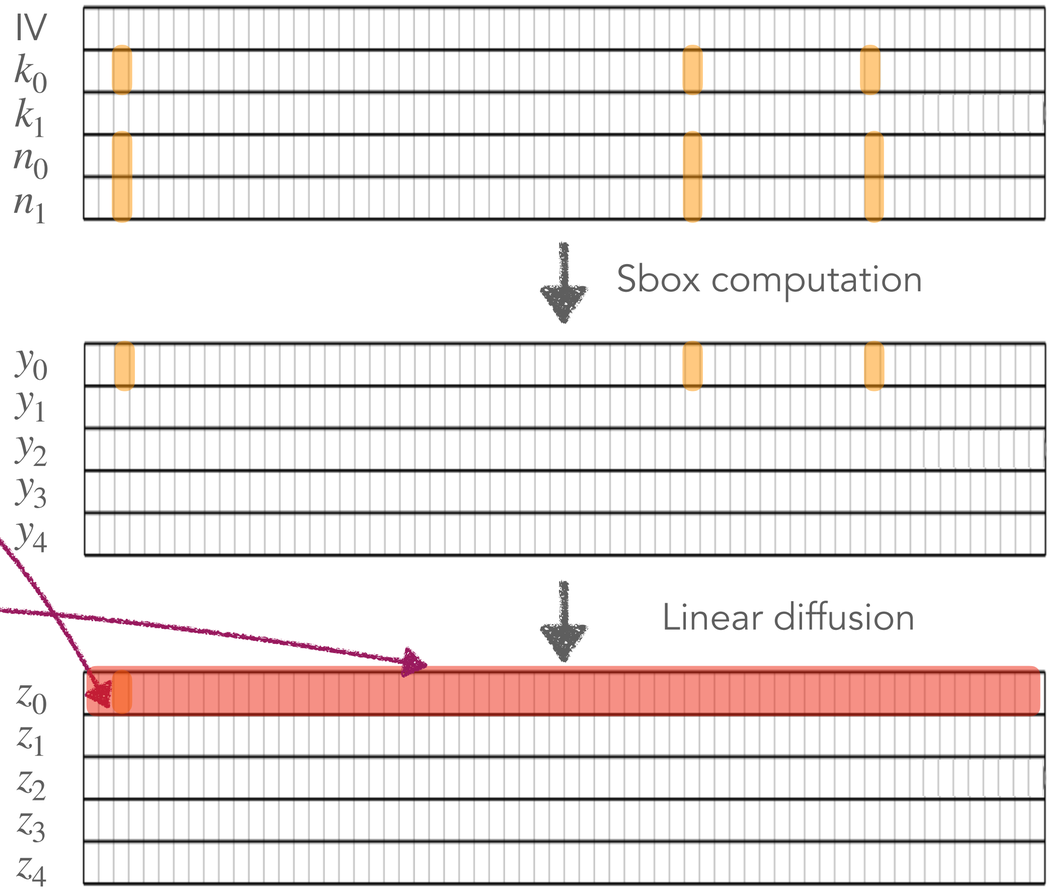
$$\rho_d = \rho_m \sqrt{\frac{d}{m}}$$

$\tilde{z}_0^j$

hypothetical power consumption (modeled on 1 bit)

$d = 1$

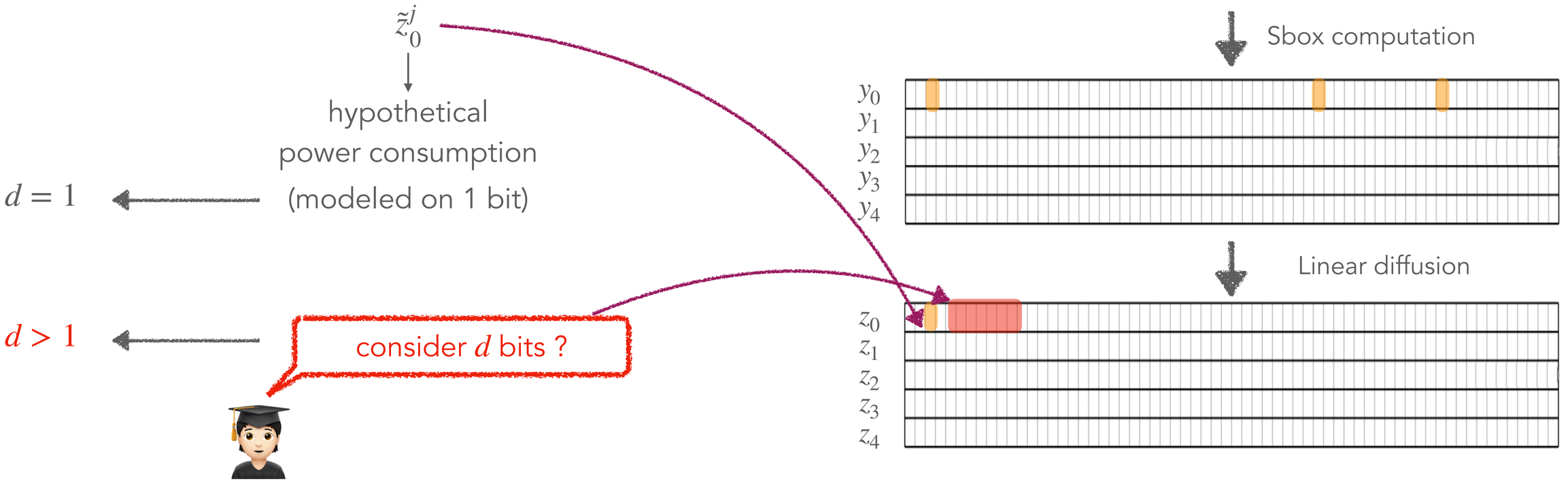
actual power consumption depends on entire m-bit machine word



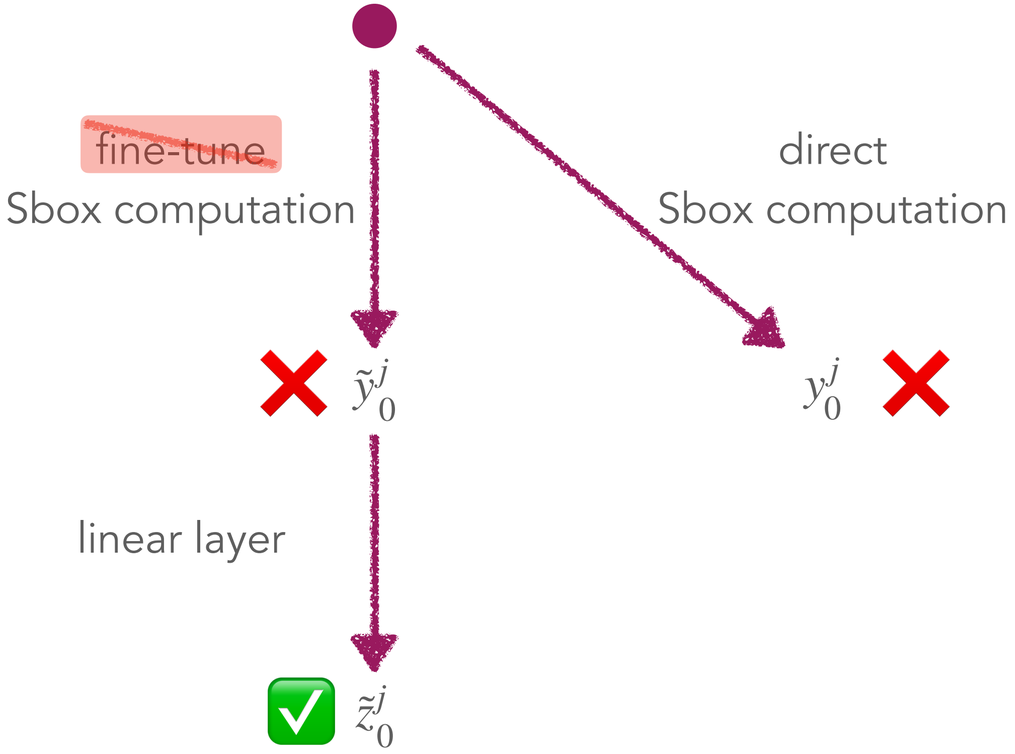
Observation

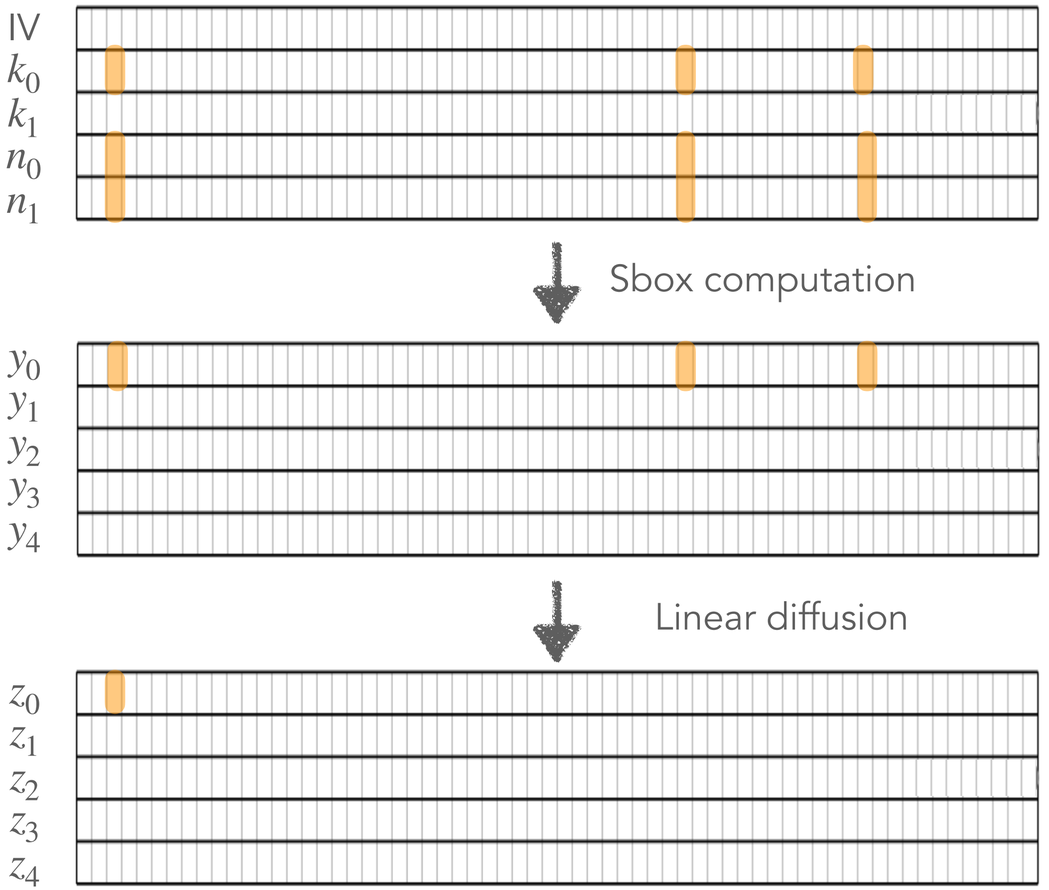
[BCO04]: partial correlation when considering d out of m bits

$$\rho_d = \rho_m \sqrt{\frac{d}{m}}$$

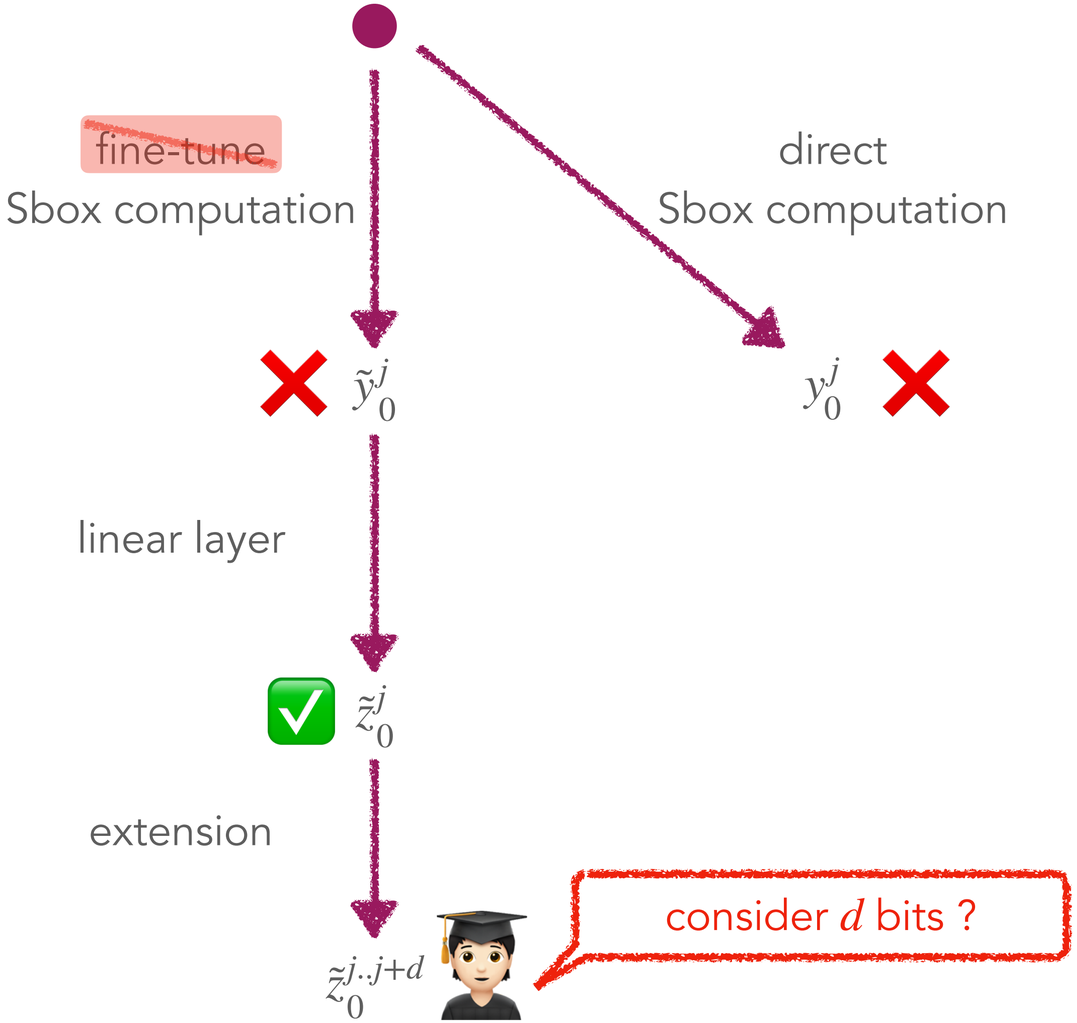


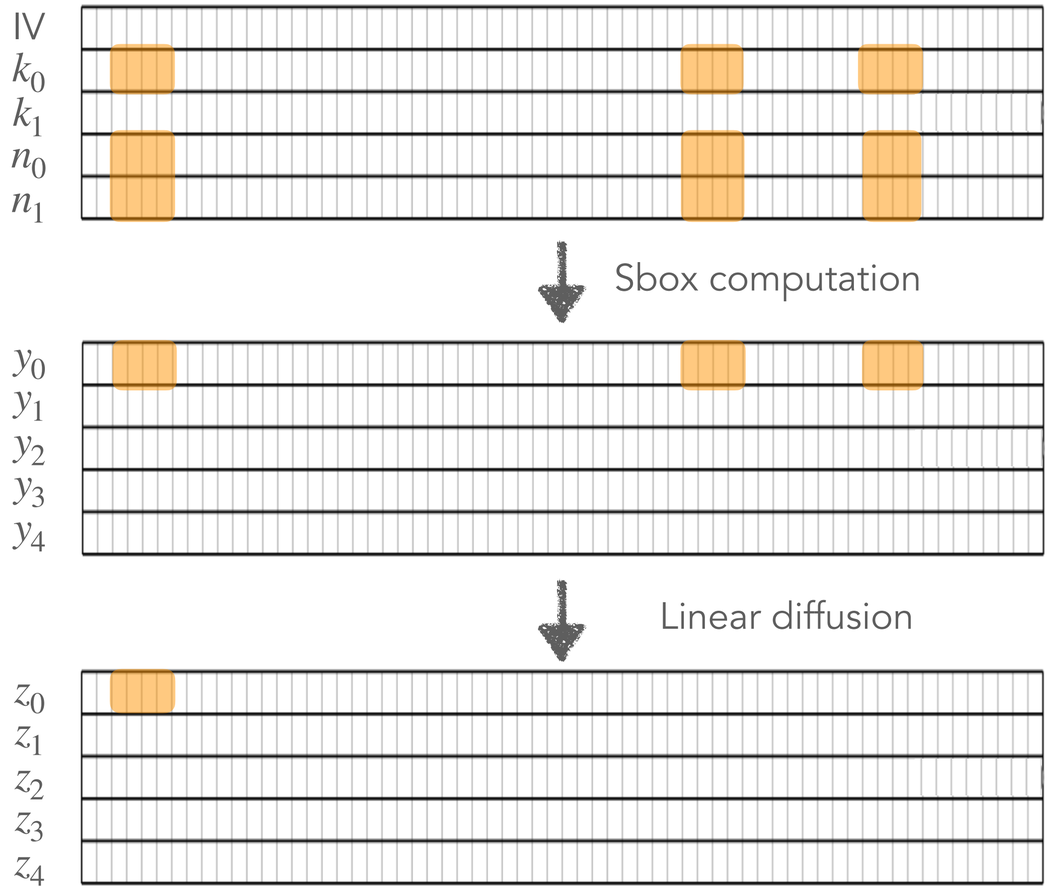
Map of selection functions



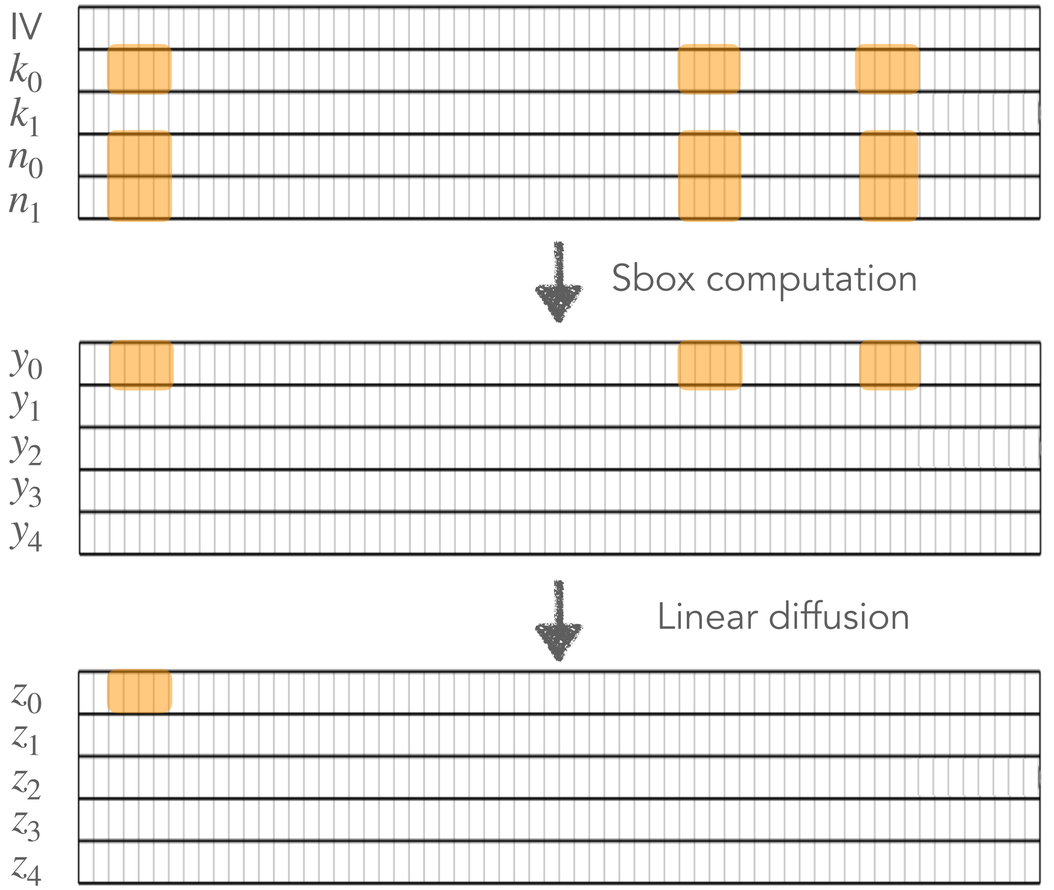


Map of selection functions





Is that simple?



Linear computation:

$$\tilde{z}_0^j = \tilde{y}_0^j \oplus \tilde{y}_0^{j+19} \oplus \tilde{y}_0^{j+28}$$

$$\tilde{z}_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j \oplus k_0^{j+19}(n_1^{j+19} \oplus 1) \oplus n_0^{j+19} \oplus k_0^{j+28}(n_1^{j+28} \oplus 1) \oplus n_0^{j+28}$$

Is that simple?

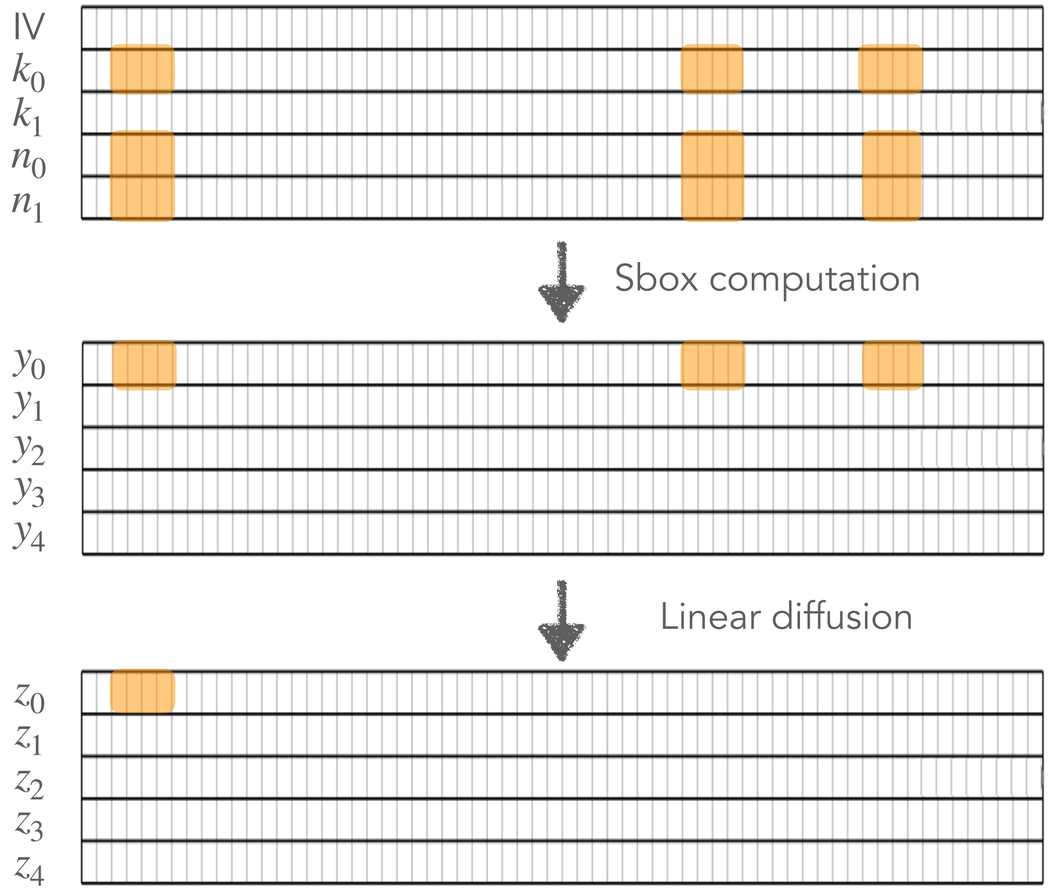
So just extend everything to d bits ? 🤔

Linear computation:

$$\tilde{z}_0^j = \tilde{y}_0^j \oplus \tilde{y}_0^{j+19} \oplus \tilde{y}_0^{j+28}$$

$$\tilde{z}_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j \oplus k_0^{j+19}(n_1^{j+19} \oplus 1) \oplus n_0^{j+19} \oplus k_0^{j+28}(n_1^{j+28} \oplus 1) \oplus n_0^{j+28}$$

target bit



Is that simple?

So just extend everything to d bits ? 🤔

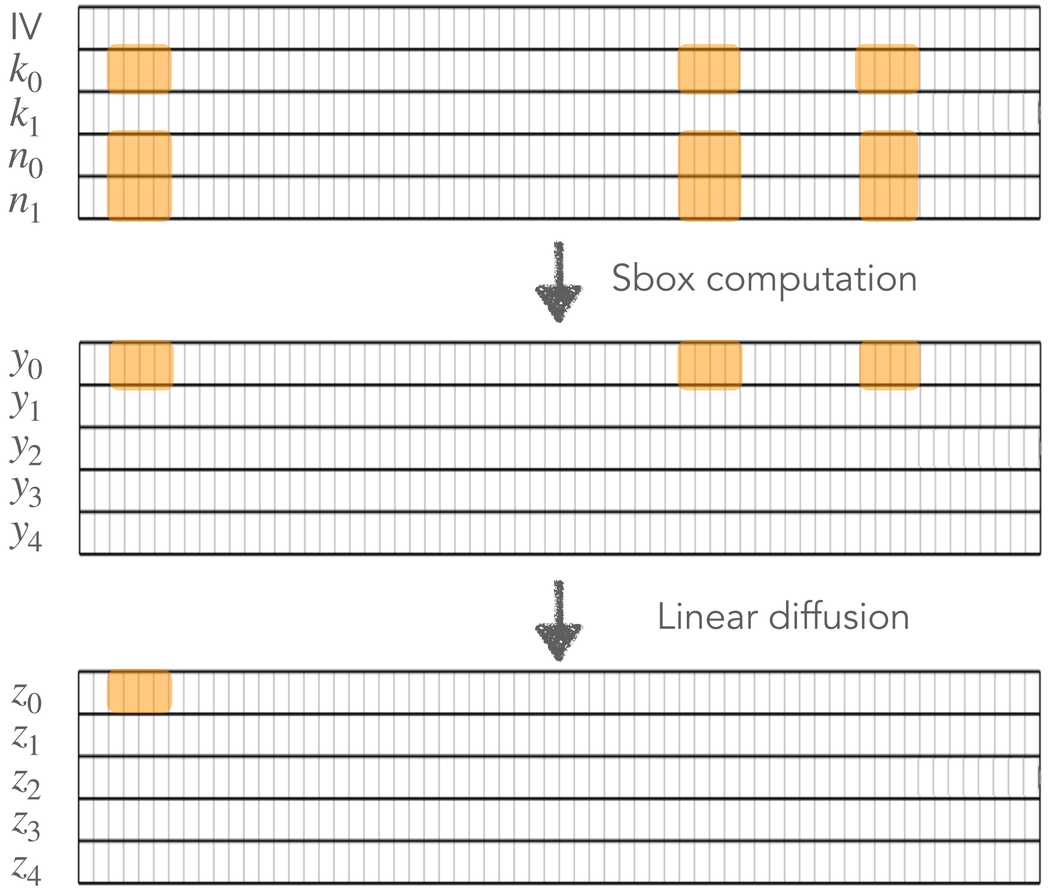
It's a bit more complicated than that 😈

Linear computation:

$$\tilde{z}_0^j = \tilde{y}_0^j \oplus \tilde{y}_0^{j+19} \oplus \tilde{y}_0^{j+28}$$

$$\tilde{z}_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j \oplus k_0^{j+19}(n_1^{j+19} \oplus 1) \oplus n_0^{j+19} \oplus k_0^{j+28}(n_1^{j+28} \oplus 1) \oplus n_0^{j+28}$$

target bit →

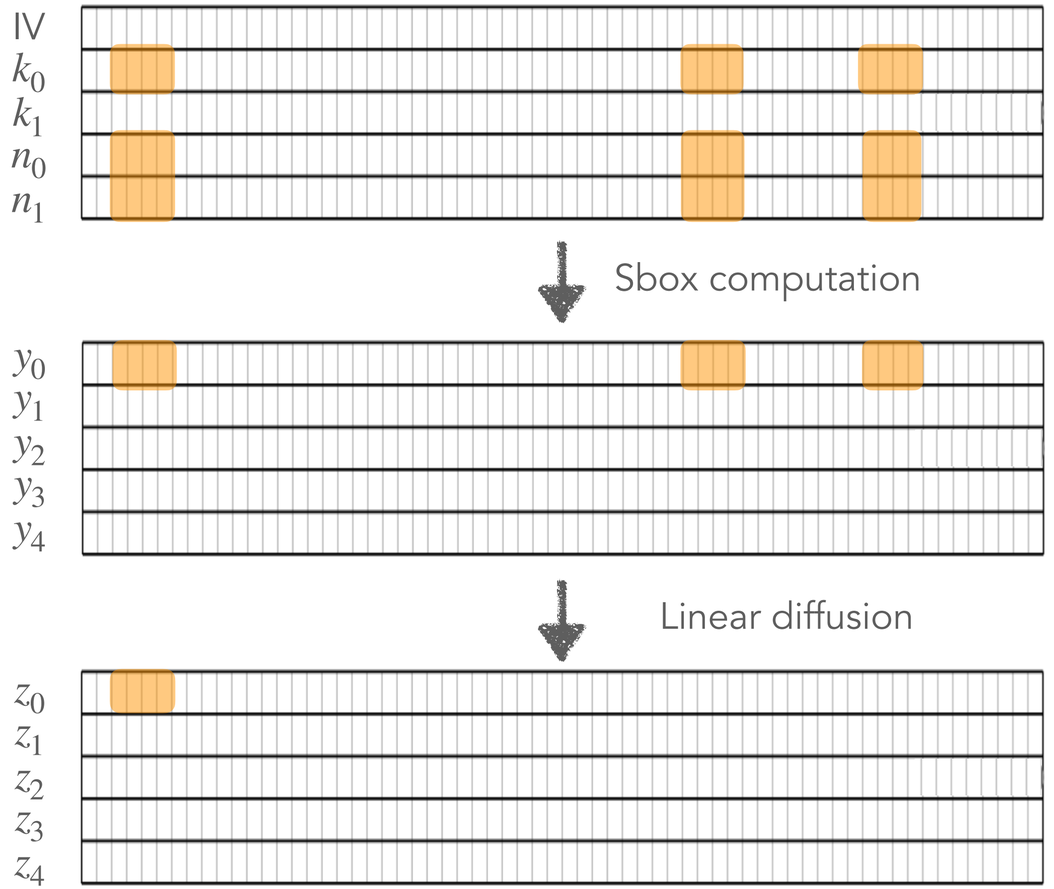


😈 Extension to multiple bits

$$y_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j \oplus k_1^j k_0^j \oplus k_0^j \mathrm{IV}^j \oplus k_1^j \oplus \mathrm{IV}^j$$

what we do before:

$$\tilde{y}_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j$$



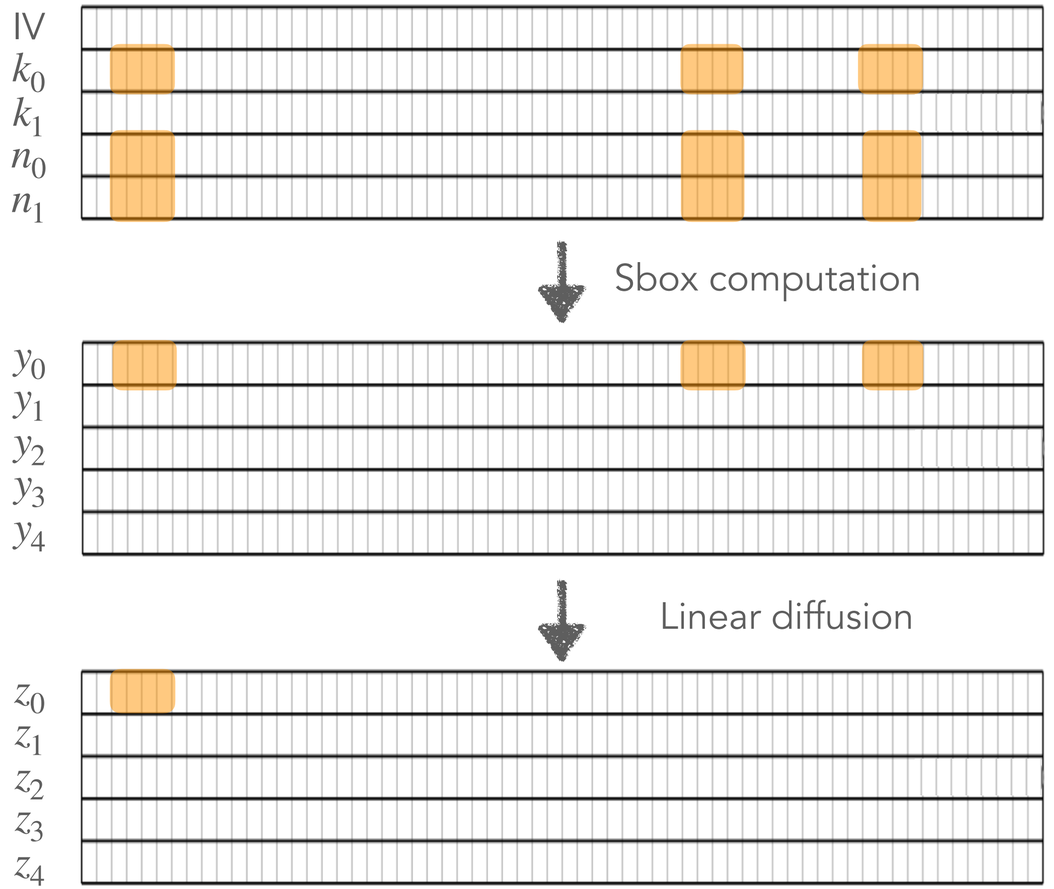
😈 Extension to multiple bits

$$y_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j \oplus k_1^j k_0^j \oplus k_0^j \mathsf{IV}^j \oplus k_1^j \oplus \mathsf{IV}^j$$

what we do before:

$$\tilde{y}_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j$$

$$y_0^j = \tilde{y}_0^j \oplus \delta_0^j$$



😈 Extension to multiple bits

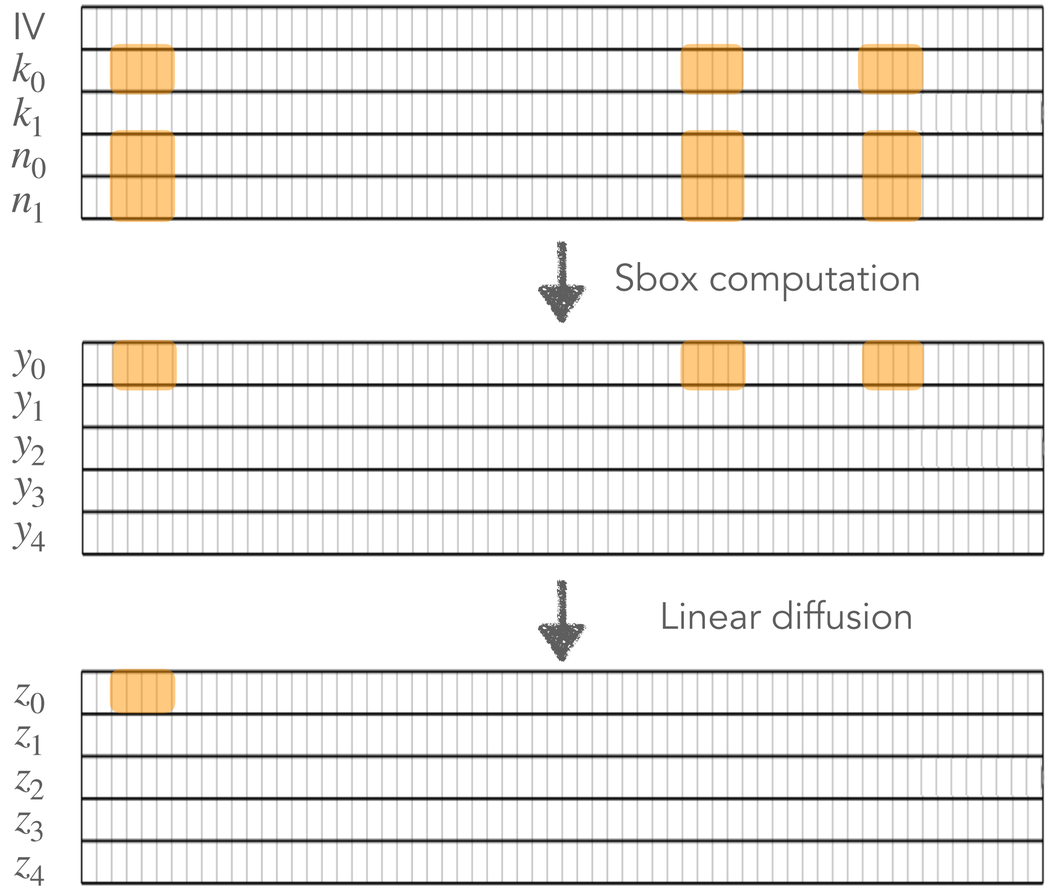
$$y_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j \oplus k_1^j k_0^j \oplus k_0^j \mathrm{IV}^j \oplus k_1^j \oplus \mathrm{IV}^j$$

what we do before:

$$\tilde{y}_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j$$

$$y_0^j = \tilde{y}_0^j \oplus \delta_0^j$$

$\delta_0^j = 0$: ok



😈 Extension to multiple bits

$$y_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j \oplus k_1^j k_0^j \oplus k_0^j \mathrm{IV}^j \oplus k_1^j \oplus \mathrm{IV}^j$$

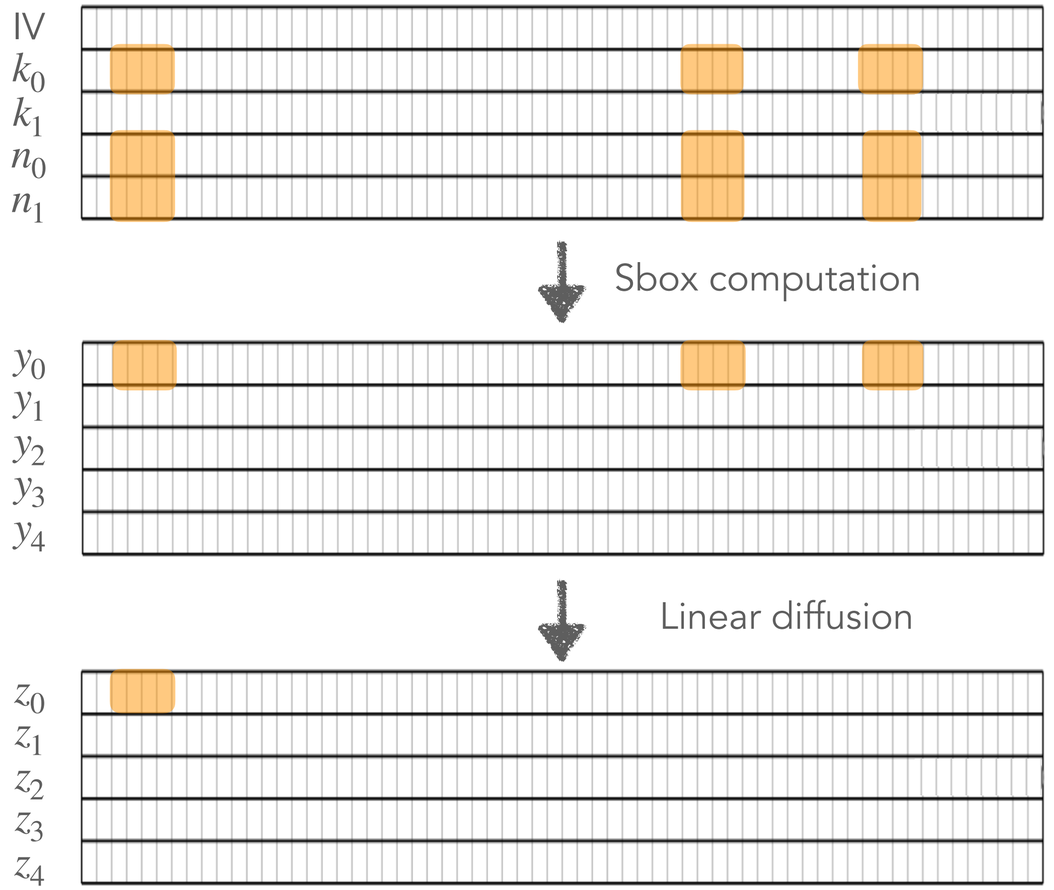
what we do before:

$$\tilde{y}_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j$$

$$y_0^j = \tilde{y}_0^j \oplus \delta_0^j$$

$\delta_0^j = 0$: ok

$\delta_0^j = 1$: distribution negated



😈 Extension to multiple bits

$$y_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j \oplus k_1^j k_0^j \oplus k_0^j \mathrm{IV}^j \oplus k_1^j \oplus \mathrm{IV}^j$$

what we do before:

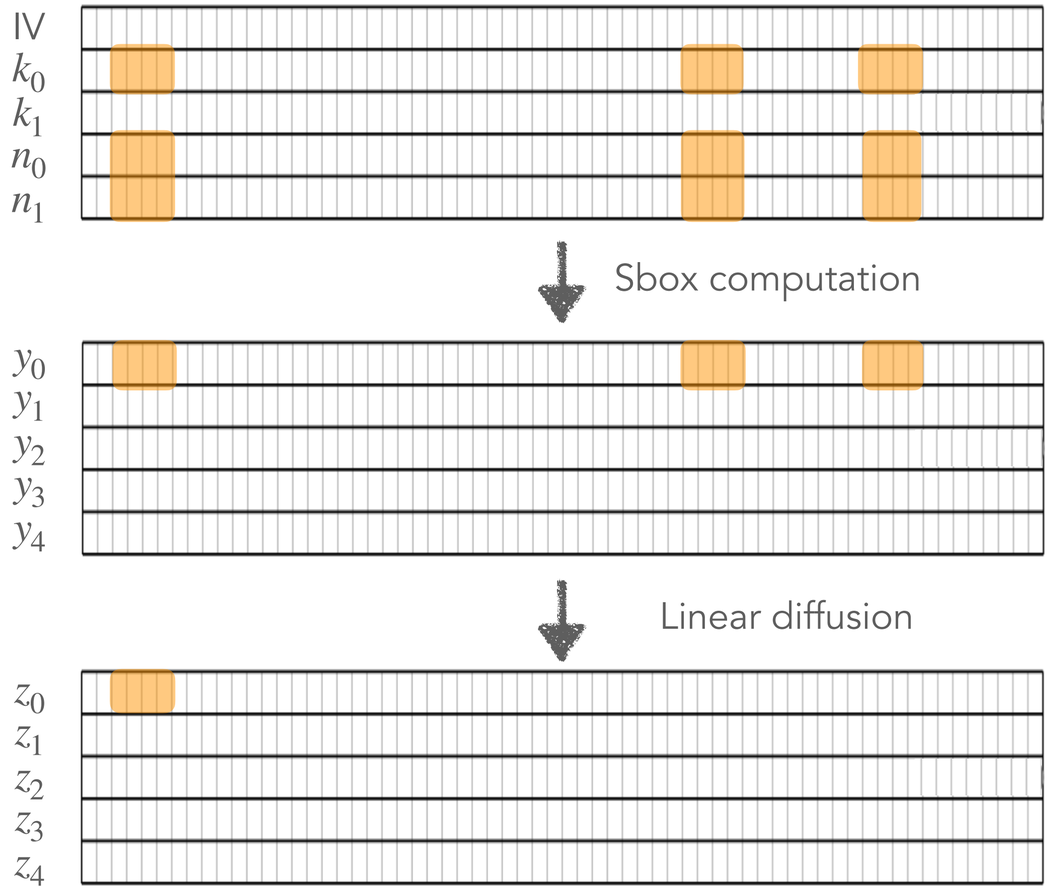
$$\tilde{y}_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j$$

$$y_0^j = \tilde{y}_0^j \oplus \delta_0^j$$

$\delta_0^j = 0$: ok

$\delta_0^j = 1$: distribution negated

but <u>absolute</u> correlation coefficient doesn't change → still ok



😈 Extension to multiple bits

$$y_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j \oplus k_1^j k_0^j \oplus k_0^j \mathrm{IV}^j \oplus k_1^j \oplus \mathrm{IV}^j$$

what we do before:

$$\tilde{y}_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j$$

$$y_0^j = \tilde{y}_0^j \oplus \delta_0^j$$

$\delta_0^j = 0$: ok

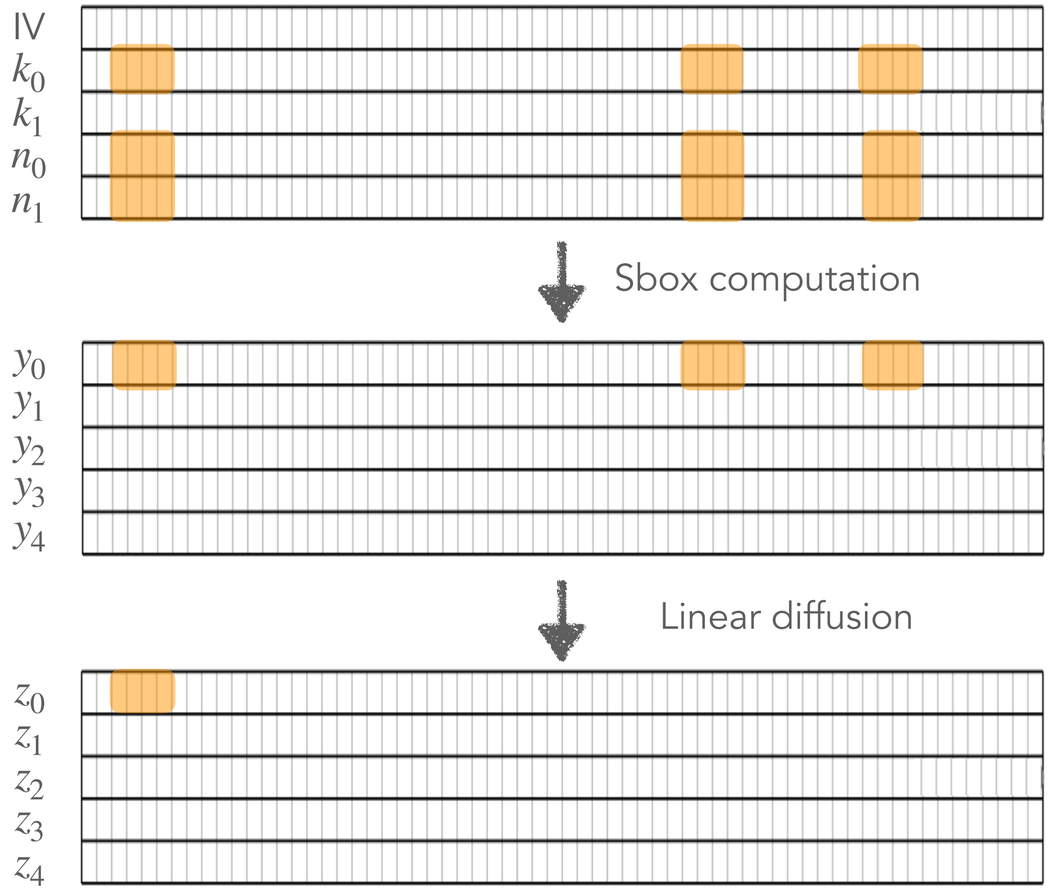
$\delta_0^j = 1$: distribution negated

but <u>absolute</u> correlation coefficient doesn't change → still ok

If we extend:

$$y_0^{j..j+d} = \tilde{y}_0^{j..j+d} \oplus \delta_0^{j..j+d}$$

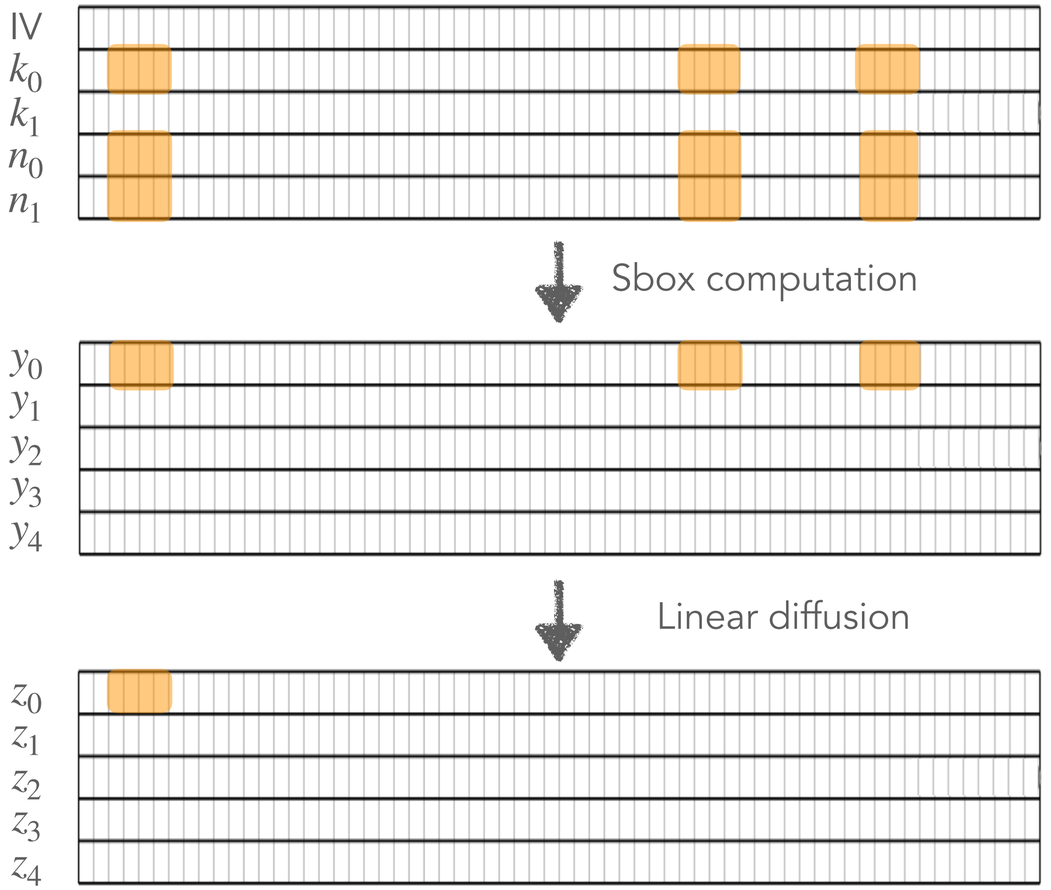
distribution of $y_0^{j..j+d}$ isn't simply negated



😈 Extension to multiple bits

$$y_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j \oplus k_1^j k_0^j \oplus k_0^j \mathrm{IV}^j \oplus k_1^j \oplus \mathrm{IV}^j$$

$$y_0^j = \tilde{y}_0^j \oplus \delta_0^j$$

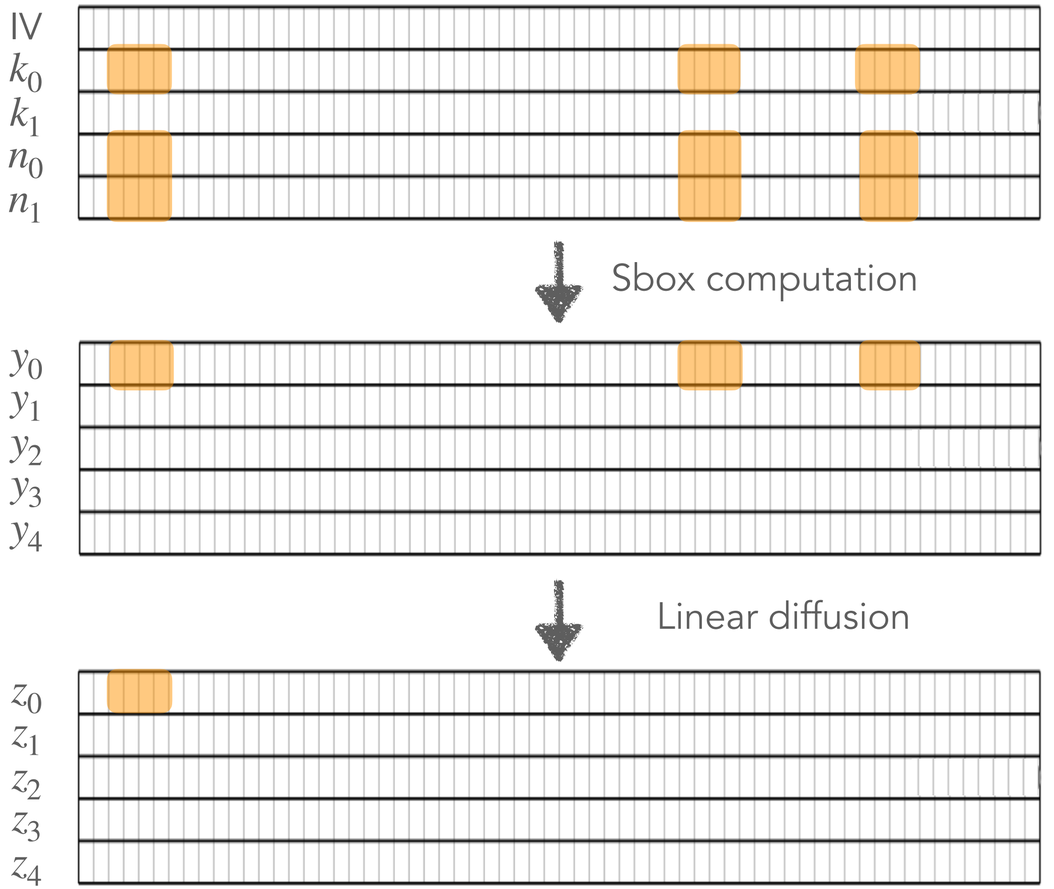


😈 Extension to multiple bits

$$y_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j \oplus k_1^j k_0^j \oplus k_0^j \mathrm{IV}^j \oplus k_1^j \oplus \mathrm{IV}^j$$

$$y_0^j = \tilde{y}_0^j \oplus \delta_0^j$$

$$y_0^{j..j+d} = \tilde{y}_0^{j..j+d} \oplus \delta_0^{j..j+d}$$



😈 Extension to multiple bits

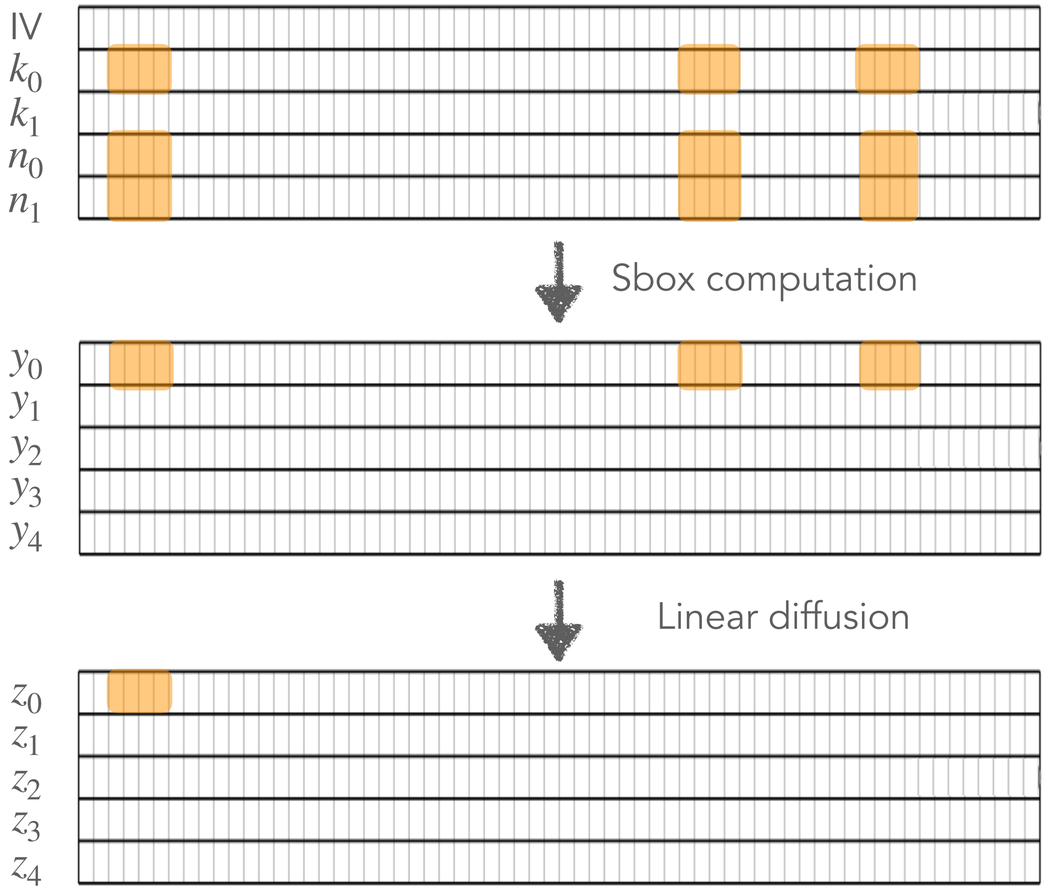
$$y_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j \oplus k_1^j k_0^j \oplus k_0^j \mathrm{IV}^j \oplus k_1^j \oplus \mathrm{IV}^j$$

$$y_0^j = \tilde{y}_0^j \oplus \delta_0^j$$

$$y_0^{j..j+d} = \tilde{y}_0^{j..j+d} \oplus \delta_0^{j..j+d}$$

$$y_0^{j+19..j+19+d} = \tilde{y}_0^{j+19..j+19+d} \oplus \delta_0^{j+19..j+19+d}$$

$$y_0^{j+28..j+28+d} = \tilde{y}_0^{j+28..j+28+d} \oplus \delta_0^{j+28..j+28+d}$$



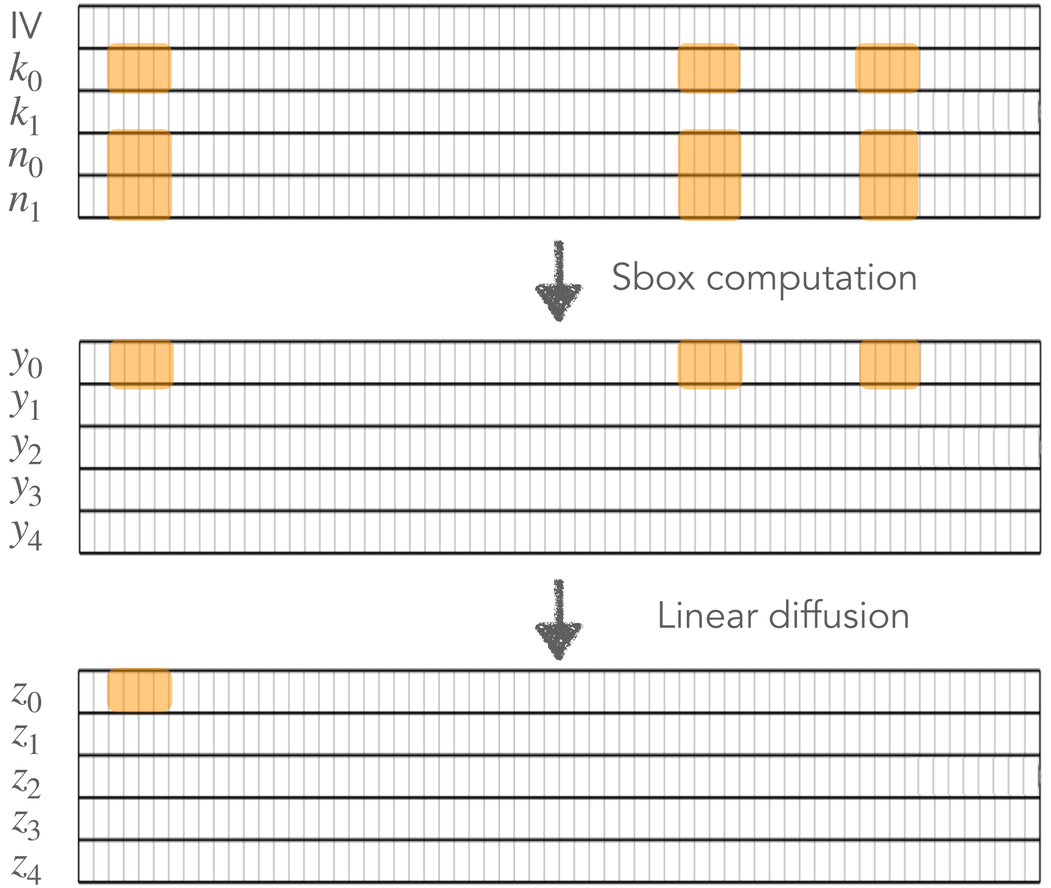
😈 Extension to multiple bits

$$y_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j \oplus k_1^j k_0^j \oplus k_0^j \mathrm{IV}^j \oplus k_1^j \oplus \mathrm{IV}^j$$

$$y_0^j = \tilde{y}_0^j \oplus \delta_0^j$$

$$\begin{aligned}
y_0^{j..j+d} &= \tilde{y}_0^{j..j+d} \oplus \delta_0^{j..j+d} \\
y_0^{j+19..j+19+d} &= \tilde{y}_0^{j+19..j+19+d} \oplus \delta_0^{j+19..j+19+d} \\
y_0^{j+28..j+28+d} &= \tilde{y}_0^{j+28..j+28+d} \oplus \delta_0^{j+28..j+28+d}
\end{aligned}$$

$$z_0^{j..j+d} = \tilde{z}_0^{j..j+d} \oplus \Delta_0^{j..j+d}$$



😈 Extension to multiple bits

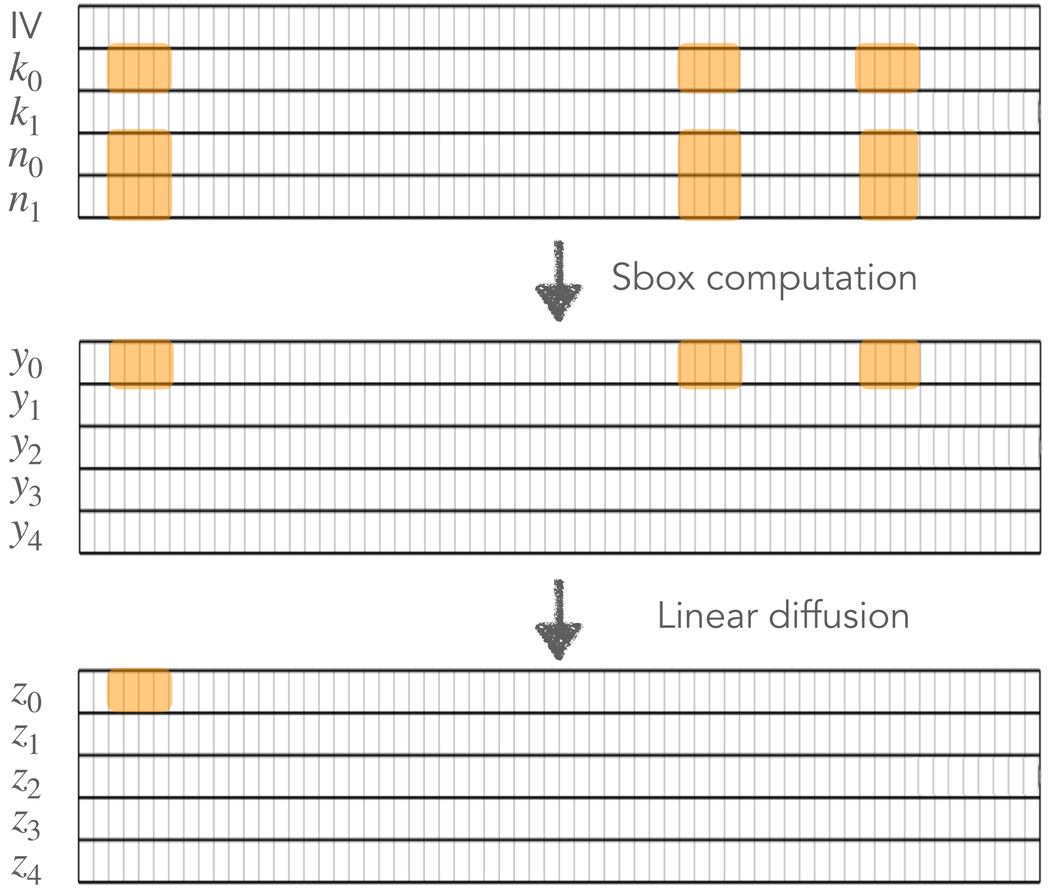
$$y_0^j = k_0^j(n_1^j \oplus 1) \oplus n_0^j \oplus k_1^j k_0^j \oplus k_0^j \mathrm{IV}^j \oplus k_1^j \oplus \mathrm{IV}^j$$

$$y_0^j = \tilde{y}_0^j \oplus \delta_0^j$$

$$\begin{aligned} y_0^{j..j+d} &= \tilde{y}_0^{j..j+d} \oplus \delta_0^{j..j+d} \\ y_0^{j+19..j+19+d} &= \tilde{y}_0^{j+19..j+19+d} \oplus \delta_0^{j+19..j+19+d} \\ y_0^{j+28..j+28+d} &= \tilde{y}_0^{j+28..j+28+d} \oplus \delta_0^{j+28..j+28+d} \end{aligned}$$

$$z_0^{j..j+d} = \tilde{z}_0^{j..j+d} \oplus \Delta_0^{j..j+d}$$

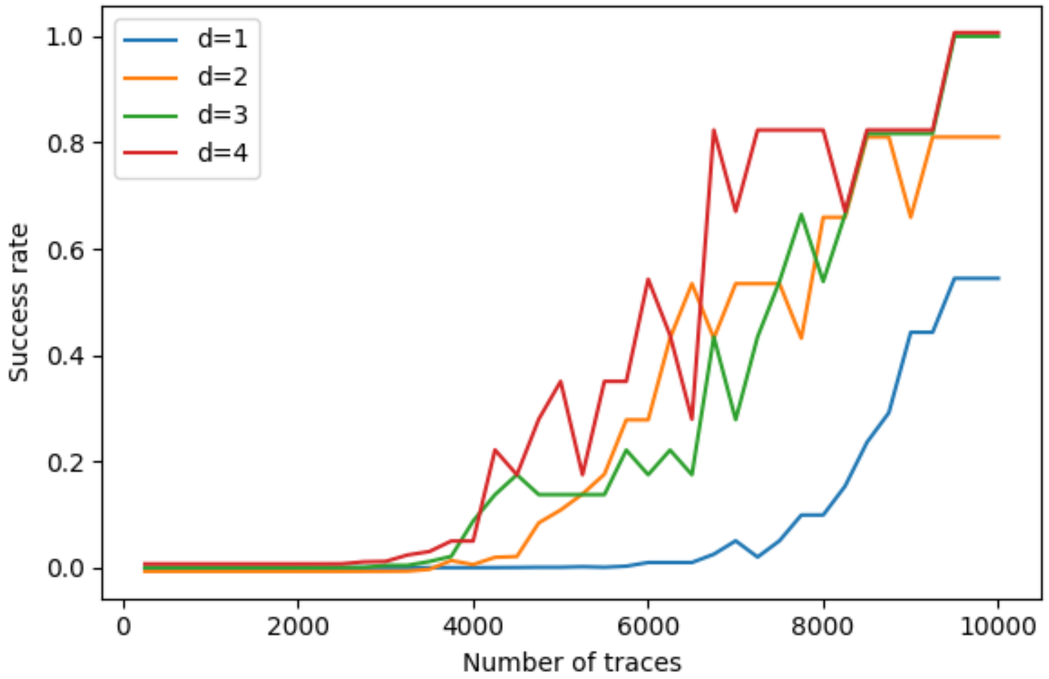
to be guessed
(apart from the key bits)



😈 Extension to multiple bits

Number of bits for $z_0^{j..j+d}$	d	$d=1$	$d=2$	$d=3$	$d=4$
Number of candidates (key and $\Delta_0^{j..j+d}$)	2^{4d-1}	2^3	2^7	2^{11}	2^{15}
Measured time (sequential recovery)		3m	10m	2.7h	34.7h
Measured time (parallel recovery)				8m	2.3h

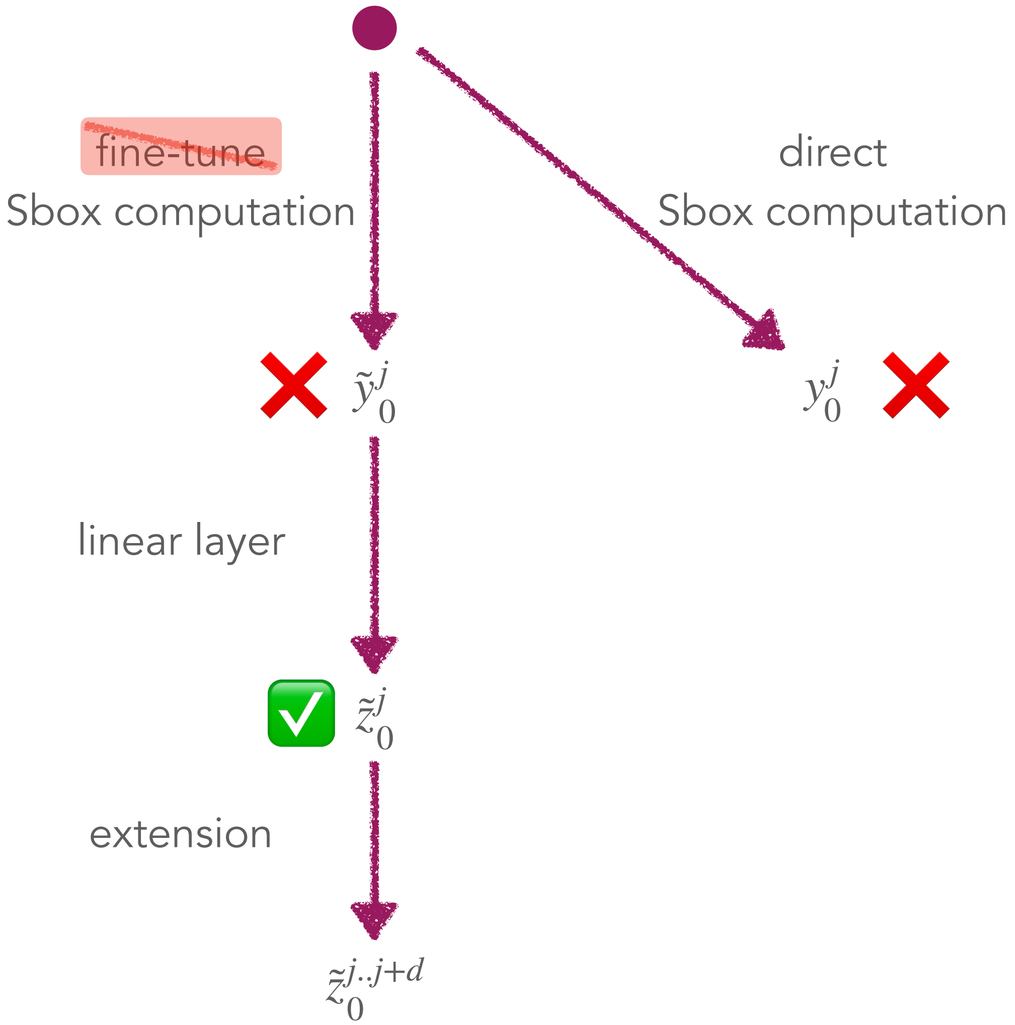
😈 Extension to multiple bits



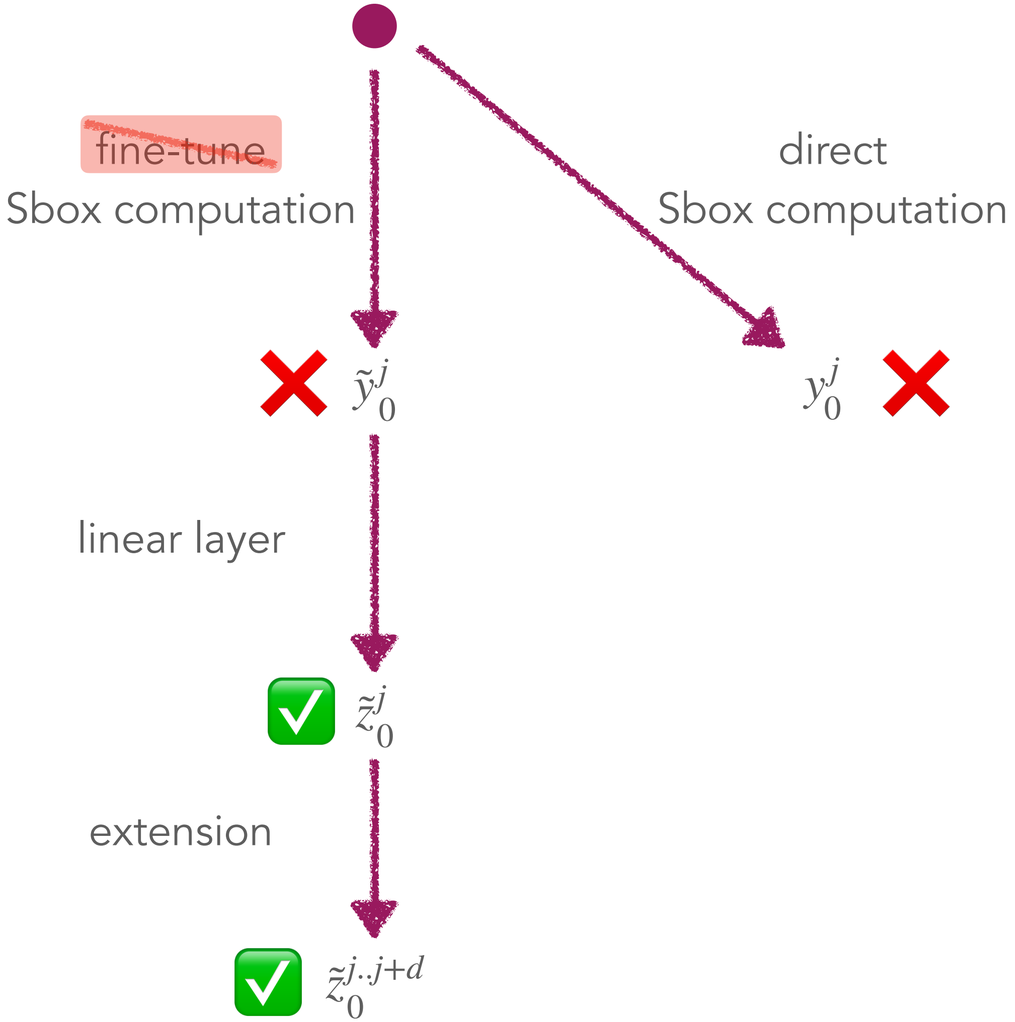
higher success rates when $d > 1$

Number of bits for $z_0^{j..j+d}$	d	$d=1$	$d=2$	$d=3$	$d=4$
Number of candidates (key and $\Delta_0^{j..j+d}$)	2^{4d-1}	2^3	2^7	2^{11}	2^{15}
Measured time (sequential recovery)		3m	10m	2.7h	34.7h
Measured time (parallel recovery)				8m	2.3h

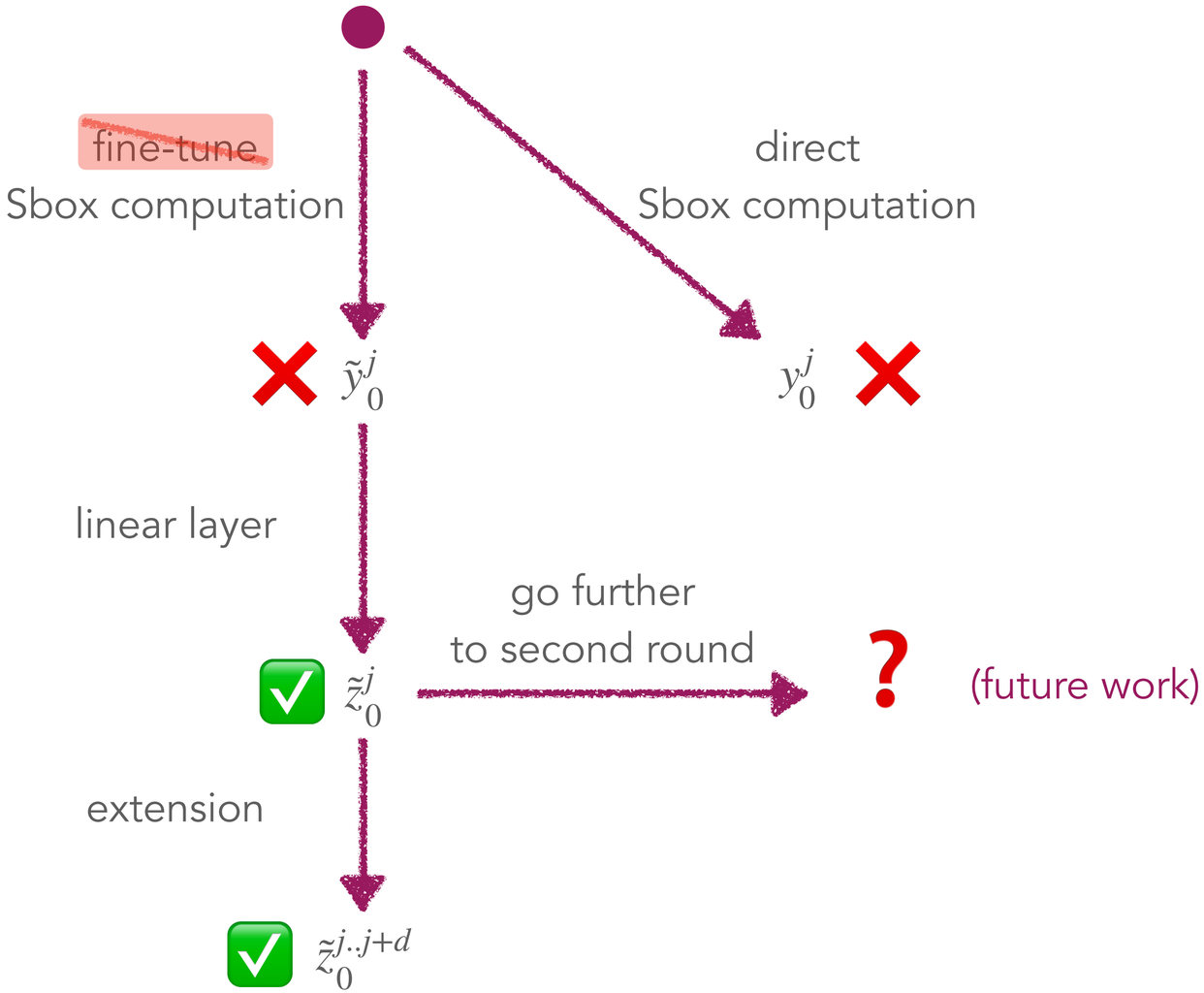
Map of selection functions

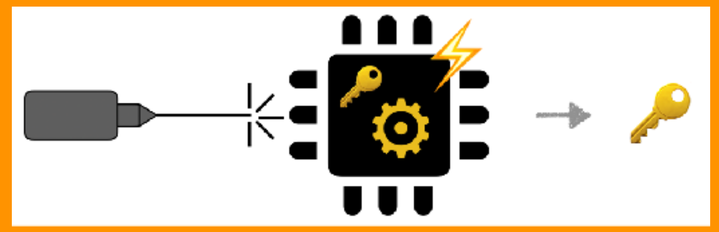


Map of selection functions



Map of selection functions





Statistical Ineffective Fault Attacks (SIFA) on Ascon-AEAD

👨‍🎓 Contribution 3:

Motivation

AEAD schemes use unique nonce for each encryption

Motivation

AEAD schemes use unique nonce for each encryption

Differential Fault Attacks (DFA) are not applicable

Motivation

AEAD schemes use unique nonce for each encryption

Differential Fault Attacks (DFA) are not applicable

[DMMP18] proposed strategy to apply SIFA with demonstration on 2 ciphers

Motivation

AEAD schemes use unique nonce for each encryption

Differential Fault Attacks (DFA) are not applicable

[DMMP18] proposed strategy to apply SIFA with demonstration on 2 ciphers

conjectured that this attack strategy is applicable to Ascon-AEAD

Motivation

AEAD schemes use unique nonce for each encryption

Differential Fault Attacks (DFA) are not applicable

[DMMP18] proposed strategy to apply SIFA with demonstration on 2 ciphers

conjectured that this attack strategy is applicable to Ascon-AEAD

I tried, but failed

Motivation

AEAD schemes use unique nonce for each encryption

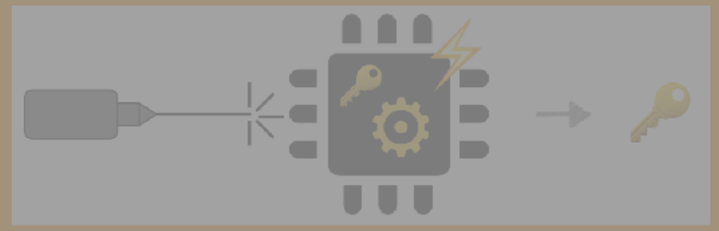
Differential Fault Attacks (DFA) are not applicable

[DMMP18] proposed strategy to apply SIFA with demonstration on 2 ciphers

conjectured that this attack strategy is applicable to Ascon-AEAD

I tried, but failed

Why did I fail?

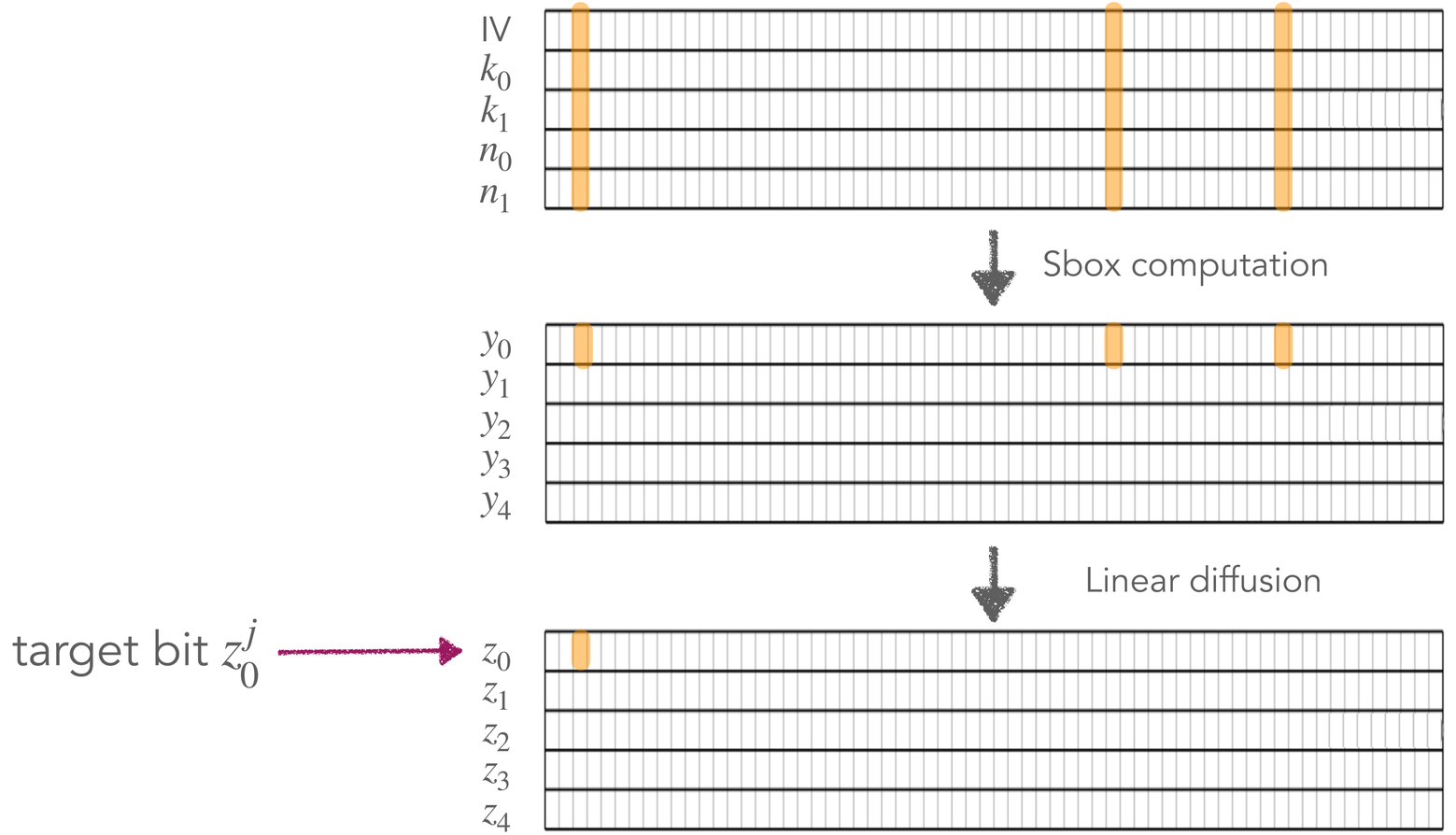


Statistical Ineffective Fault Attacks (SIFA) on Ascon-AEAD

👨‍🎓 Contribution 3:

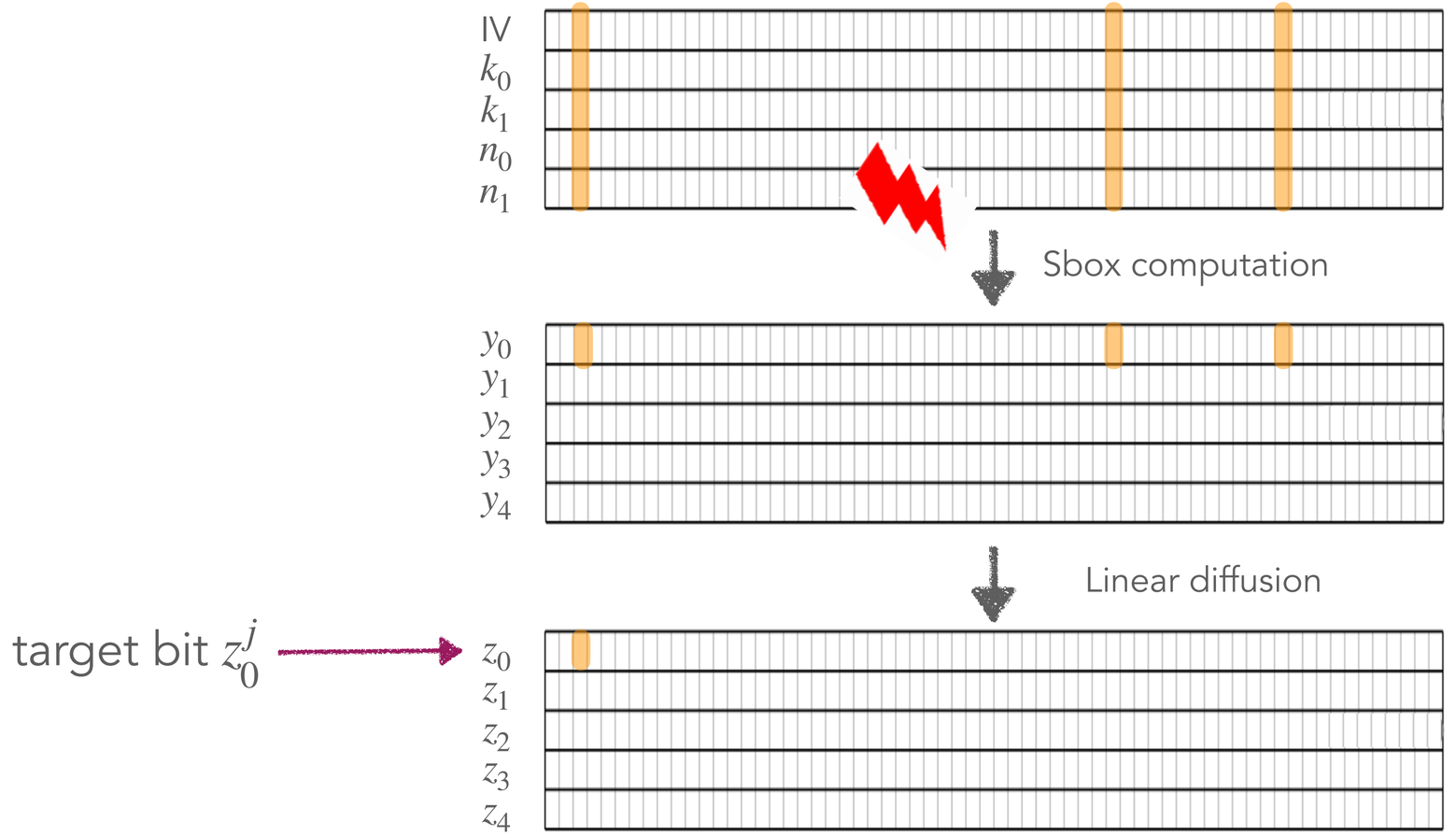
SIFA on Ascon-AEAD

(following the attack strategy of [DMMP18])



SIFA on Ascon-AEAD

(following the attack strategy of [DMMP18])



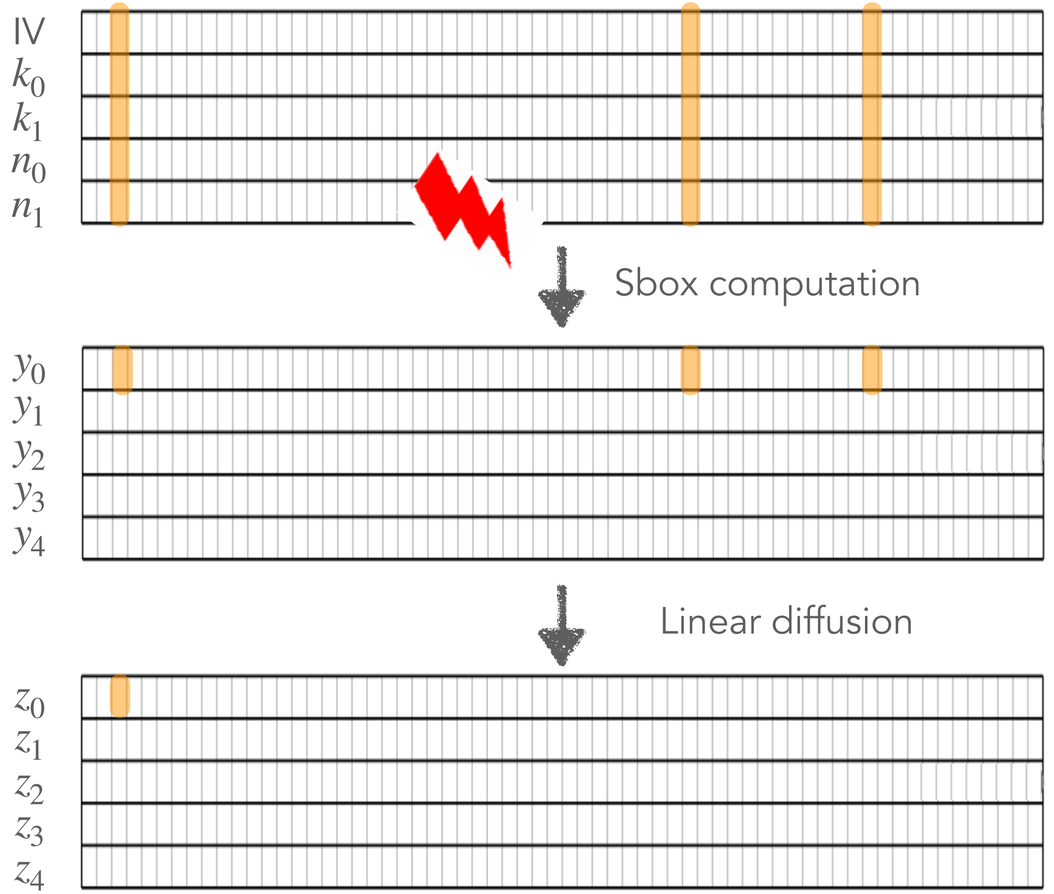
SIFA on Ascon-AEAD

(following the attack strategy of [DMMP18])

Suppose a fault countermeasure is in place

Can only collect <u>correct</u> outputs
(ineffective faults)

target bit z_0^j



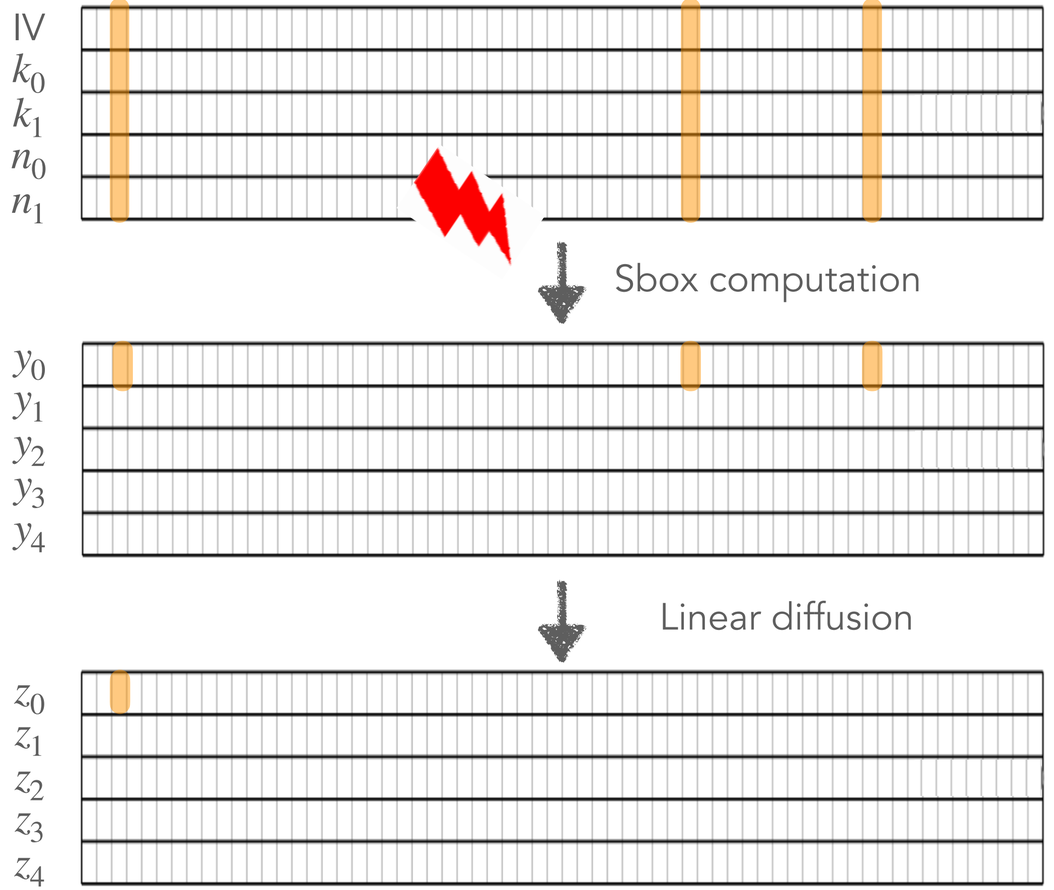
SIFA on Ascon-AEAD

(following the attack strategy of [DMMP18])

Suppose a fault countermeasure is in place

Can only collect <u>correct</u> outputs (ineffective faults)

target bit z_0^j



SIFA on Ascon-AEAD

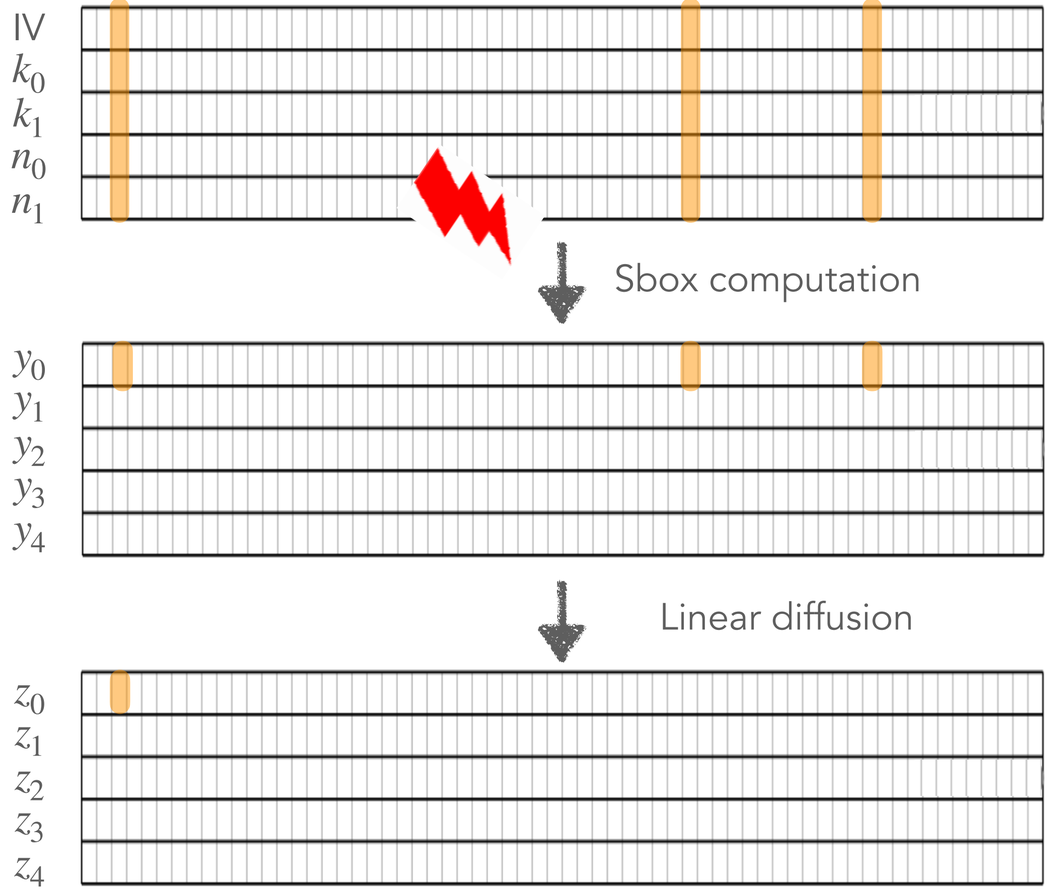
(following the attack strategy of [DMMP18])

Suppose a fault countermeasure is in place

Can only collect correct outputs (ineffective faults)

Example: $x \xrightarrow{\text{stuck-at-0}} x'$

target bit z_0^j



SIFA on Ascon-AEAD

(following the attack strategy of [DMMP18])

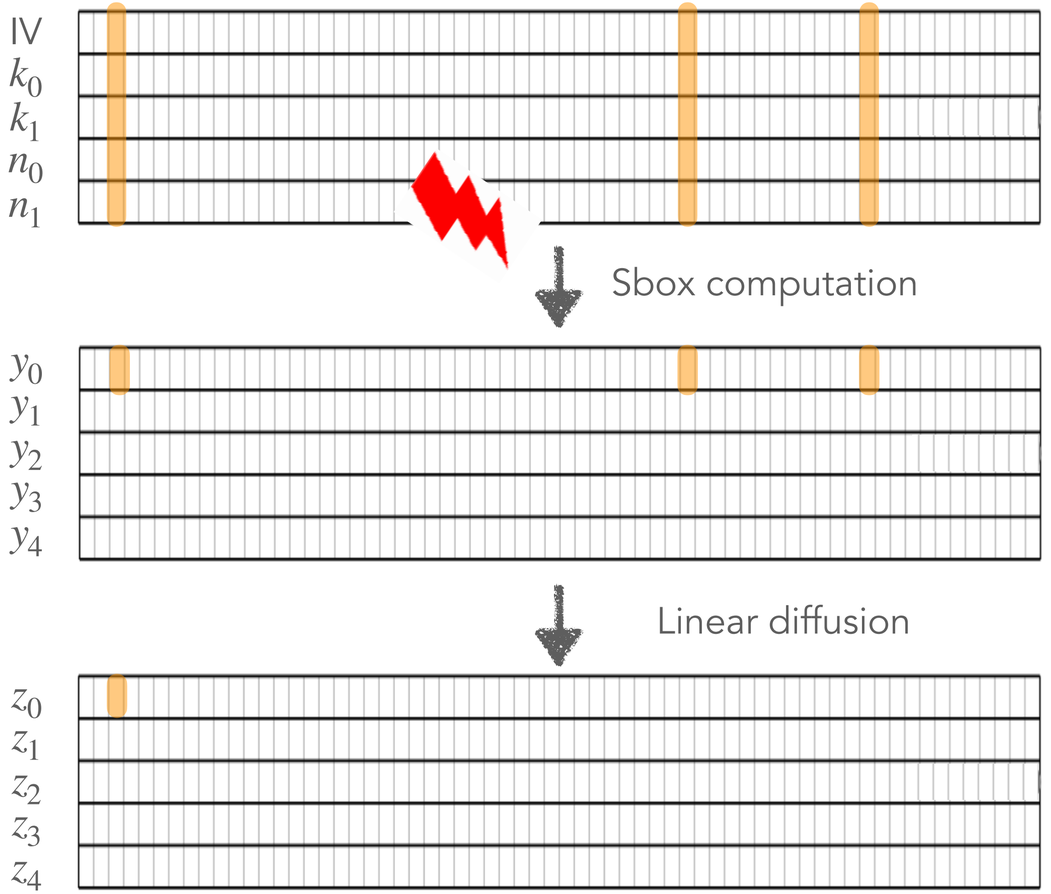
Suppose a fault countermeasure is in place

Can only collect correct outputs (ineffective faults)

target bit z_0^j

Example: x —stuck-at-0→ x'

Probability	x'	
x becomes x'	0	1
x 0	1	0
x 1	1	0



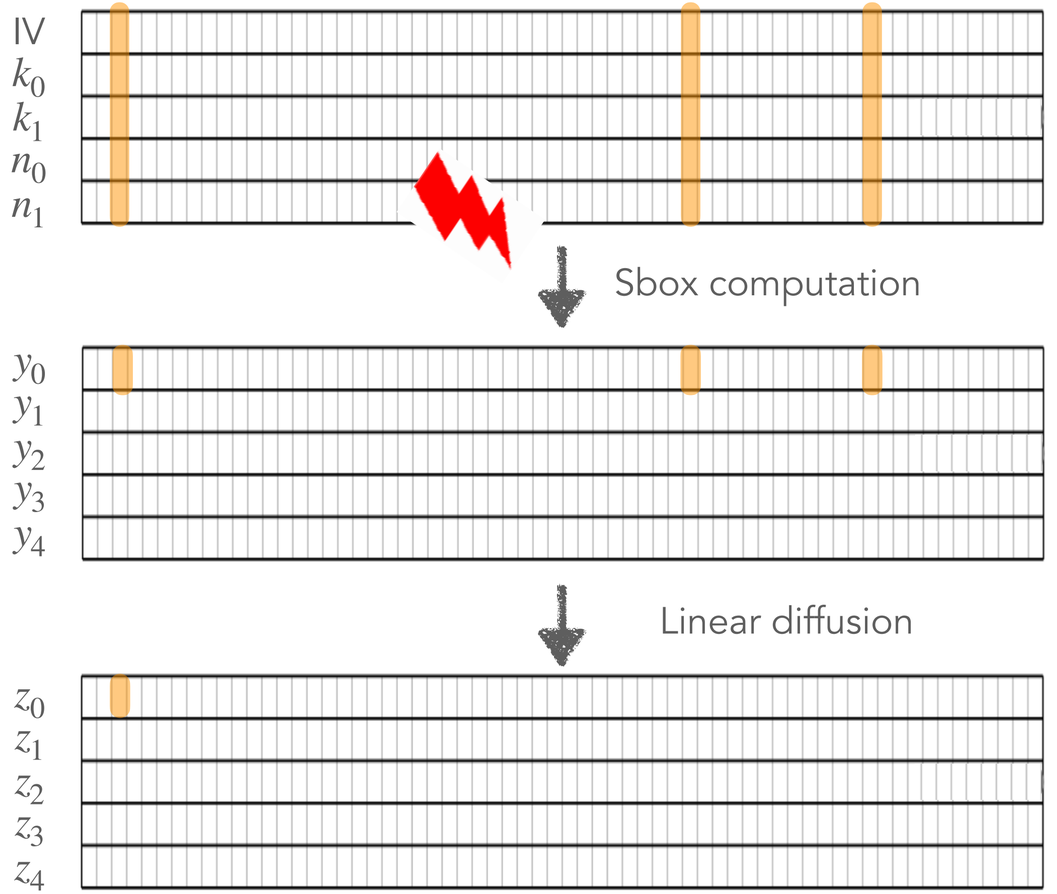
SIFA on Ascon-AEAD

(following the attack strategy of [DMMP18])

Suppose a fault countermeasure is in place

Can only collect correct outputs (ineffective faults)

target bit z_0^j



Example: x —stuck-at-0→ x'

Probability: x becomes x'	$x'=0$	$x'=1$
$x=0$	1	0
$x=1$	1	0

→ biased distribution of ineffective faults

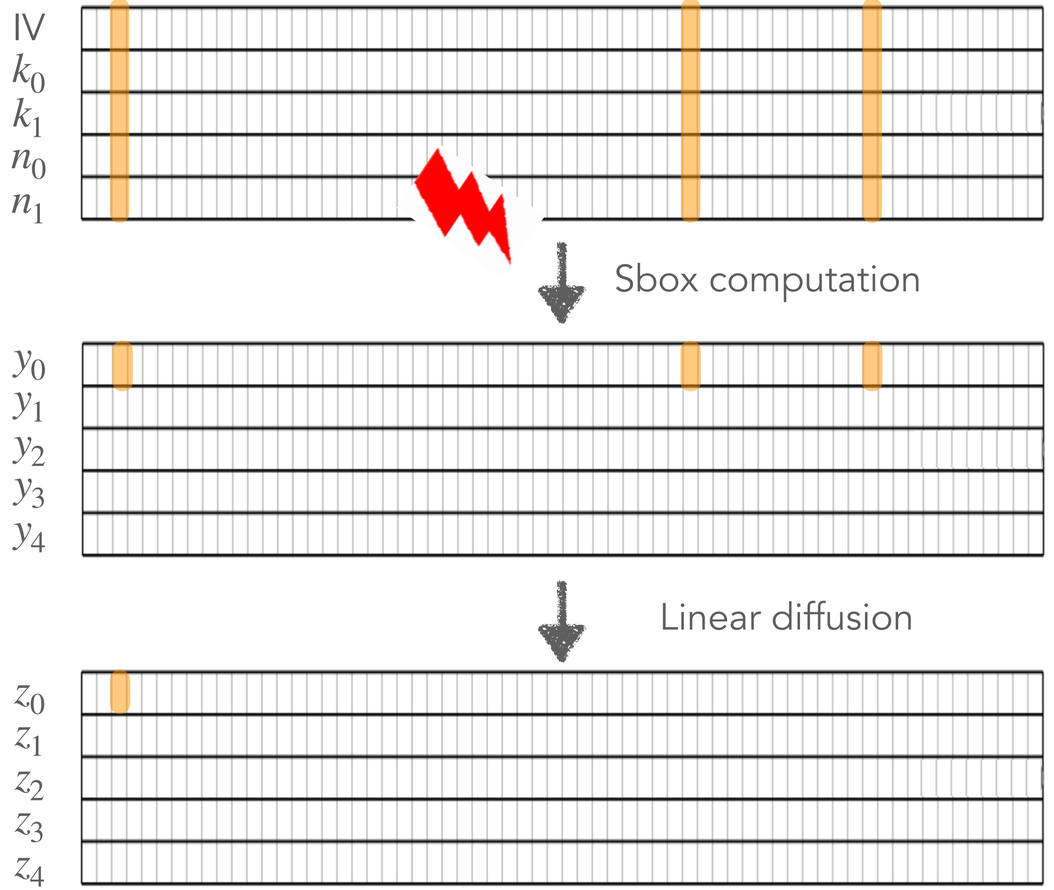
SIFA on Ascon-AEAD

(following the attack strategy of [DMMP18])

Suppose a fault countermeasure is in place

Can only collect <u>correct</u> outputs (ineffective faults)

target bit z_0^j



SIFA on Ascon-AEAD

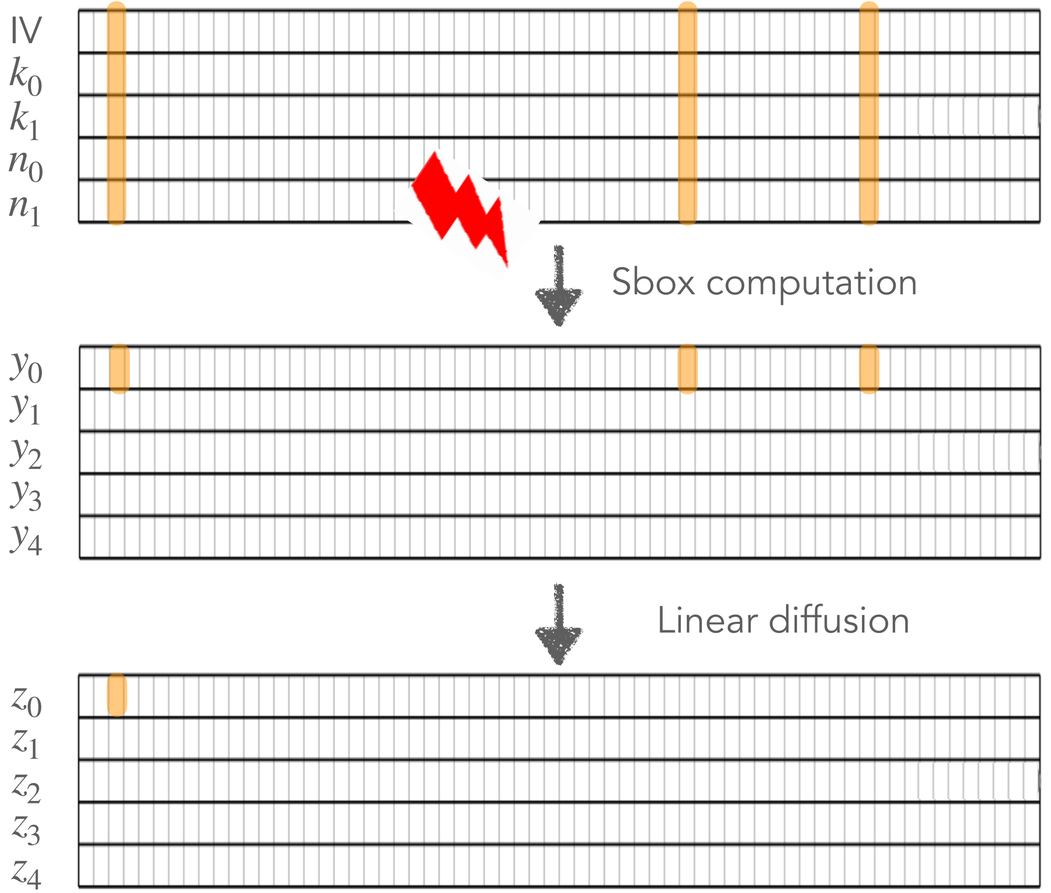
(following the attack strategy of [DMMP18])

Suppose a fault countermeasure is in place

Can only collect <u>correct</u> outputs (ineffective faults)

<u>Attack</u>: compute distribution of z_0^j

target bit z_0^j



SIFA on Ascon-AEAD

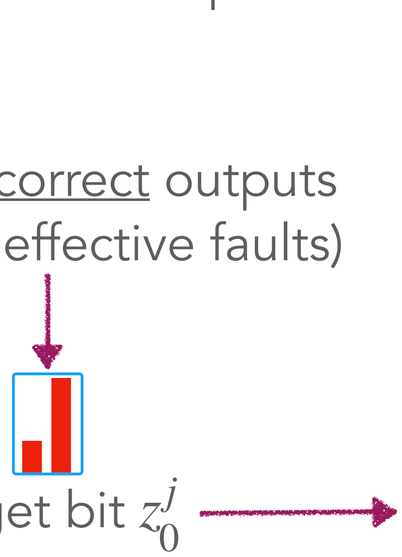
(following the attack strategy of [DMMP18])

Suppose a fault countermeasure is in place

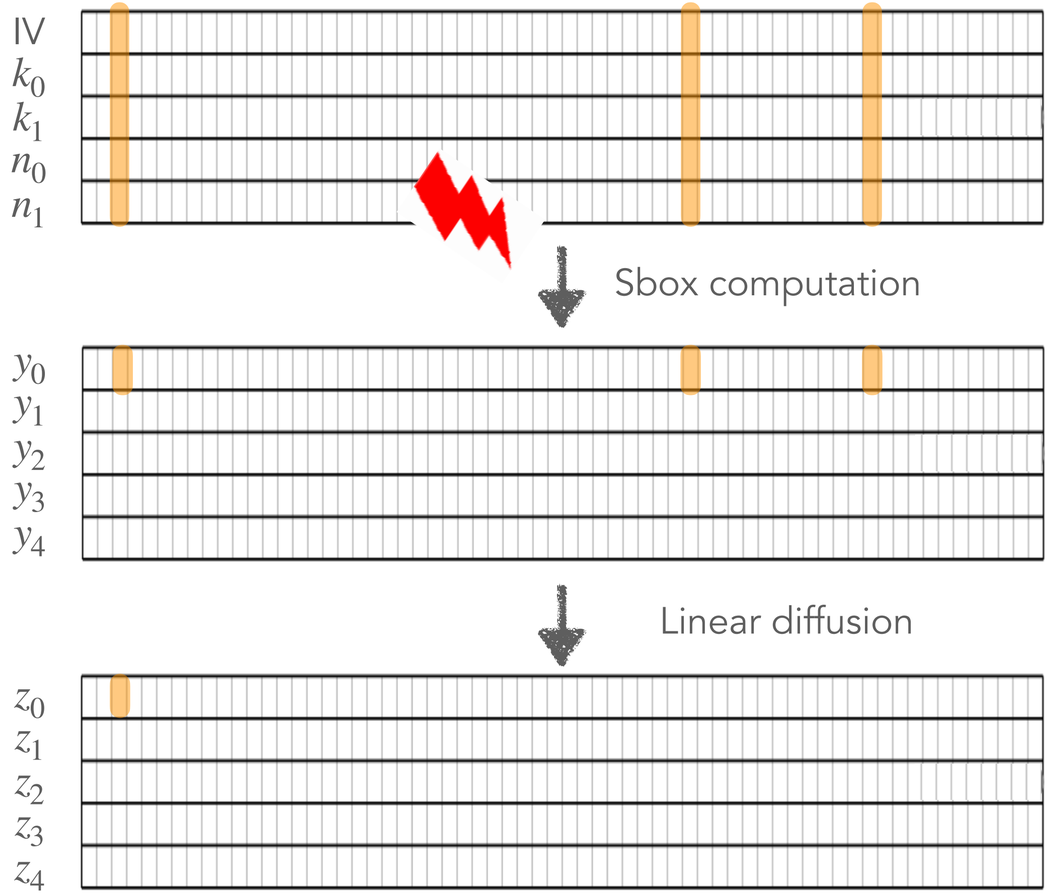
Can only collect correct outputs (ineffective faults)

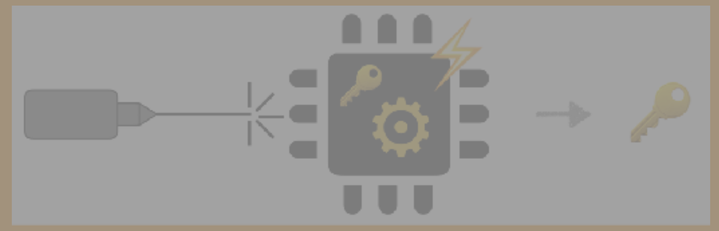
Attack: compute distribution of z_0^j

- correct key guess →
- wrong key guess →



target bit z_0^j





Statistical Ineffective Fault Attacks (SIFA) on Ascon-AEAD

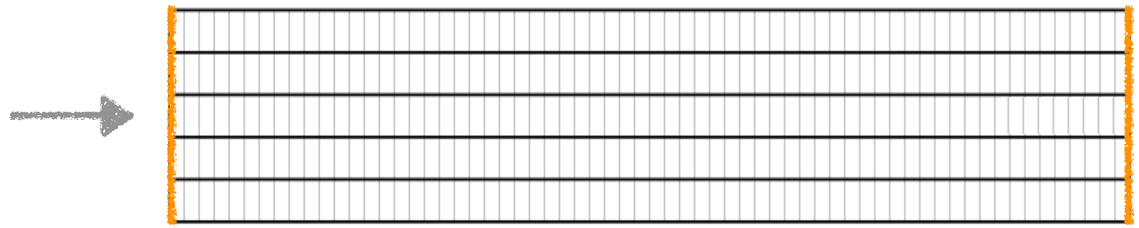
👨‍🎓 Contribution 3: In-depth analysis of SIFA with instruction-skip fault

Nguyen, Grosso, Cayrel - FDTC 2025

Implementations of Ascon-AEAD

On 320-bit state = 5 x 64-bit words

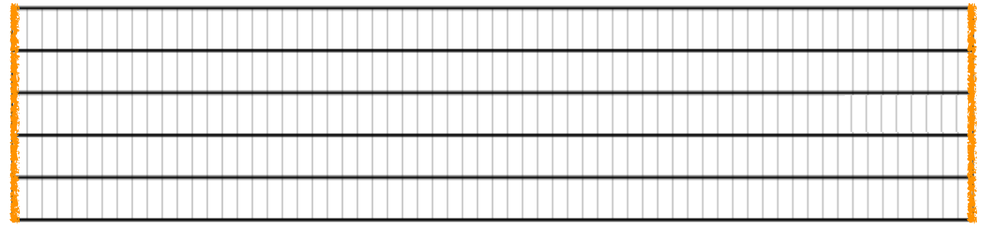
64-bit architecture



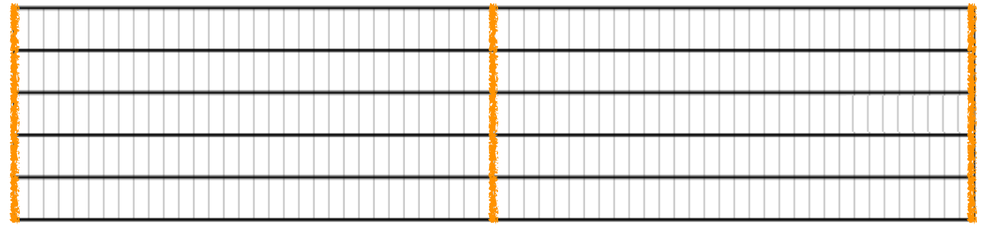
Implementations of Ascon-AEAD

On 320-bit state = 5 x 64-bit words

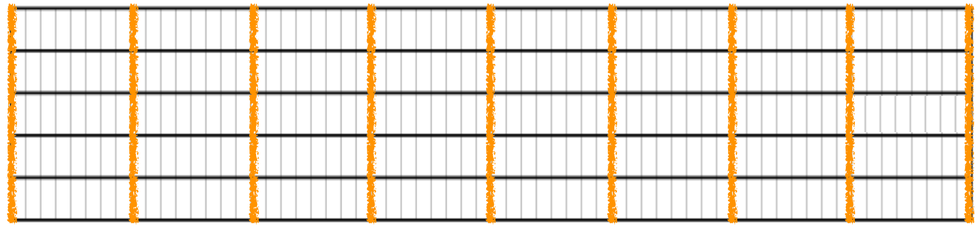
64-bit architecture



32-bit architecture



8-bit architecture



Instruction-skip model

Instruction-skip model

$\mathtt{OP}_0 \quad \mathtt{R}_2 \quad - \quad -$

$\cdots$

$\mathtt{XOR} \quad \mathtt{R}_2 \quad \mathtt{R}_1 \quad \mathtt{R}_0$

$\cdots$

$\mathtt{OP}_2 \quad - \quad \mathtt{R}_2 \quad -$

Scenario $1 : \mathtt{R}_2 = \mathtt{R}_1 \oplus \mathtt{R}_0$

Instruction-skip model

$\mathtt{OP}_0 \quad \mathtt{R}_2 \quad - \quad -$

$\cdots$

$\mathtt{XOR} \quad \mathtt{R}_2 \quad \mathtt{R}_1 \quad \mathtt{R}_0$

$\cdots$

$\mathtt{OP}_2 \quad - \quad \mathtt{R}_2 \quad -$

Scenario 1 : $\mathtt{R}_2 = \mathtt{R}_1 \oplus \mathtt{R}_0$

No fault	Fault occurs
$R_2 = v_2$	

Instruction-skip model

	No fault	Fault occurs
$\mathtt{OP}_0$ $\mathtt{R}_2$ $-$ $-$	$R_2 = v_2$	
...		
$\mathtt{XOR}$ $\mathtt{R}_2$ $\mathtt{R}_1$ $\mathtt{R}_0$	$R_2 = v_2' = v_1 \oplus v_0$	
...		
$\mathtt{OP}_2$ $-$ $\mathtt{R}_2$ $-$		

Scenario 1 : $\mathtt{R}_2 = \mathtt{R}_1 \oplus \mathtt{R}_0$

Instruction-skip model

	No fault	Fault occurs
$\mathtt{OP}_0 \quad \mathtt{R}_2 \quad - \quad -$	$R_2 = v_2$	
…		
$\mathtt{XOR} \quad \mathtt{R}_2 \quad \mathtt{R}_1 \quad \mathtt{R}_0$	$R_2 = v_2' = v_1 \oplus v_0$	
…		
$\mathtt{OP}_2 \quad - \quad \mathtt{R}_2 \quad -$	v_2' is used	

Scenario 1 : $\mathtt{R}_2 = \mathtt{R}_1 \oplus \mathtt{R}_0$

Instruction-skip model

	No fault	Fault occurs
$\mathtt{OP}_0 \quad \mathtt{R}_2 \quad - \quad -$	$R_2 = v_2$	$R_2 = v_2$
…		
$\mathtt{XOR} \quad \mathtt{R}_2 \quad \mathtt{R}_1 \quad \mathtt{R}_0$	$R_2 = v_2' = v_1 \oplus v_0$	
…		
$\mathtt{OP}_2 \quad - \quad \mathtt{R}_2 \quad -$	v_2' is used	

Scenario 1 : $\mathtt{R}_2 = \mathtt{R}_1 \oplus \mathtt{R}_0$

Instruction-skip model

	No fault	Fault occurs
$\mathtt{OP}_0 \quad \mathtt{R}_2 \quad - \quad -$	$R_2 = v_2$	$R_2 = v_2$
…		
$\mathtt{XOR} \quad \mathtt{R}_2 \quad \mathtt{R}_1 \quad \mathtt{R}_0$	$R_2 = v_2' = v_1 \oplus v_0$	(skipped)
…		
$\mathtt{OP}_2 \quad - \quad \mathtt{R}_2 \quad -$	v_2' is used	

Scenario 1 : $\mathtt{R}_2 = \mathtt{R}_1 \oplus \mathtt{R}_0$

Instruction-skip model

	No fault	Fault occurs
$\mathtt{OP_0}\ \ \mathtt{R_2}\ \ -\ \ -$	$R_2 = v_2$	$R_2 = v_2$
…		
$\mathtt{XOR}\ \ \mathtt{R_2}\ \ \mathtt{R_1}\ \ \mathtt{R_0}$	$R_2 = v_2' = v_1 \oplus v_0$	(skipped)
…		
$\mathtt{OP_2}\ \ -\ \ \mathtt{R_2}\ \ -$	v_2' is used	v_2 is used

Scenario 1 : $\mathtt{R_2} = \mathtt{R_1} \oplus \mathtt{R_0}$

Instruction-skip model

$\mathtt{OP}_0 \quad \mathtt{R}_2 \quad - \quad -$

$\ldots$

$\mathtt{XOR} \quad \mathtt{R}_2 \quad \mathtt{R}_1 \quad \mathtt{R}_0$

$\ldots$

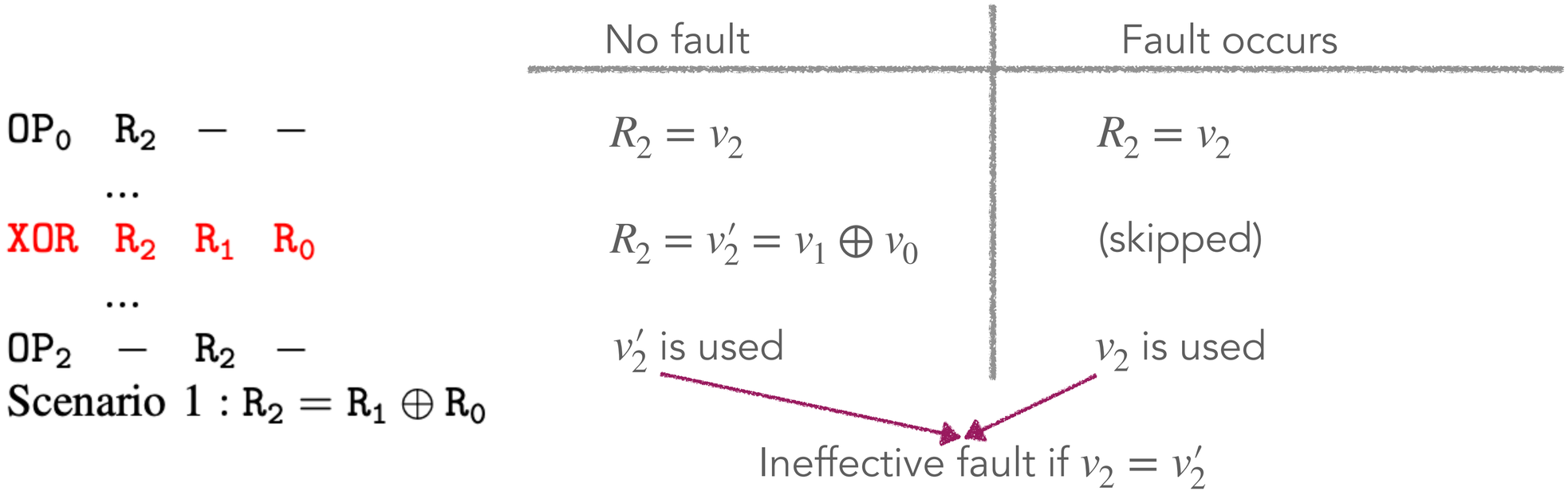
$\mathtt{OP}_2 \quad - \quad \mathtt{R}_2 \quad -$

Scenario 1 : $\mathtt{R}_2 = \mathtt{R}_1 \oplus \mathtt{R}_0$

No fault	Fault occurs
$R_2 = v_2$	$R_2 = v_2$
$R_2 = v_2' = v_1 \oplus v_0$	(skipped)
v_2' is used	v_2 is used

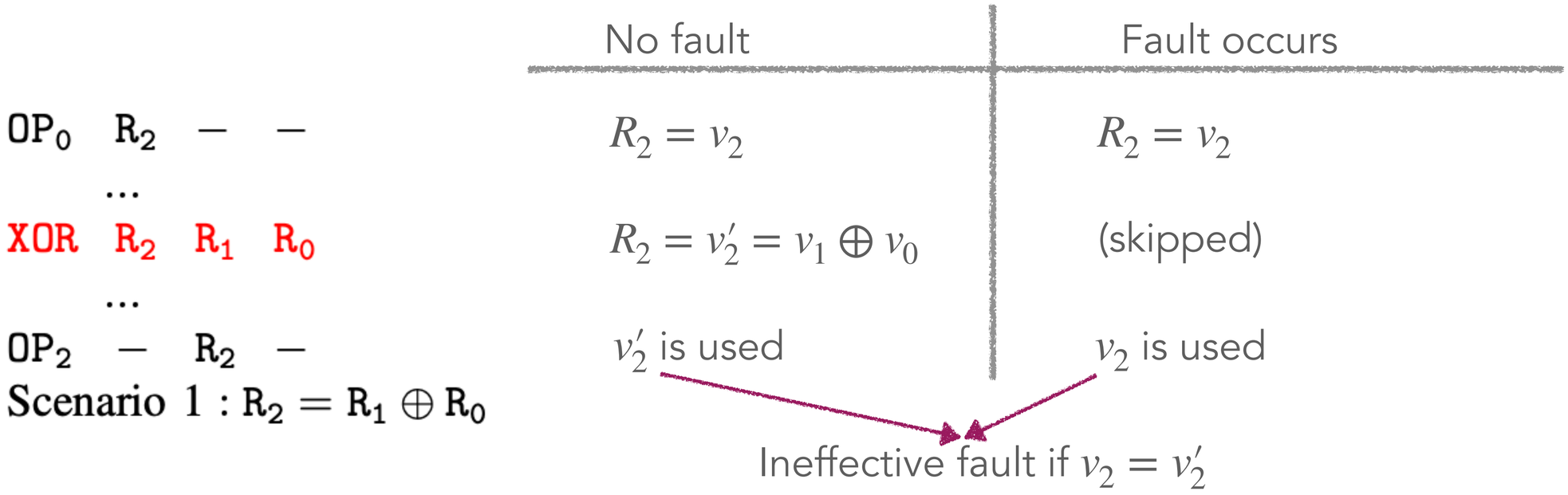
Ineffective fault if $v_2 = v_2'$

Instruction-skip model



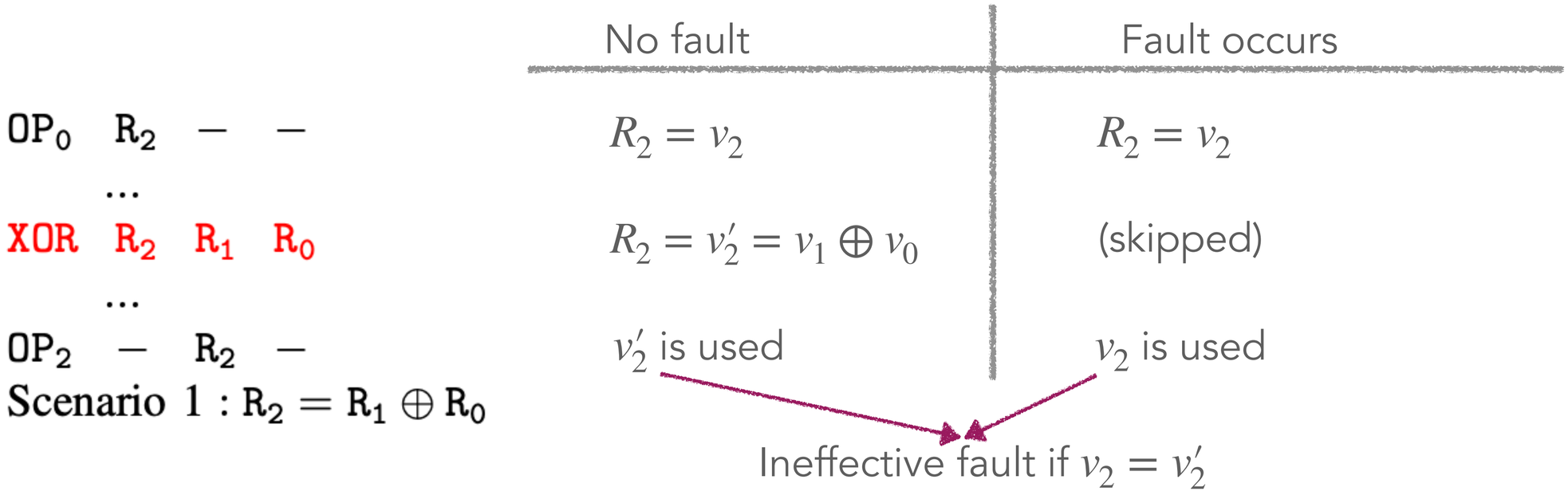
Suppose v_0, v_1, v_2 are uniform,

Instruction-skip model



Suppose v_0, v_1, v_2 are uniform, then $\Pr[v_2 = v_2'] = 0.5^w$, w is register bit-width

Instruction-skip model



Suppose v_0, v_1, v_2 are uniform, then $\Pr[v_2 = v_2'] = 0.5^w$, w is register bit-width

Architecture	8-bit	32-bit	64-bit
$\Pr[v_2 = v_2']$			

Instruction-skip model

$\mathtt{OP_0}\ \ \mathtt{R_2}\ \ -\ \ -$

$\cdots$

$\mathtt{XOR}\ \ \mathtt{R_2}\ \ \mathtt{R_1}\ \ \mathtt{R_0}$

$\cdots$

$\mathtt{OP_2}\ \ -\ \ \mathtt{R_2}\ \ -$

Scenario 1 : $\mathtt{R_2} = \mathtt{R_1} \oplus \mathtt{R_0}$

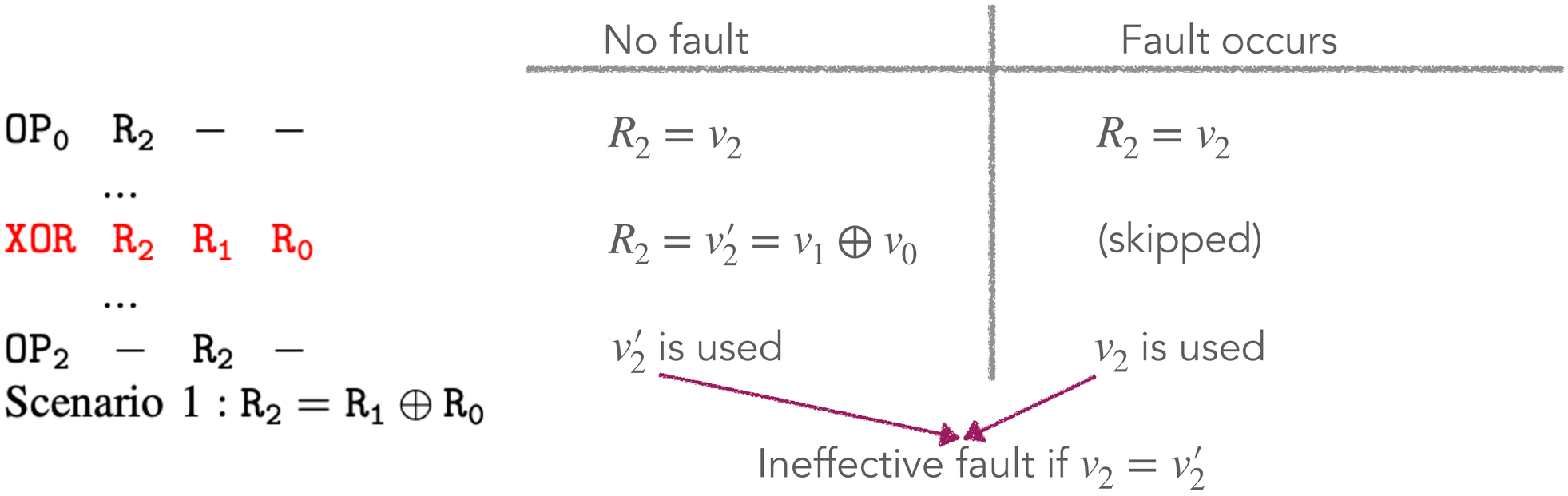
No fault	Fault occurs
$R_2 = v_2$	$R_2 = v_2$
$R_2 = v_2' = v_1 \oplus v_0$	(skipped)
v_2' is used	v_2 is used

Ineffective fault if $v_2 = v_2'$

Suppose v_0, v_1, v_2 are uniform, then $\Pr[v_2 = v_2'] = 0.5^w$, w is register bit-width

Architecture	8-bit	32-bit	64-bit
$\Pr[v_2 = v_2']$	≈ 0.0039		

Instruction-skip model



Suppose v_0, v_1, v_2 are uniform, then $\Pr[v_2 = v_2'] = 0.5^w$, w is register bit-width

Architecture	8-bit	32-bit	64-bit
$\Pr[v_2 = v_2']$	≈ 0.0039	≈ 0	≈ 0
	😈	😈	😈

Instruction-skip model

$\mathtt{OP_0}\ \ \mathtt{R_2}\ \ -\ \ -$

$\cdots$

$\mathtt{XOR}\ \ \mathtt{R_2}\ \ \mathtt{R_1}\ \ \mathtt{R_0}$

$\cdots$

$\mathtt{OP_2}\ \ -\ \ \mathtt{R_2}\ \ -$

Scenario 1 : $\mathtt{R_2} = \mathtt{R_1} \oplus \mathtt{R_0}$

	No fault	Fault occurs
	$R_2 = v_2$	$R_2 = v_2$
	$R_2 = v_2' = v_1 \oplus v_0$	(skipped)
	v_2' is used	v_2 is used

Ineffective fault if $v_2 = v_2'$

Suppose v_0, v_1, v_2 are uniform, then $\Pr[v_2 = v_2'] = 0.5^w$, w is register bit-width

Architecture	8-bit	32-bit	64-bit
$\Pr[v_2 = v_2']$	≈ 0.0039	≈ 0	≈ 0

Impractical to obtain ineffective fault ! 😈

Instruction-skip model

$\mathtt{OP}_0 \quad \mathtt{R}_2 \quad - \quad -$

$\ldots$

$\mathtt{XOR} \quad \mathtt{R}_2 \quad \mathtt{R}_1 \quad \mathtt{R}_0$

$\ldots$

$\mathtt{OP}_2 \quad - \quad \mathtt{R}_2 \quad -$

Scenario 1 : $\mathtt{R}_2 = \mathtt{R}_1 \oplus \mathtt{R}_0$

3-operand instruction

$\mathtt{OP}_0 \quad \mathtt{R}_2 \quad - \quad -$

$\ldots$

$\mathtt{XOR} \quad \mathtt{R}_2 \quad \mathtt{R}_2 \quad \mathtt{R}_0$

$\ldots$

$\mathtt{OP}_2 \quad - \quad \mathtt{R}_2 \quad -$

Scenario 2 : $\mathtt{R}_2 = \mathtt{R}_2 \oplus \mathtt{R}_0$

2-operand instruction

(or R_2 is reused as source register)

Instruction-skip model

	Probability of an ineffective fault $\Pr[v_2 = v_2']$
XOR (scenario 1)	0.5^w
XOR (scenario 2)	0.5^w

Instruction-skip model

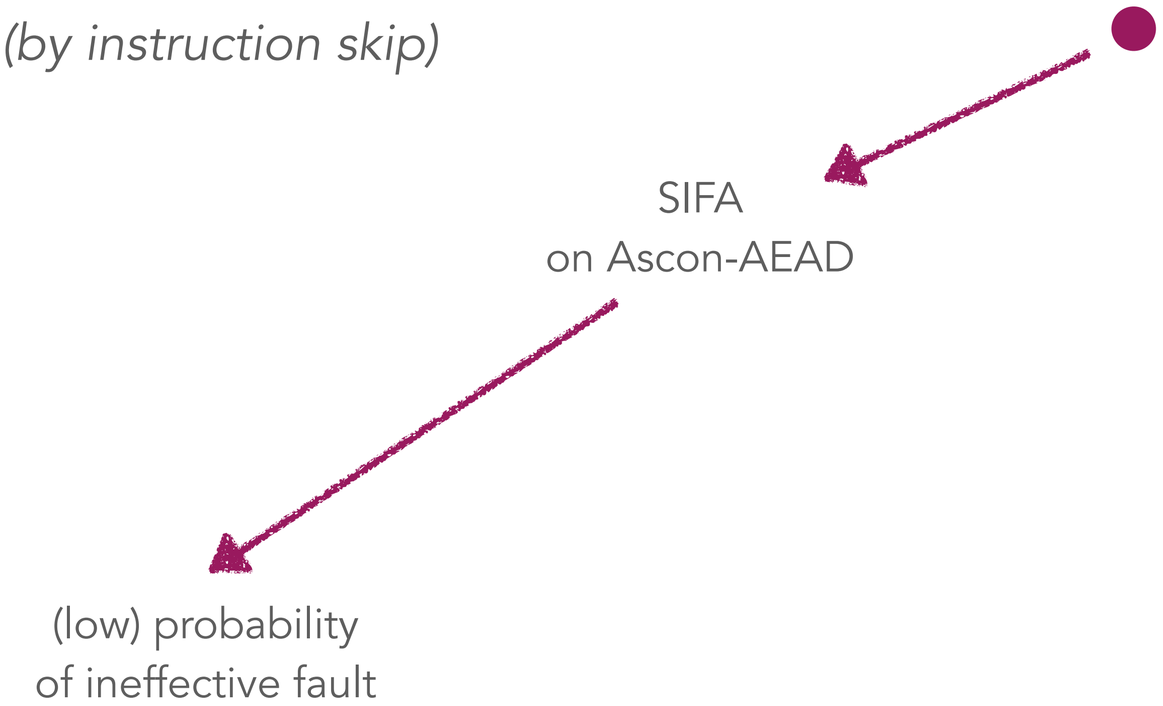
	Probability of an ineffective fault $\Pr[v_2 = v_2']$
XOR (scenario 1)	0.5^w
XOR (scenario 2)	0.5^w
AND (scenario 1)	0.5^w
AND (scenario 2)	0.75^w

Instruction-skip model

	Probability of an ineffective fault $\Pr[v_2 = v_2']$
XOR (scenario 1)	0.5^w
XOR (scenario 2)	0.5^w
AND (scenario 1)	0.5^w
AND (scenario 2)	0.75^w
NOT (scenario 1)	0.5^w
NOT (scenario 2)	0

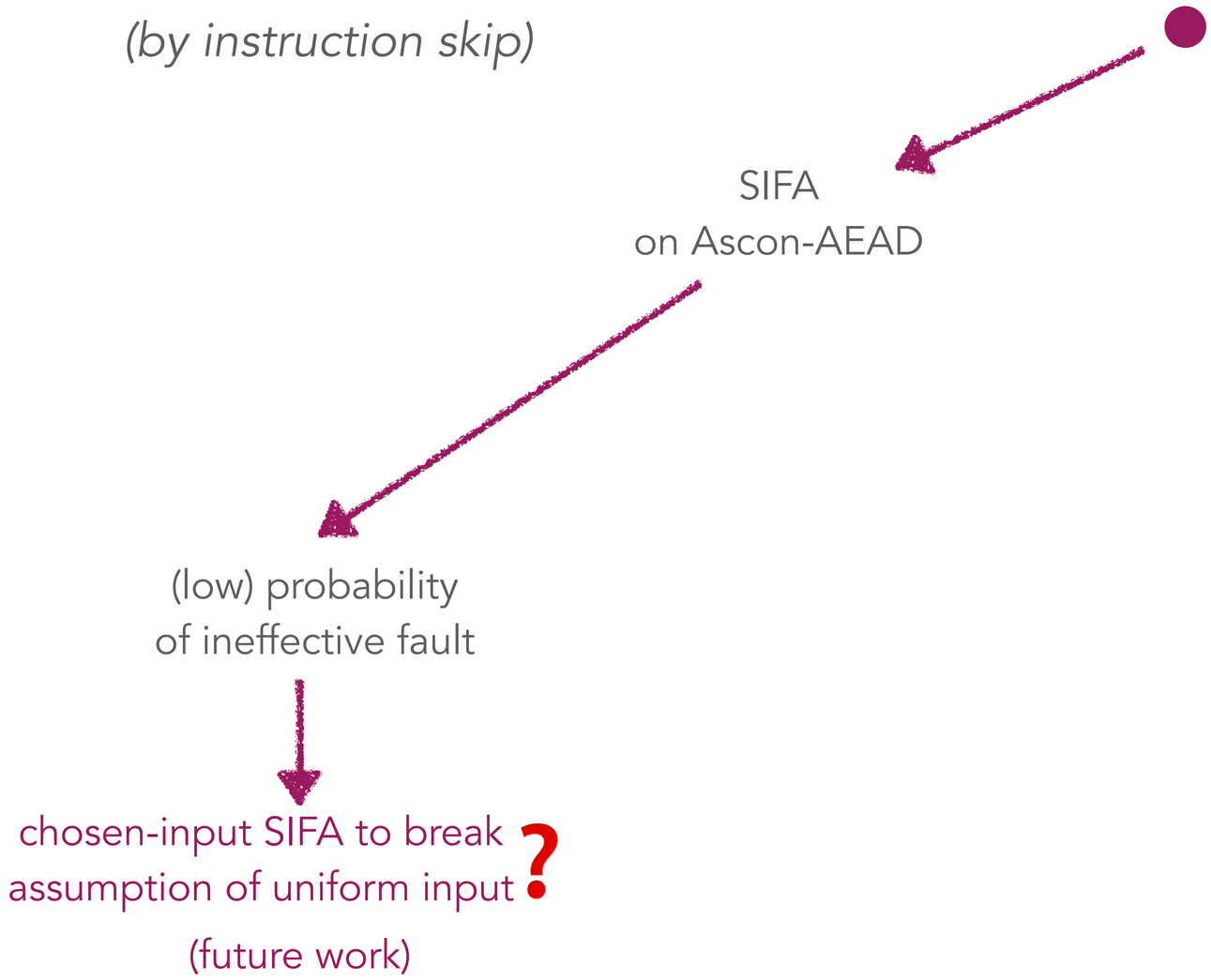
Map of fault attacks

(by instruction skip)



Map of fault attacks

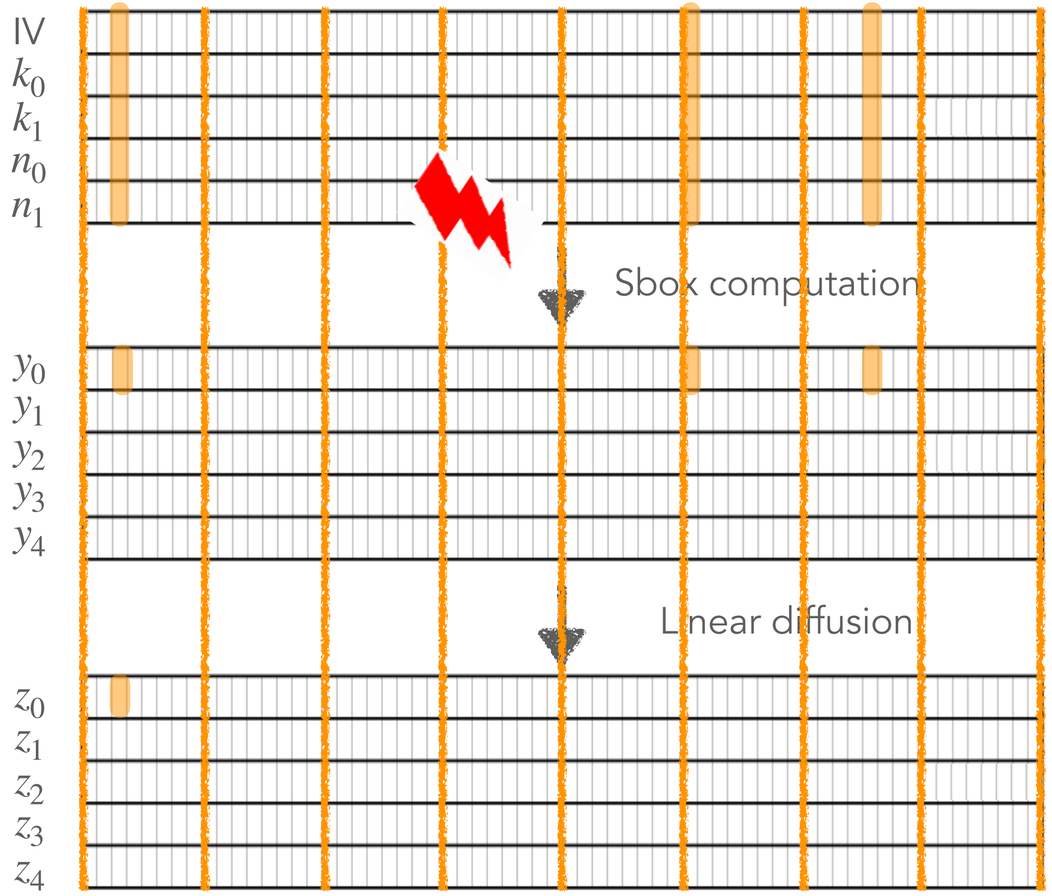
(by instruction skip)



SIFA on Ascon-AEAD

(following the attack strategy of [DMMP18])

on 8-bit architecture



target bit z_0^j

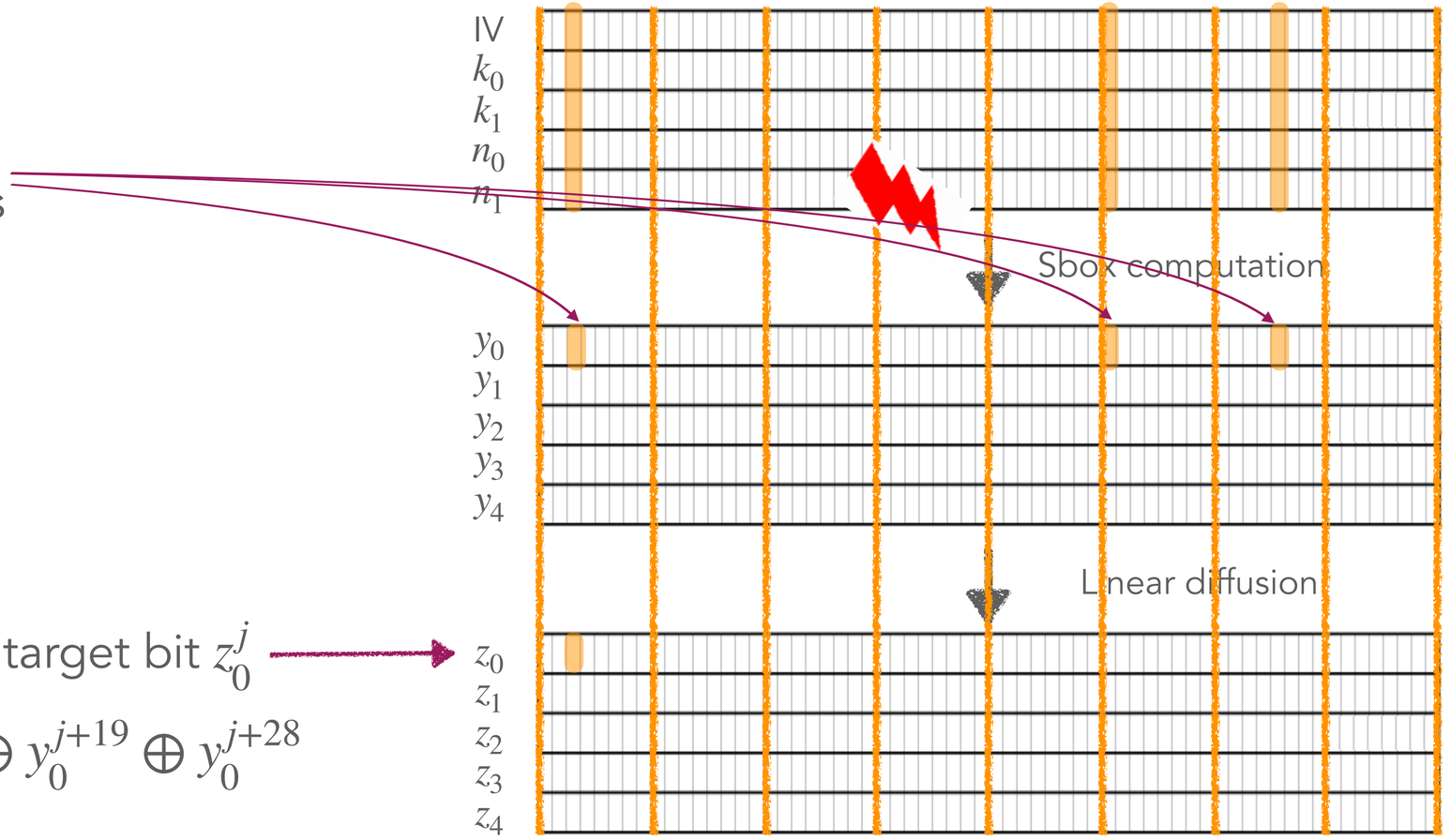
$$z_0^j = y_0^j \oplus y_0^{j+19} \oplus y_0^{j+28}$$

SIFA on Ascon-AEAD

(following the attack strategy of [DMMP18])

on 8-bit architecture

An instruction skip
does <u>not affect all</u> 3 bits



$$z_0^j = y_0^j \oplus y_0^{j+19} \oplus y_0^{j+28}$$

SIFA on Ascon-AEAD

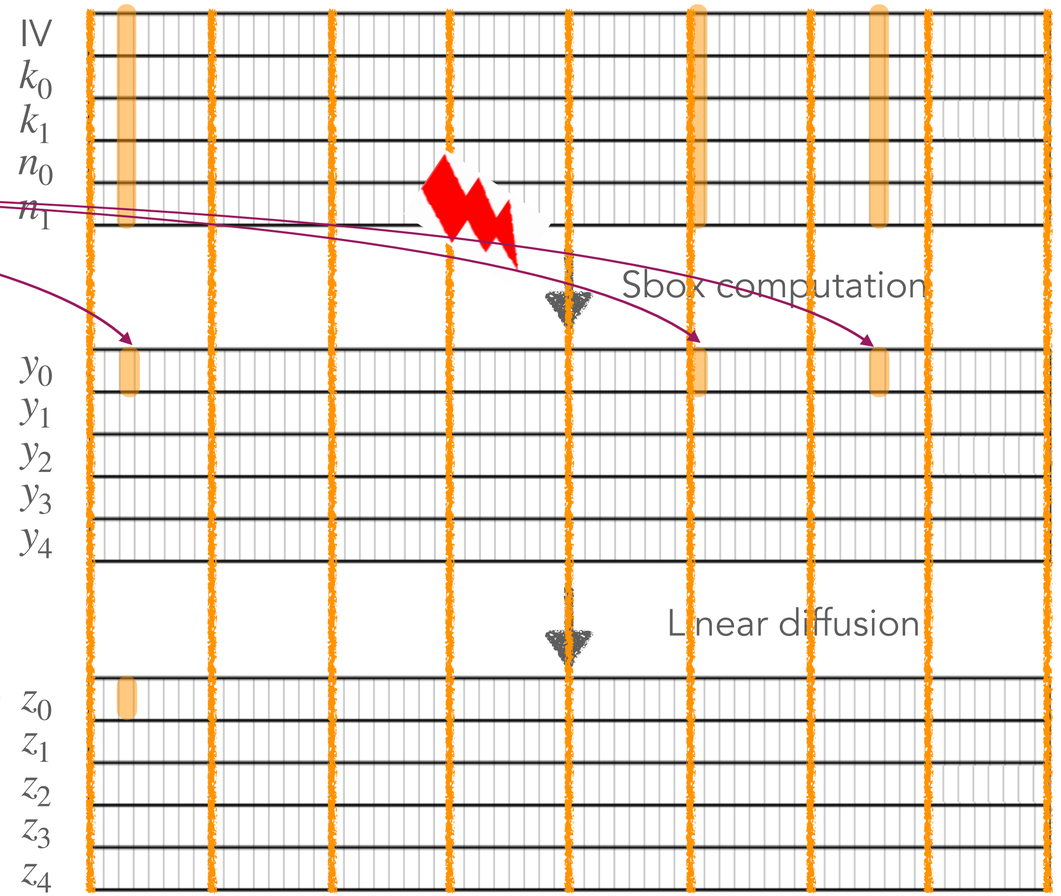
(following the attack strategy of [DMMP18])

on 8-bit architecture

An instruction skip
does <u>not affect all</u> 3 bits

→ z_0^j remains uniform

→ SIFA is not applicable

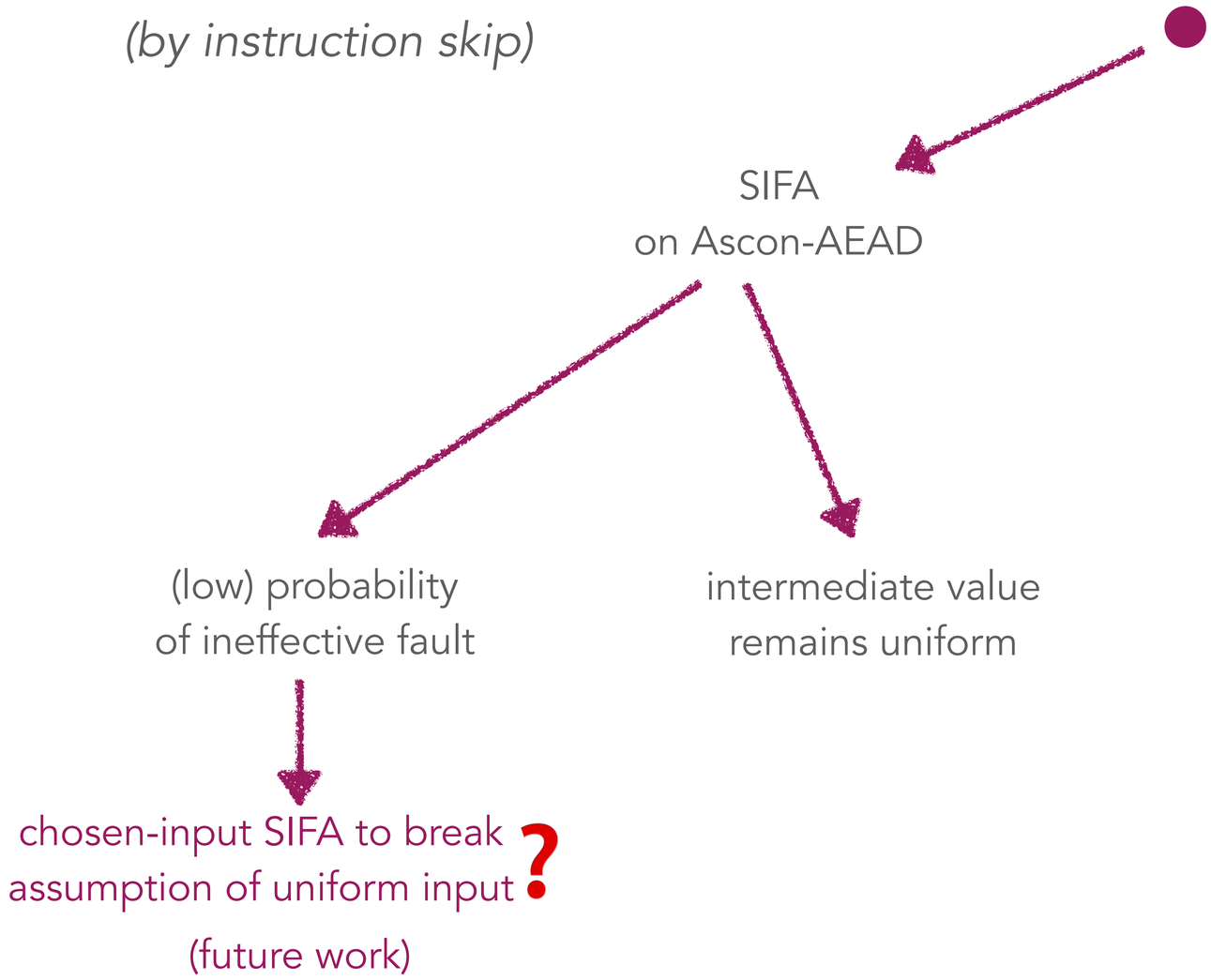


target bit z_0^j

$$z_0^j = y_0^j \oplus y_0^{j+19} \oplus y_0^{j+28}$$

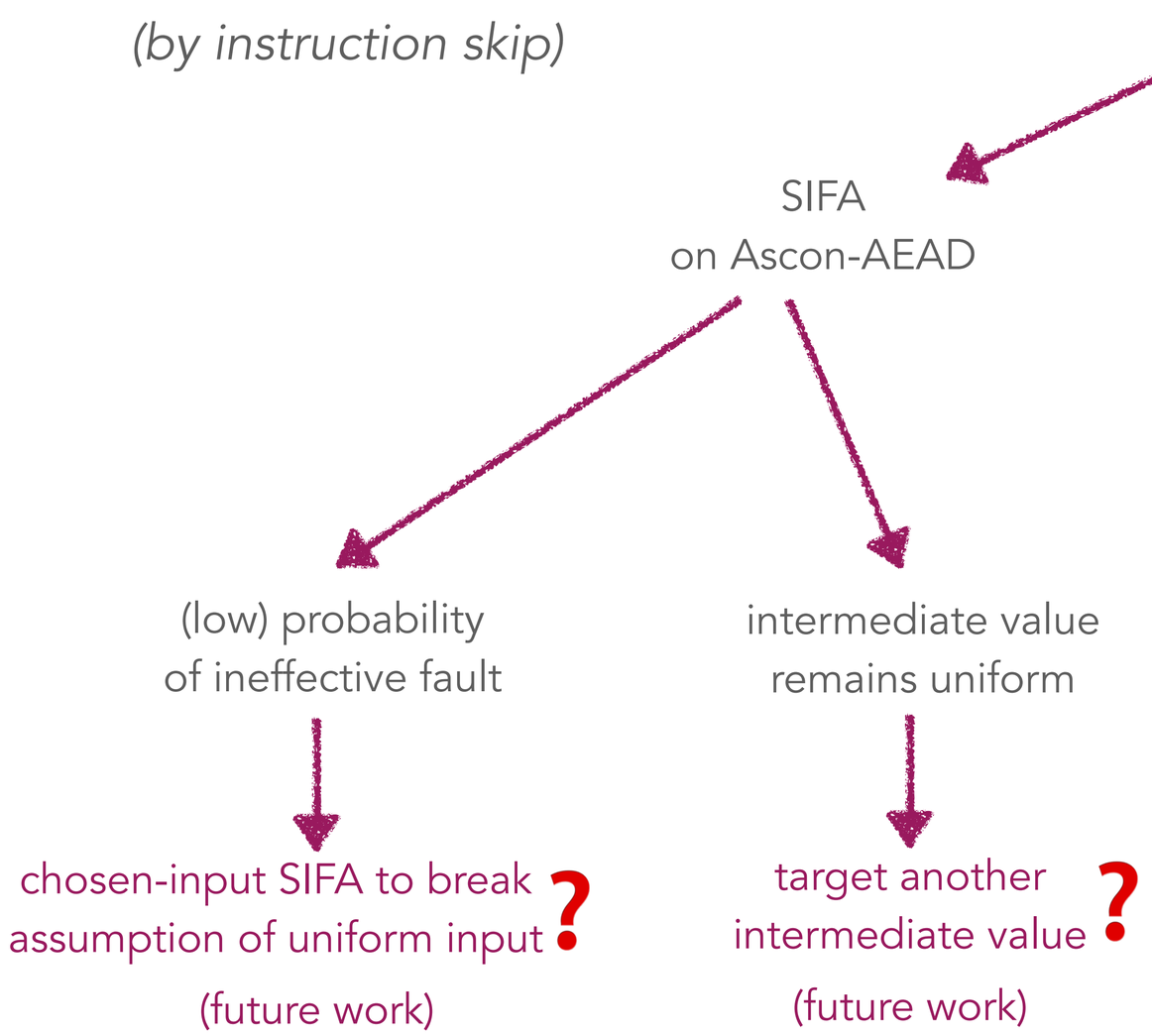
Map of fault attacks

(by instruction skip)



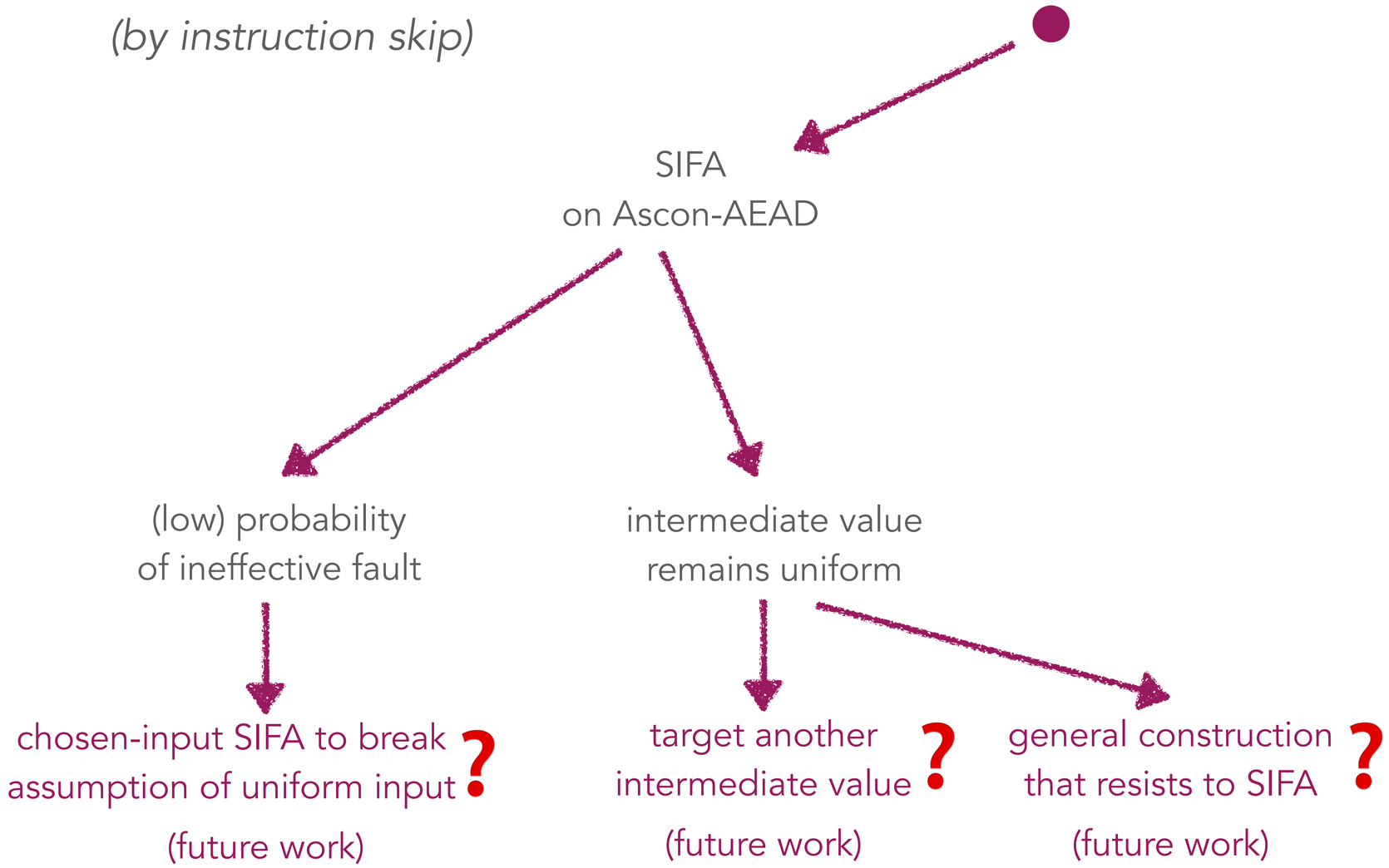
Map of fault attacks

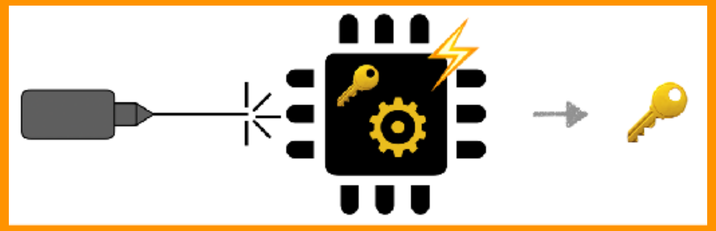
(by instruction skip)



Map of fault attacks

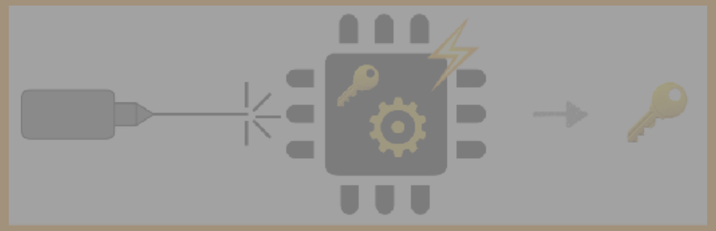
(by instruction skip)





Persistent Fault Attacks (PFA) on AES

👨🏻‍🎓 Contribution 4:



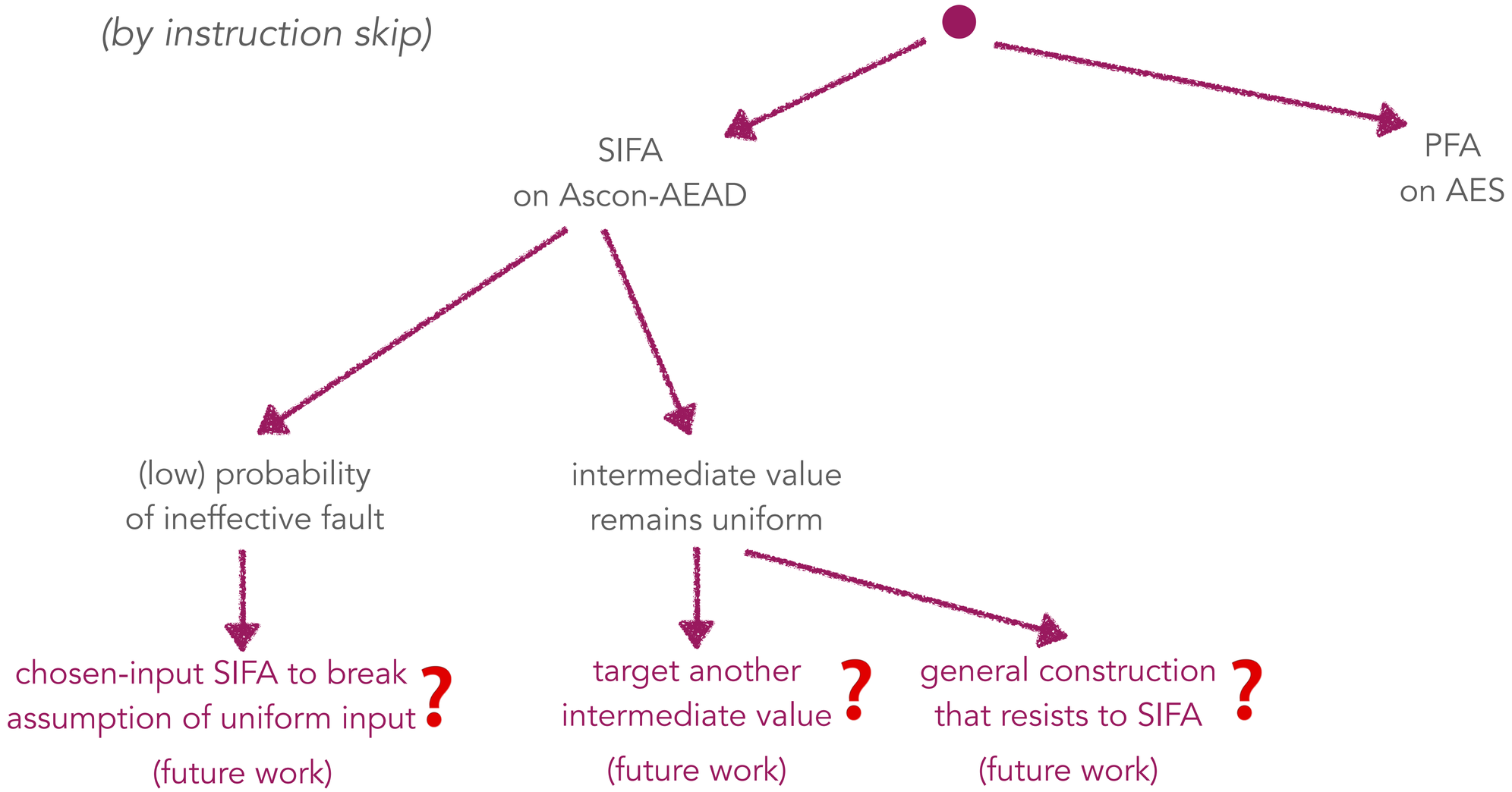
Persistent Fault Attacks (PFA) on AES

👨🏻‍🎓 Contribution 4: New approaches for attacks

Nguyen, Grosso, Cayrel - IACR CiC 2025

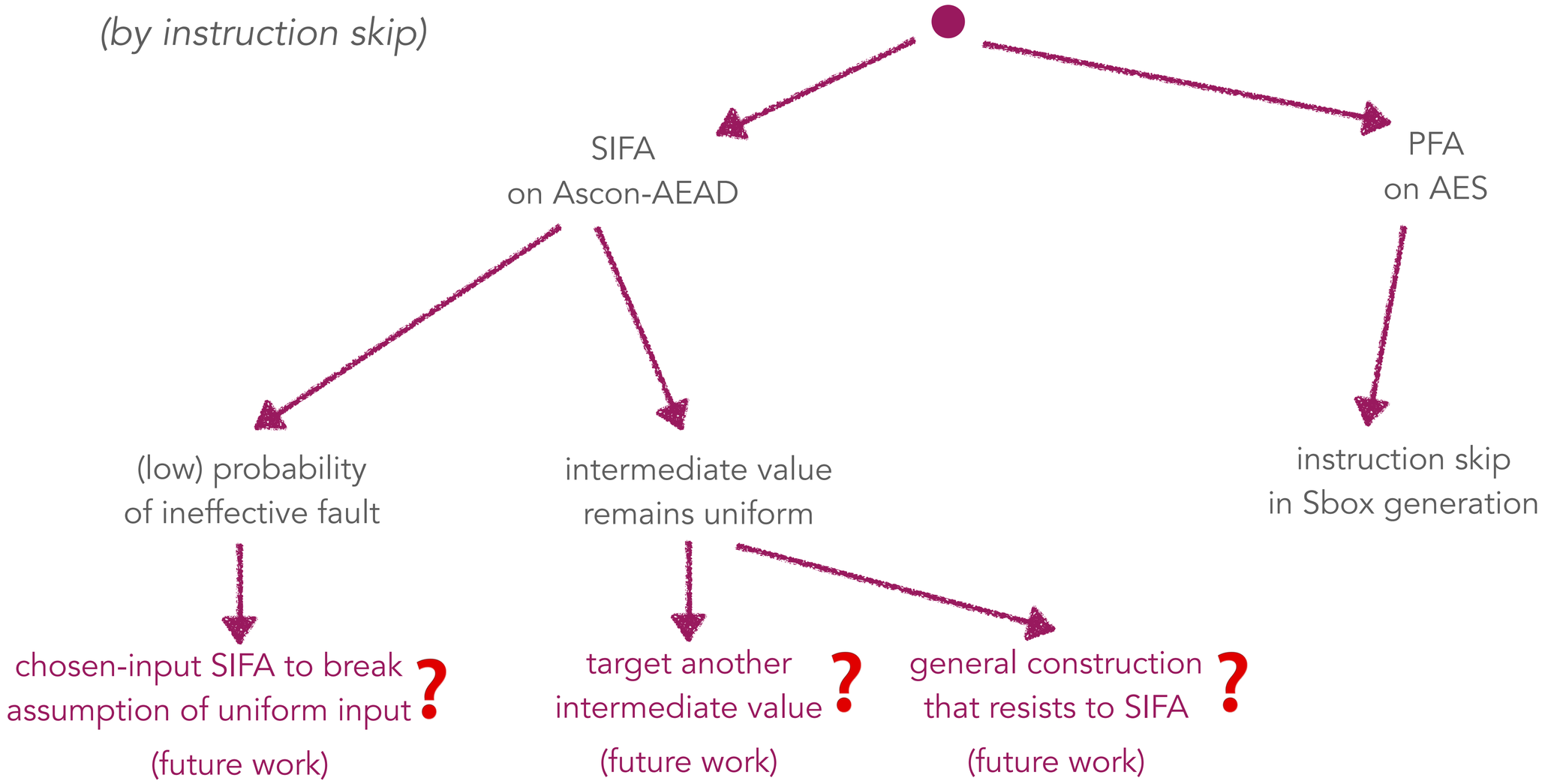
Map of fault attacks

(by instruction skip)



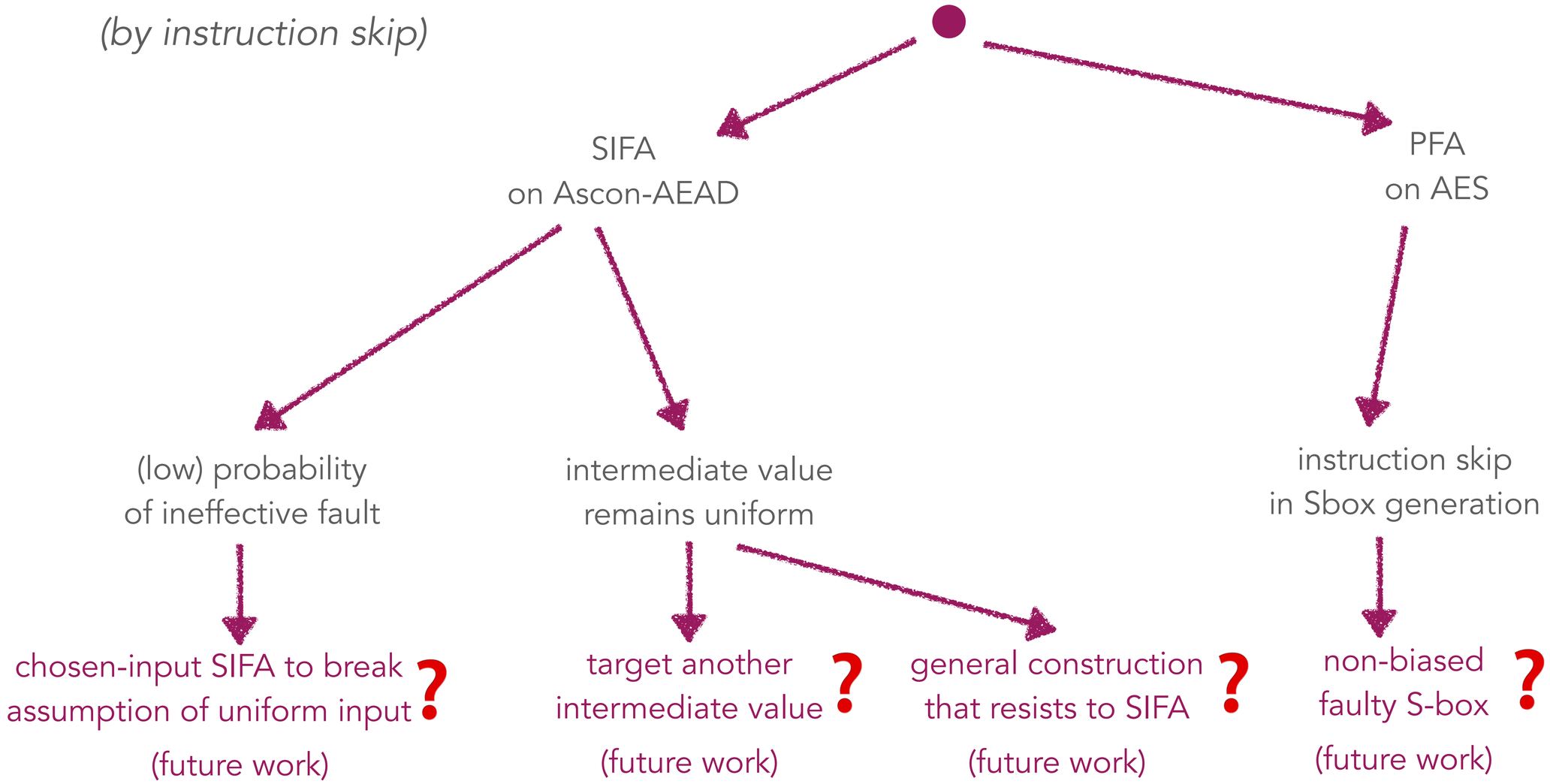
Map of fault attacks

(by instruction skip)



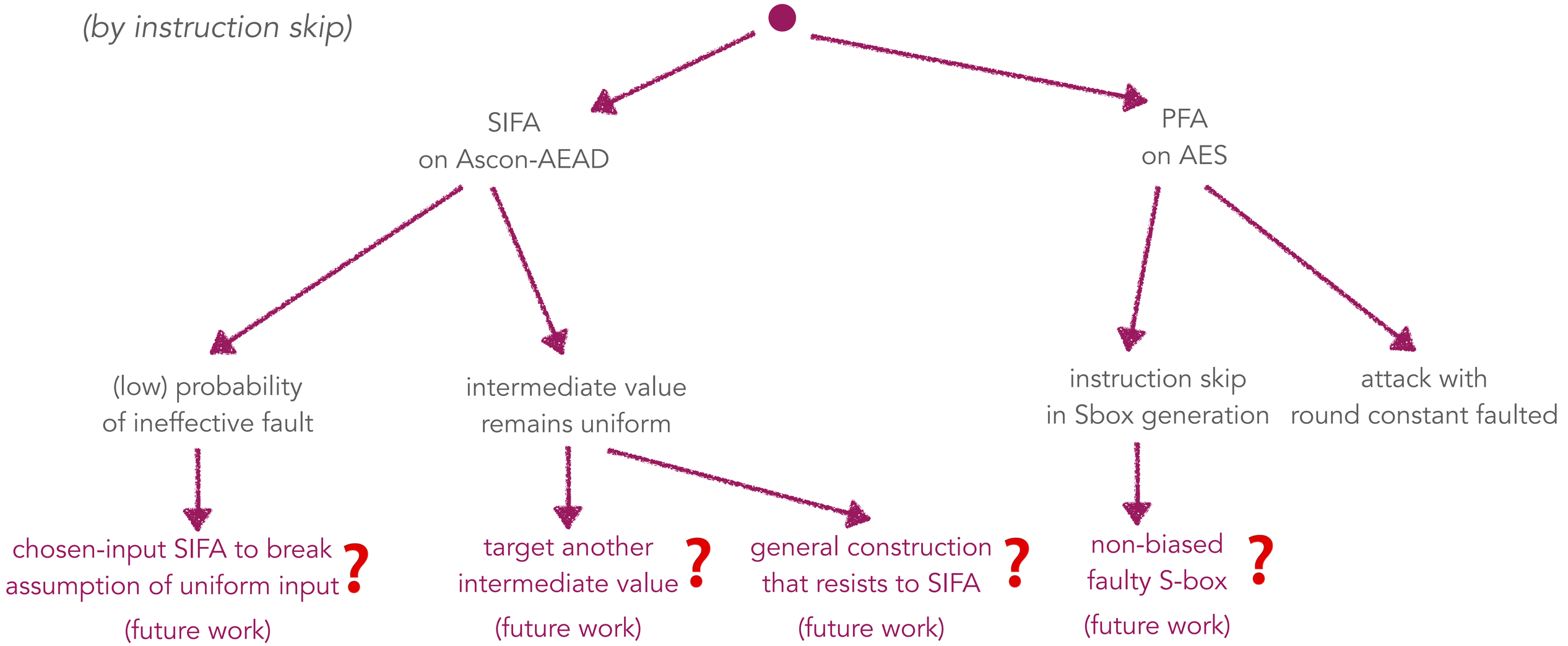
Map of fault attacks

(by instruction skip)



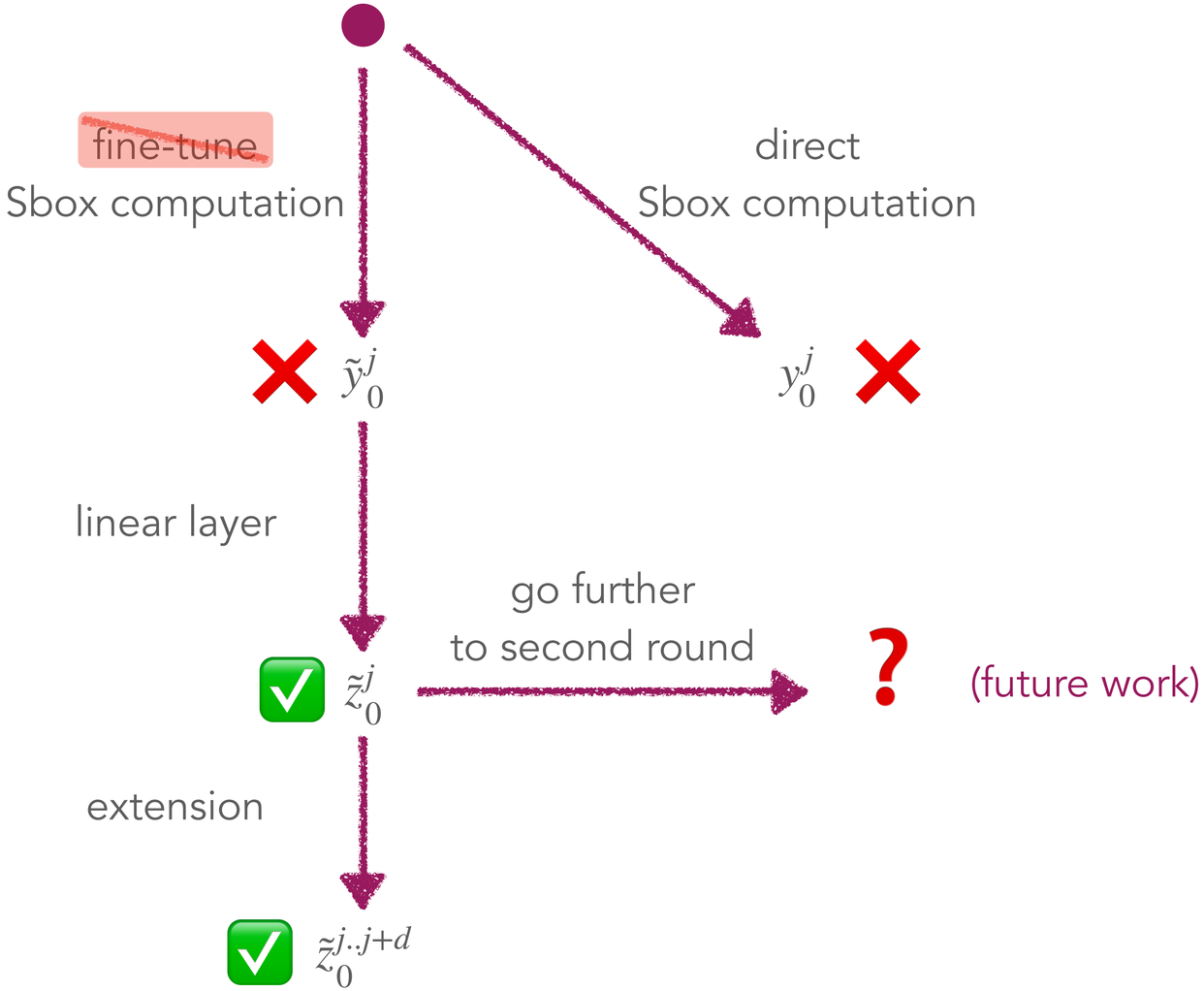
Map of fault attacks

(by instruction skip)



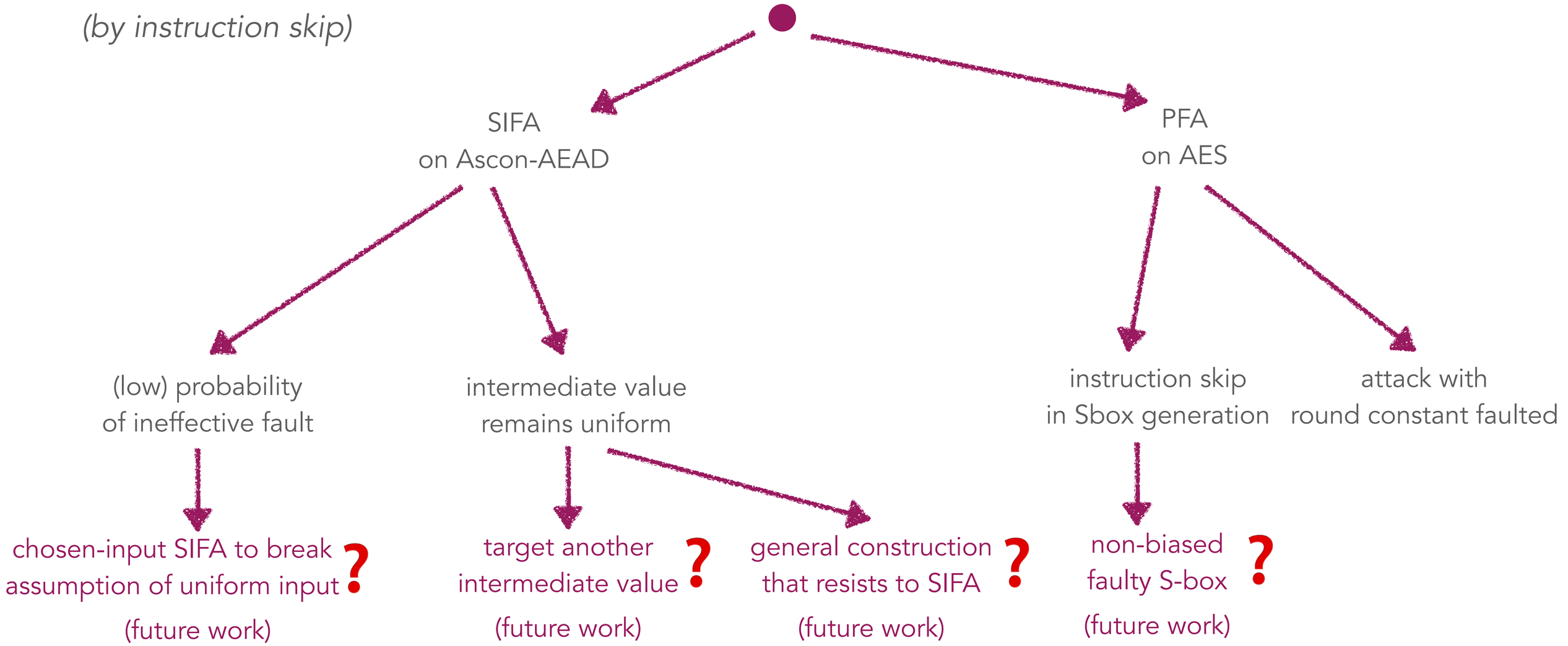
Conclusion

Map of selection functions



Map of fault attacks

(by instruction skip)



Thesis in numbers

Thesis in numbers

2 years **10** months

Thesis in numbers

2 years **10** months

4 publications, all as first author

Thesis in numbers

2 years **10** months

4 publications, all as first author

\+ experiments with open-source code

(https://github.com/nvietsang)

Thesis in numbers

2 years **10** months

4 publications, all as first author

\+ experiments with open-source code
(https://github.com/nvietsang)

2 unpublished papers

Thesis in numbers

2 years **10** months

4 publications, all as first author

\+ experiments with open-source code
(https://github.com/nvietsang)

2 unpublished papers

6 external presentations

Thesis in numbers

2 years **10** months

4 publications, all as first author

\+ experiments with open-source code
(https://github.com/nvietsang)

2 unpublished papers

6 external presentations

2 posters

Physical Attacks against Symmetric Cryptography Standards

Viet-Sang Nguyen

PhD defense

supervised by Vincent Grosso and Pierre-Louis Cayrel

Saint-Étienne, 19 December 2025

Université de Lyon
Université Jean Monnet Saint-Étienne
Laboratoire Hubert Curien
UMR • CNRS • 5516 • Saint-Étienne
anr
PROPHY ANR-22-CE39-0008-01