Differential and Correlation Power Analysis on Ascon

Viet Sang Nguyen

June 06, 2024

joint work with Vincent Grosso and Pierre-Louis Cayrel

PROPHY ANR-22-CE39-0008-01

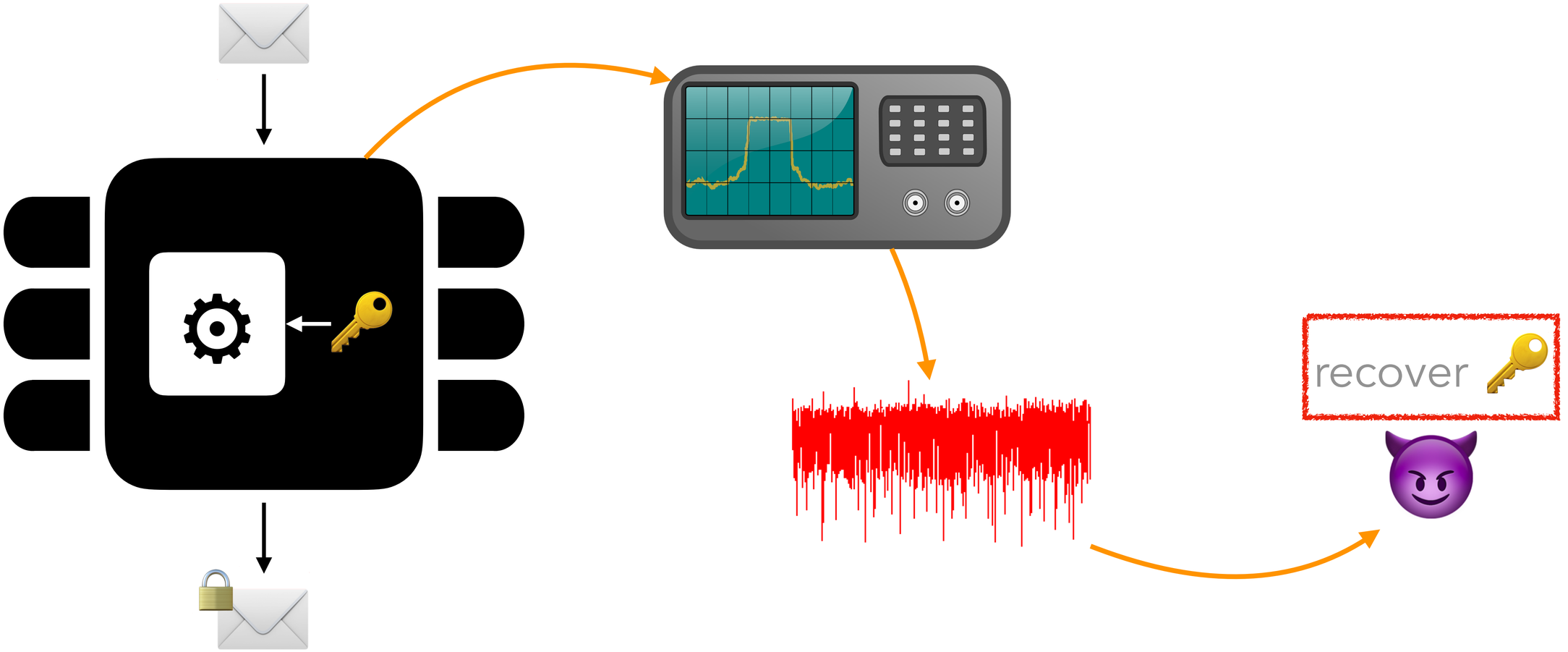
Outlines

1. Background
   - Differential and Correlation Power Analysis
   - Ascon
2. Previous Attacks
   - [SD17] and [RADKA20]
3. Our work
   - Comparison of Previous Attacks
   - Correlation Power Analysis with Less Traces
4. Conclusion

Outlines

1. Background
   - Differential and Correlation Power Analysis
   - Ascon
2. Previous Attacks
   - [SD17] and [RADKA20]
3. Our work
   - Comparison of Previous Attacks
   - Correlation Power Analysis with Less Traces
4. Conclusion

Side-Channel Attacks



Differential Power Analysis (DPA)

- Fact: register transitions:
  - 0 → 0*: consumes **not much** power
  - 0 → 1*: consumes **much** power

- So, based on this, how to recover key?
  - Choose a *selection function*: $f = \text{Sbox}(x \oplus k)$
    - x: plaintext → can be varied
    - k: key guess → to find correct key 🎯
  - Collect traces of power consumption
  - Perform analysis

Too theoretical?
How exactly? 🤔

*Assume that each register is pre-charged at 0

Example of DPA: correct key guess

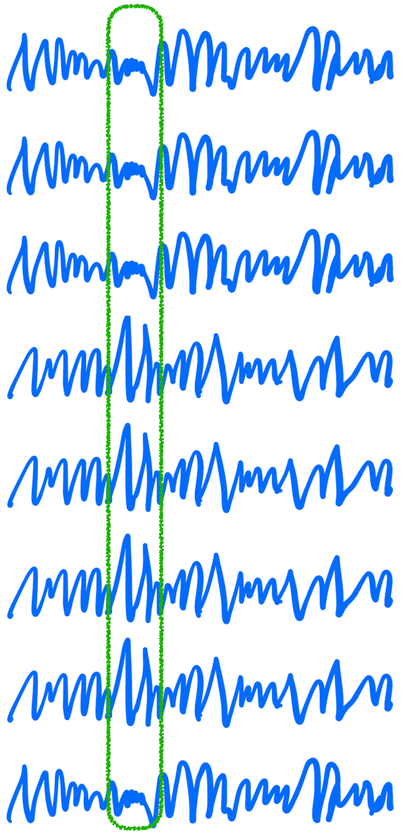
x	0	1	2	3	4	5	6	7
Sbox(x)	0	6	3	4	5	1	2	7

varied plaintext x	correct key $k^* = 1$	intermediate $f = \text{Sbox}(x \oplus k^*)$
000	001	110
001	001	000
010	001	100
011	001	011
100	001	001
101	001	101
110	001	111
111	001	010

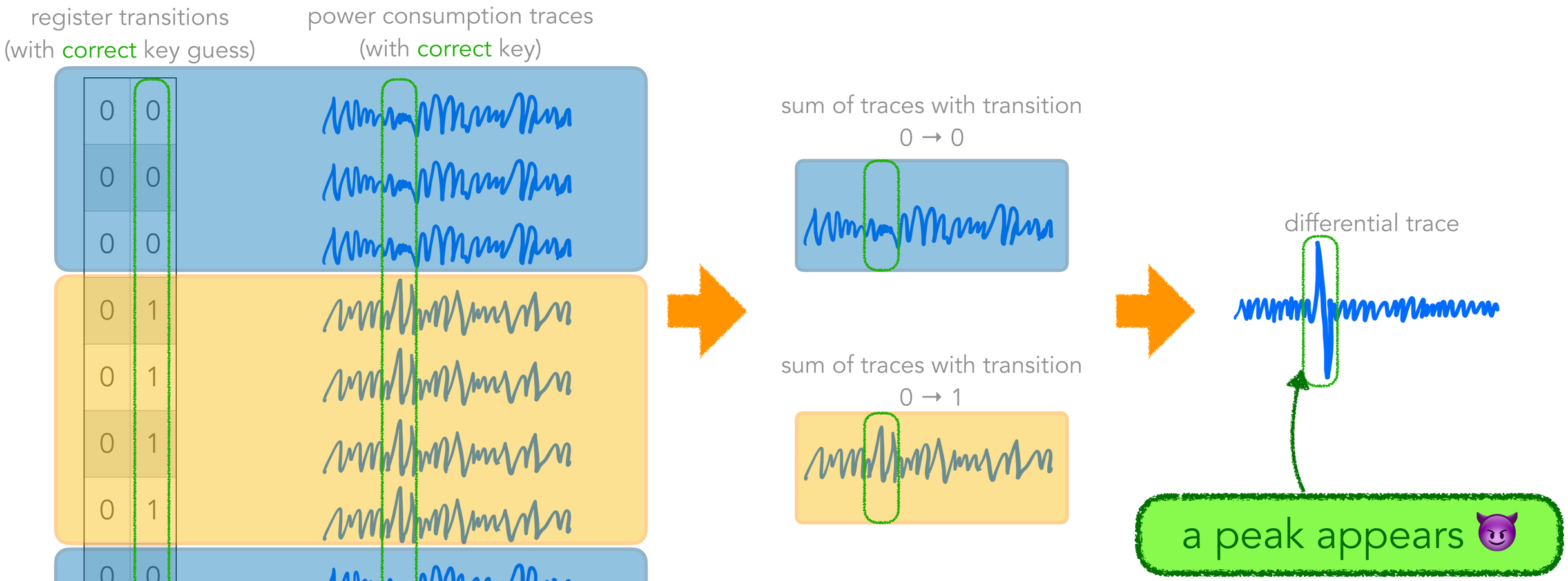
register transitions (with correct key guess)

0	0
0	0
0	0
0	1
0	1
0	1
0	1
0	0

power consumption traces (with correct key)



Example of DPA: correct key guess



Example of DPA: wrong key guess

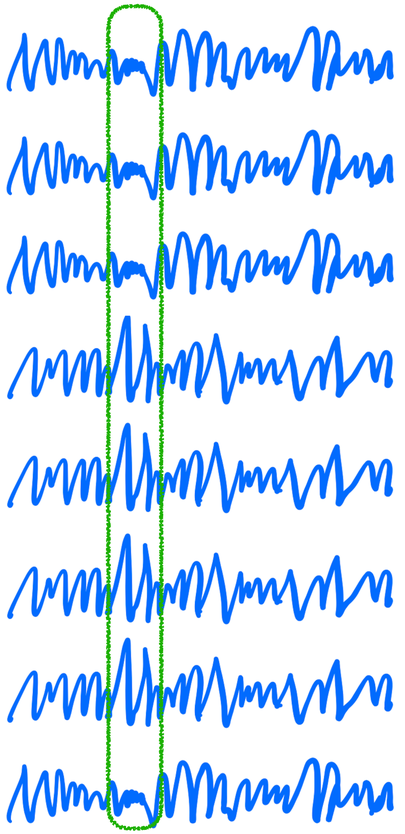
x	0	1	2	3	4	5	6	7
Sbox(x)	0	6	3	4	5	1	2	7

varied plaintext x	wrong key $k = 0$	intermediate $f = \text{Sbox}(x \oplus k)$
000	000	000
001	000	110
010	000	011
011	000	100
100	000	101
101	000	001
110	000	010
111	000	111

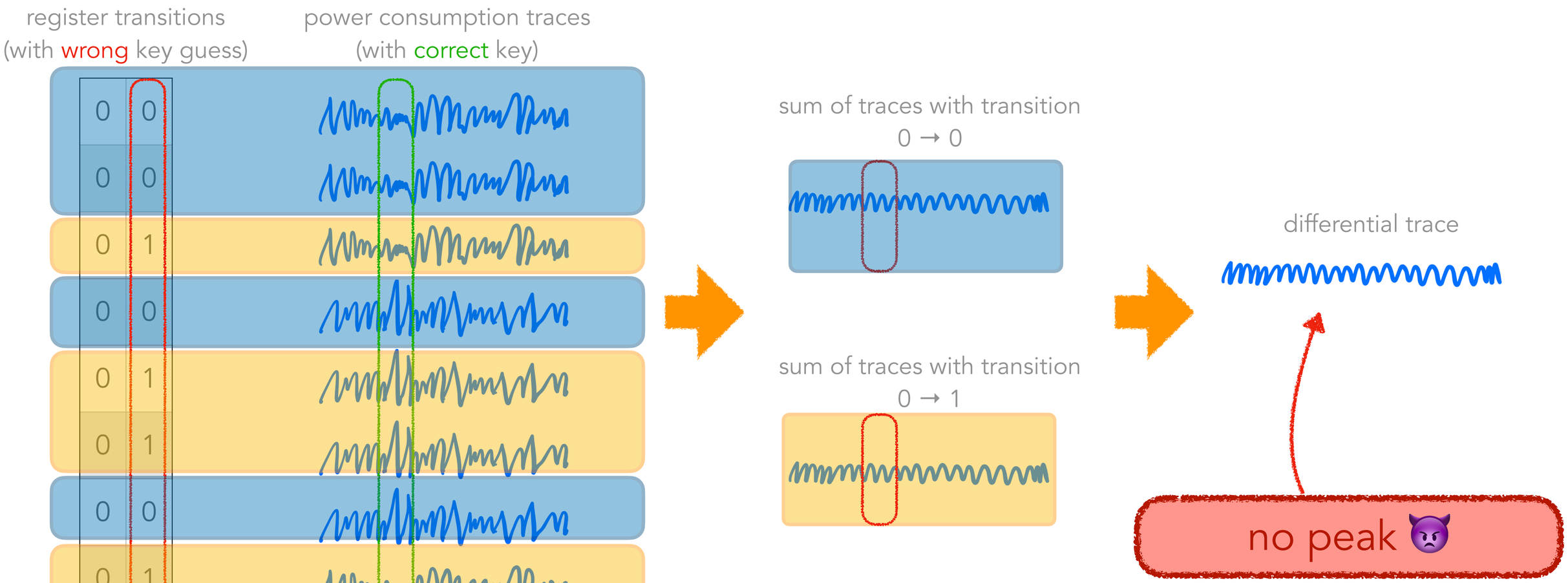
register transitions (with wrong key guess)

0	0	0
0	0	0
0	1	0
0	0	1
0	1	1
0	1	1
0	0	1
0	1	0

power consumption traces (with correct key)



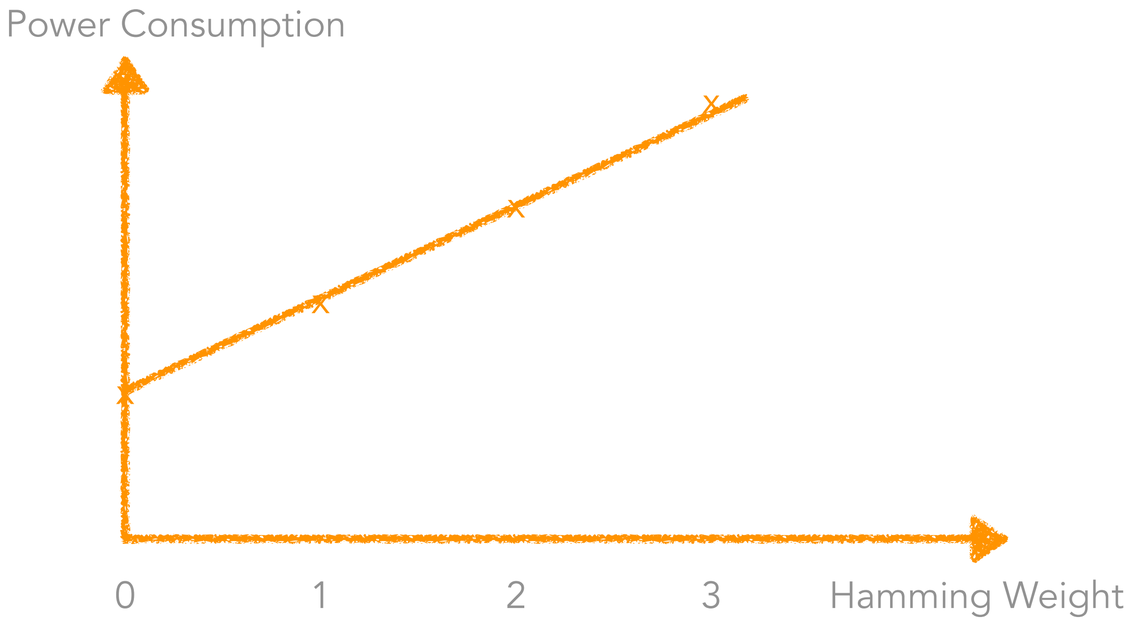
Example of DPA: wrong key guess



Correlation Power Analysis (CPA)

- Fact: Hamming Weight (HW) and power consumption have linear relation

- Register transitions
  - HW = 3: consumes *most* power
    - 000 → 111*
  - HW = 2: consumes *much* power
    - 000 → 011*, 000 → 110*, etc.
  - HW = 1: consumes *less* power
    - 000 → 001*, 000 → 010*, etc.
  - HW = 0: consumes *least* power
    - 000 → 000*



*Assume that each register is pre-charged at 0

Example of CPA: correct key guess

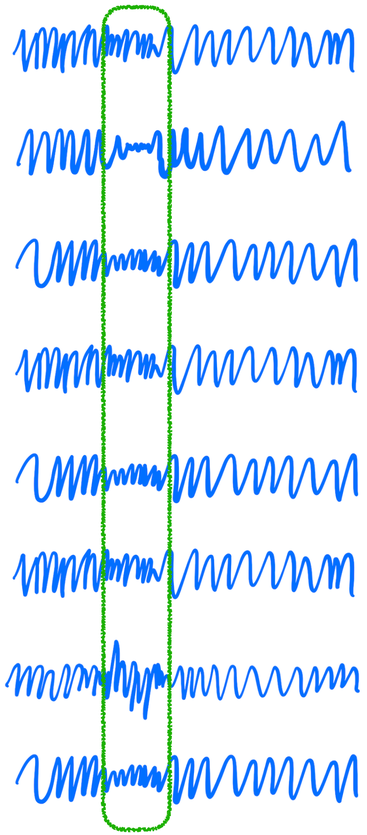
x	0	1	2	3	4	5	6	7
Sbox(x)	0	6	3	4	5	1	2	7

varied plaintext x	correct key $k^* = 1$	intermediate $f = \text{Sbox}(x \oplus k^*)$
000	001	110
001	001	000
010	001	100
011	001	011
100	001	001
101	001	101
110	001	111
111	001	010

register transitions (with correct key guess)

000	110	2
000	000	0
000	100	1
000	011	2
000	001	1
000	101	2
000	111	3
000	010	1

power consumption traces (with correct key)

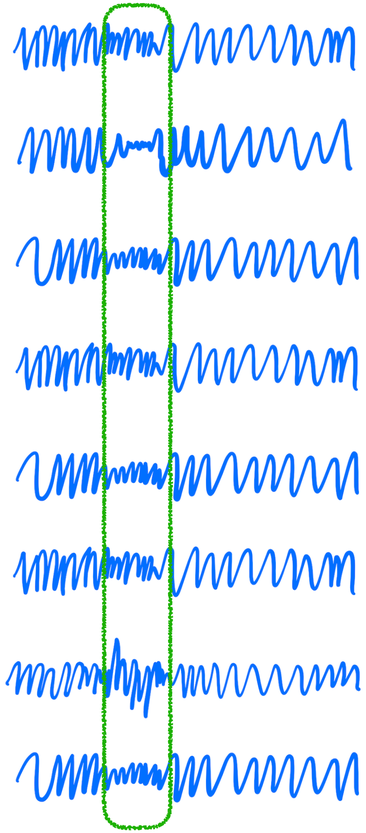


Example of CPA: correct key guess

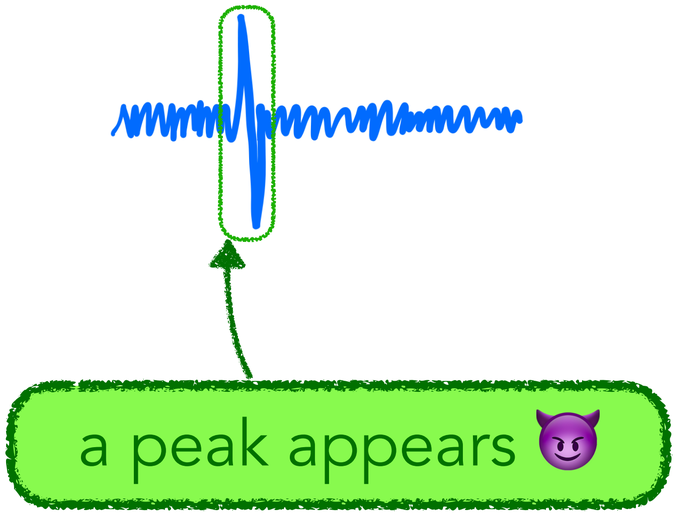
register transitions
(with correct key guess)

000	110	2
000	000	0
000	100	1
000	011	2
000	001	1
000	101	2
000	111	3
000	010	1

power consumption traces
(with correct key)



Pearson's correlation trace



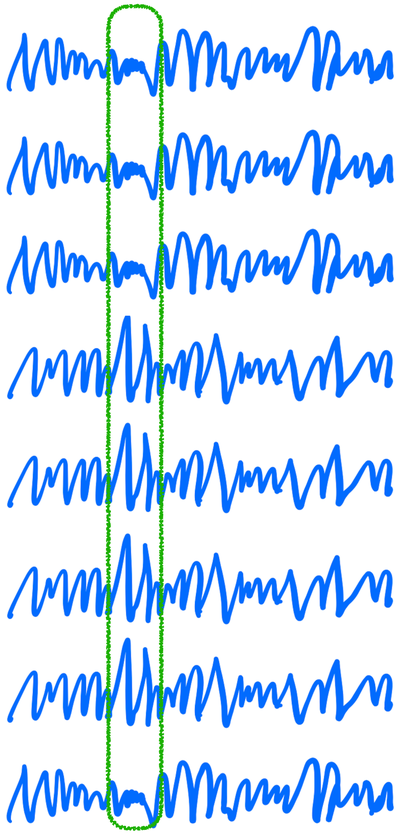
Example of CPA: wrong key guess

x	0	1	2	3	4	5	6	7
Sbox(x)	0	6	3	4	5	1	2	7

varied plaintext x	wrong key $k = 0$	intermediate $f = \text{Sbox}(x \oplus k^*)$
000	000	000
001	000	110
010	000	011
011	000	100
100	000	101
101	000	001
110	000	010
111	000	111

register transitions (with wrong key guess)			
000	000	0	2
000	110	2	0
000	011	2	1
000	100	1	2
000	101	2	1
000	001	1	2
000	010	1	3
000	111	3	1

power consumption traces (with correct key)

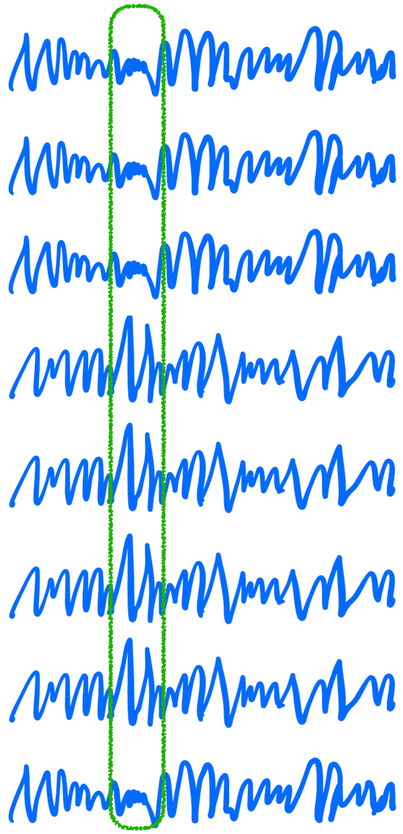


Example of CPA: wrong key guess

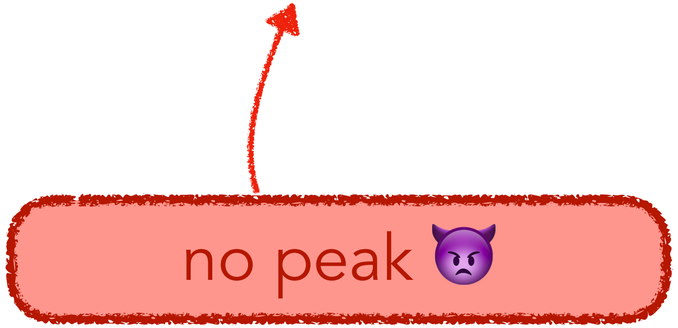
register transitions
(with wrong key guess)

000	000	0	2
000	110	2	0
000	011	2	1
000	100	1	2
000	101	2	1
000	001	1	2
000	010	1	3
000	111	3	1

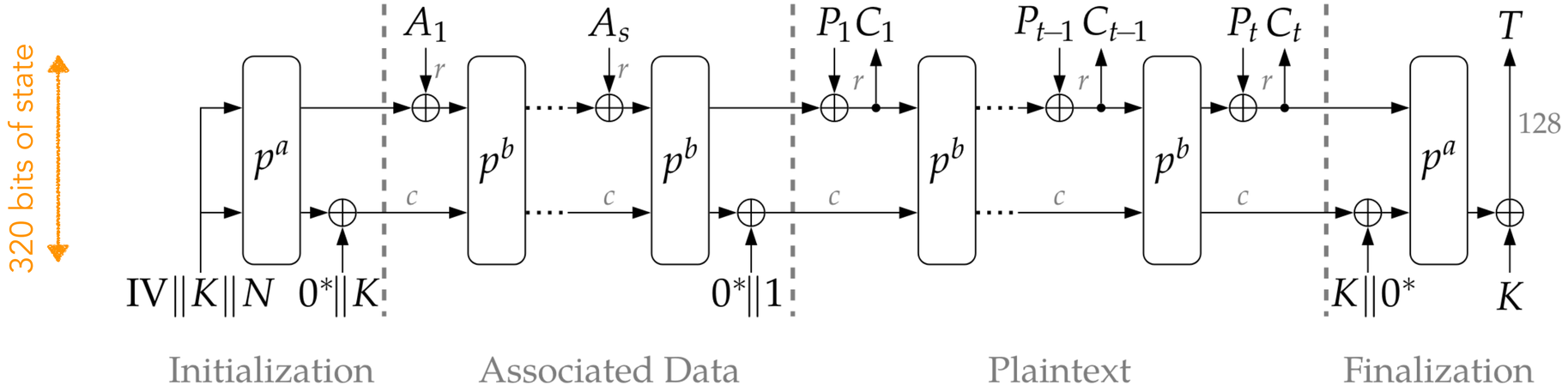
power consumption traces
(with correct key)



Pearson's correlation trace

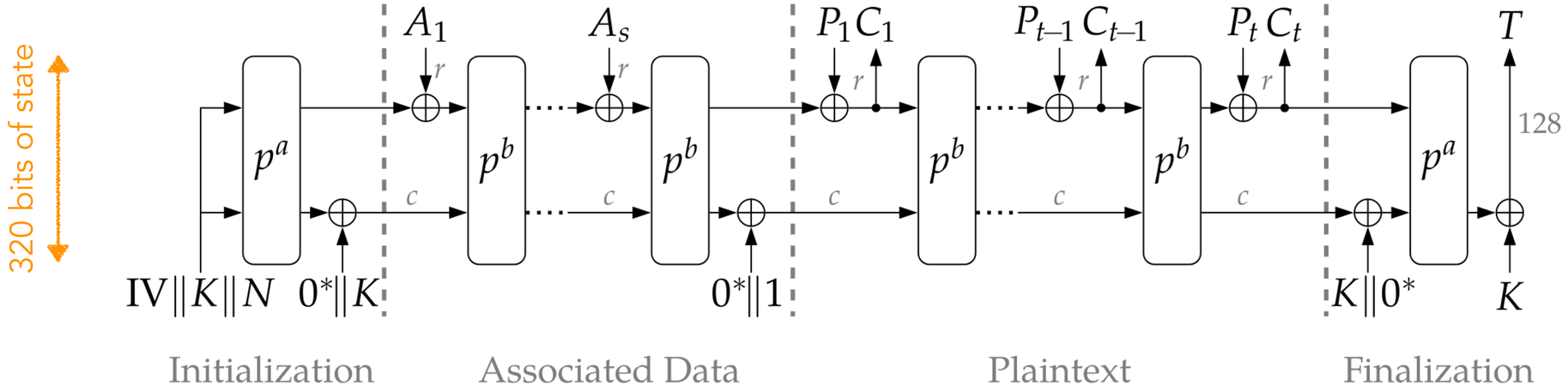


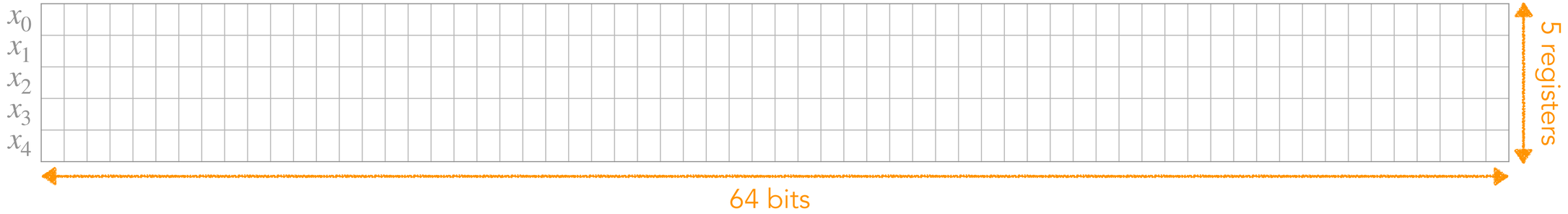
Ascon Cipher



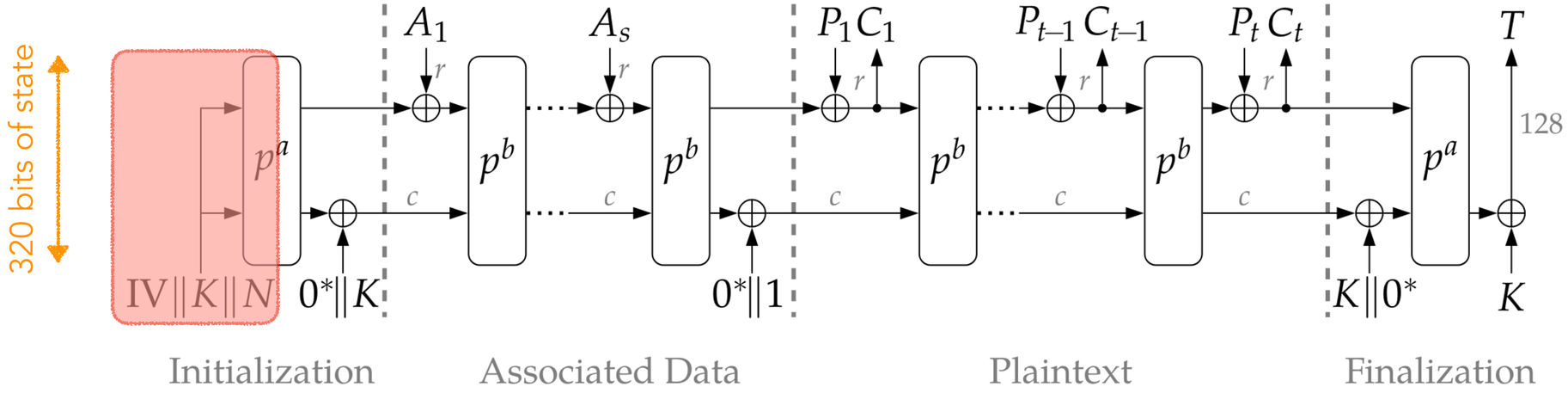
- Version 128:
  - IV: 64 bits of constant
  - K: 128 bits of key
  - N: 128 bits of nonce
  - p^a: 12 permutation rounds
  - p^b: 6 permutation rounds
  - r: 64 bits of rate
  - c: 256 bits of capacity

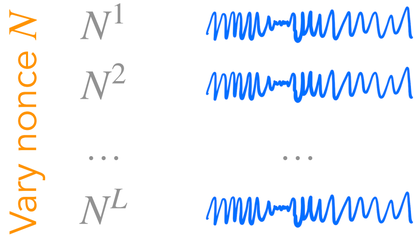
Ascon Cipher: 320-bit state

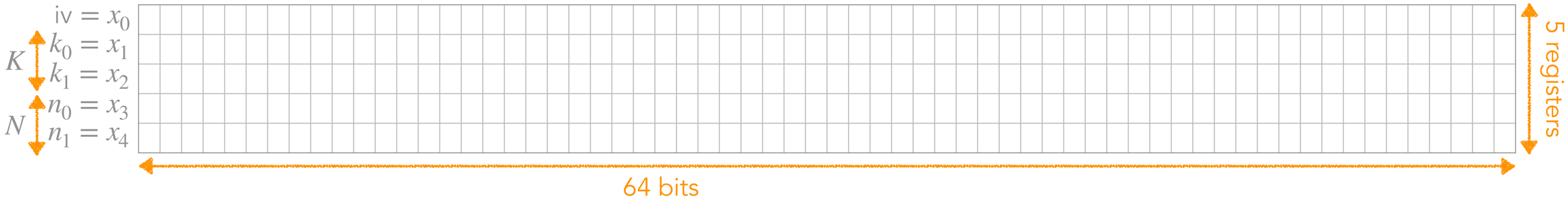




Target the very first round for attacks

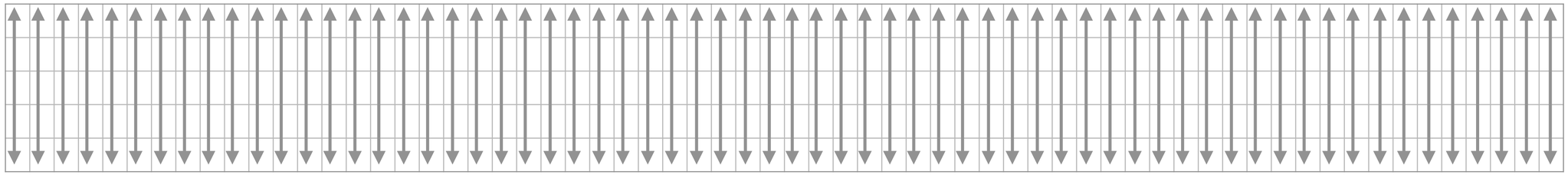




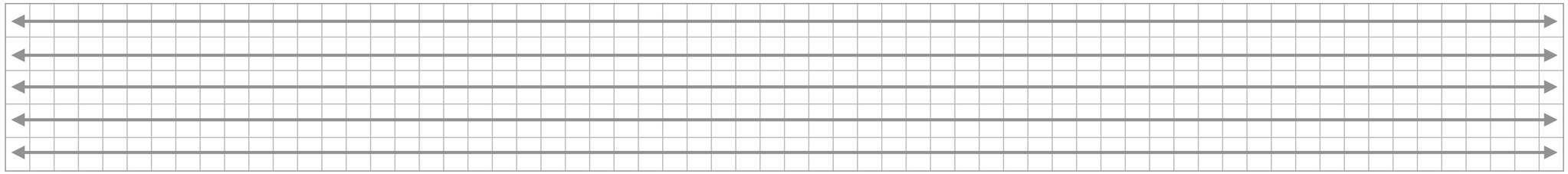


A Permutation Round

(1) Round constant addition *(ignore for now)*

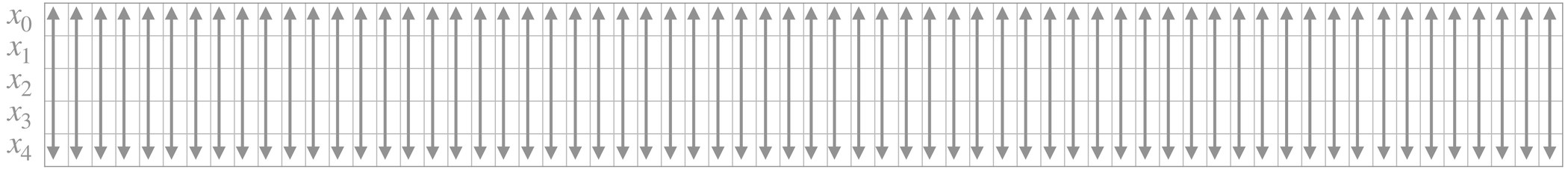


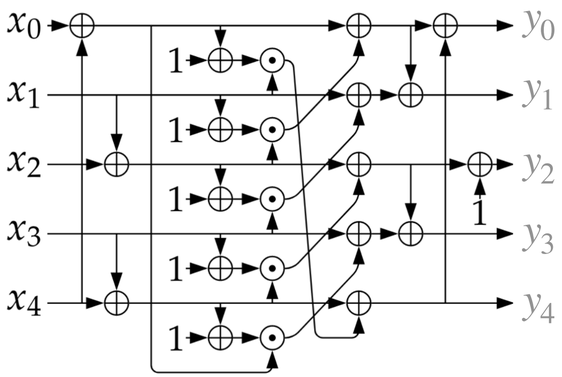
(2) Substitution layer with 5-bit Sbox



(3) Linear layer with 64-bit diffusion function

(2) Sbox layer





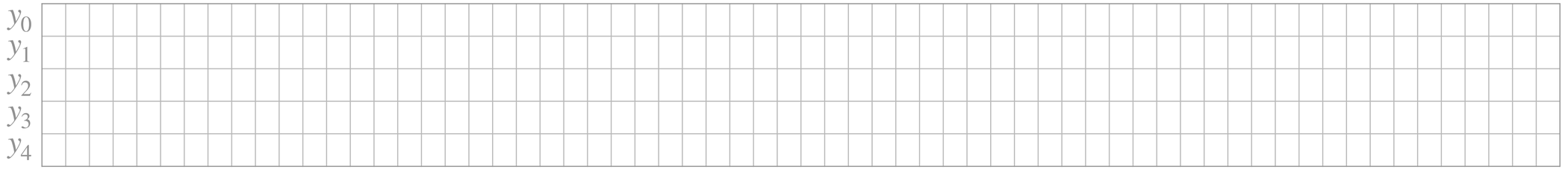
$$y_0 = x_4x_1 \oplus x_3 \oplus x_2x_1 \oplus x_2 \oplus x_1x_0 \oplus x_1 \oplus x_0$$

$$y_1 = x_4 \oplus x_3x_2 \oplus x_3x_1 \oplus x_3 \oplus x_2x_1 \oplus x_2 \oplus x_1 \oplus x_0$$

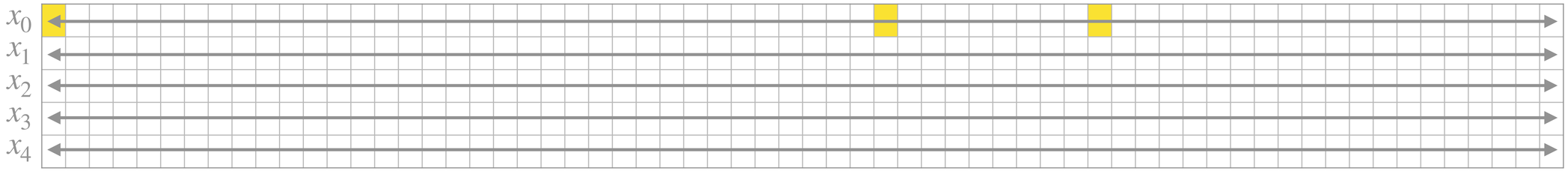
$$y_2 = x_4x_3 \oplus x_4 \oplus x_2 \oplus x_1 \oplus 1$$

$$y_3 = x_4x_0 \oplus x_4 \oplus x_3x_0 \oplus x_3 \oplus x_2 \oplus x_1 \oplus x_0$$

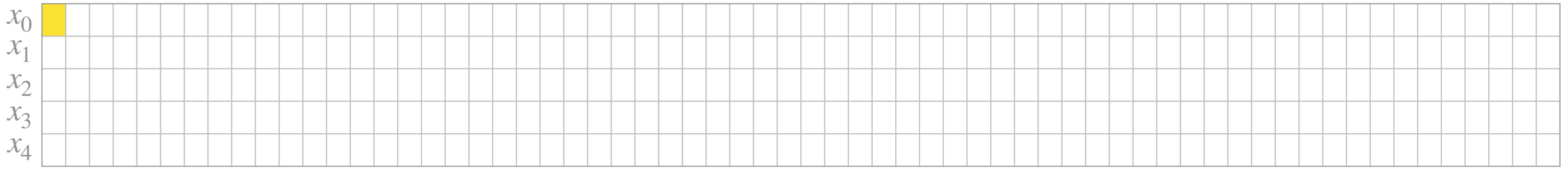
$$y_4 = x_4x_1 \oplus x_4 \oplus x_3 \oplus x_1x_0 \oplus x_1$$



(3) Linear layer



$$x_0 \leftarrow \Sigma_0(x_0) = x_0 \oplus (x_0 \ggg 19) \oplus (x_0 \ggg 28)$$
$$x_1 \leftarrow \Sigma_1(x_1) = x_1 \oplus (x_1 \ggg 61) \oplus (x_1 \ggg 39)$$
$$x_2 \leftarrow \Sigma_2(x_2) = x_2 \oplus (x_2 \ggg 1) \oplus (x_2 \ggg 6)$$
$$x_3 \leftarrow \Sigma_3(x_3) = x_3 \oplus (x_3 \ggg 10) \oplus (x_3 \ggg 17)$$
$$x_4 \leftarrow \Sigma_4(x_4) = x_4 \oplus (x_4 \ggg 7) \oplus (x_4 \ggg 41)$$



Outlines

1. Background
   - Differential and Correlation Power Analysis
   - Ascon
2. Previous Attacks
   - [SD17] and [RADKA20]
3. Our work
   - Comparison of Previous Attacks
   - Correlation Power Analysis with Less Traces
4. Conclusion

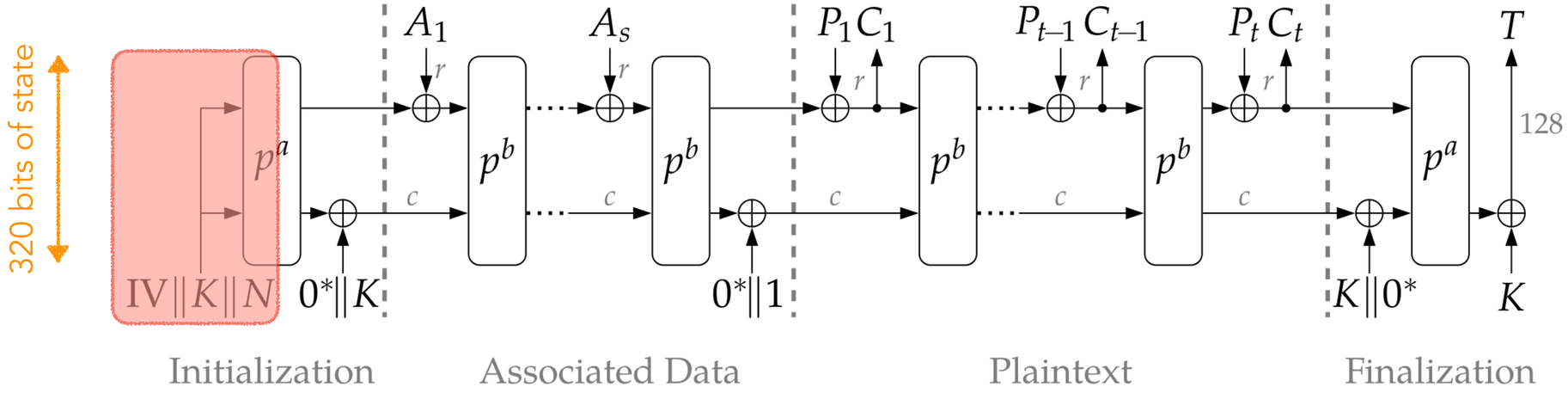
Remind: Selection Function

$$f = \text{Sbox}(x \oplus k)$$

we can vary
to collect different traces
(under correct key k^*)

we make guess
to find correct key k^*

Target the very first round

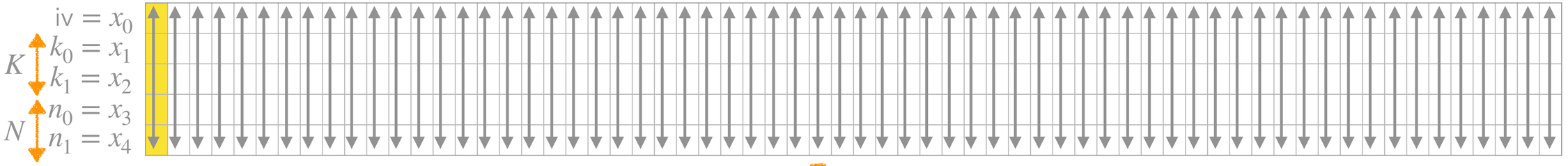


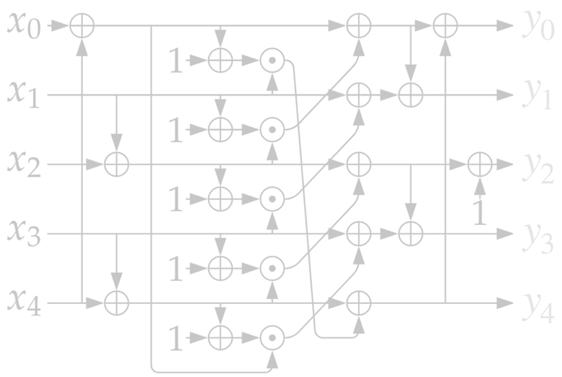
$f = \mathrm{Sbox}(x \oplus k)$ now is $f = \varphi(n, k)$

nonce (to be varied)

key (to be guessed)

Target the very first round





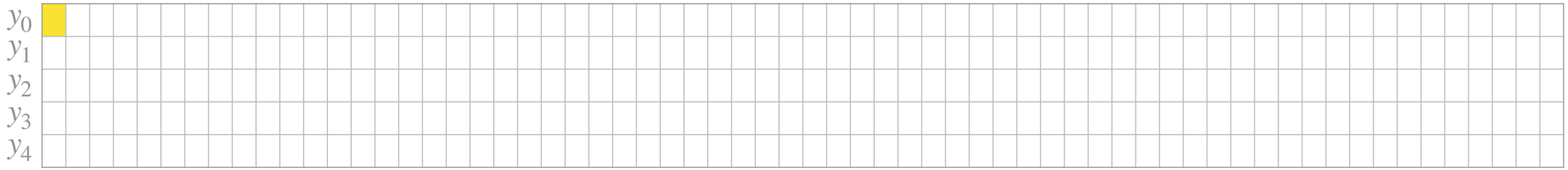
$$y_0 = x_4x_1 \oplus x_3 \oplus x_2x_1 \oplus x_2 \oplus x_1x_0 \oplus x_1 \oplus x_0$$

$$y_1 = x_4 \oplus x_3x_2 \oplus x_3x_1 \oplus x_3 \oplus x_2x_1 \oplus x_2 \oplus x_1 \oplus x_0$$

$$y_2 = x_4x_3 \oplus x_4 \oplus x_2 \oplus x_1 \oplus 1$$

$$y_3 = x_4x_0 \oplus x_4 \oplus x_3x_0 \oplus x_3 \oplus x_2 \oplus x_1 \oplus x_0$$

$$y_4 = x_4x_1 \oplus x_4 \oplus x_3 \oplus x_1x_0 \oplus x_1$$



Attack of Ramezanpour et al. [RADKA20]

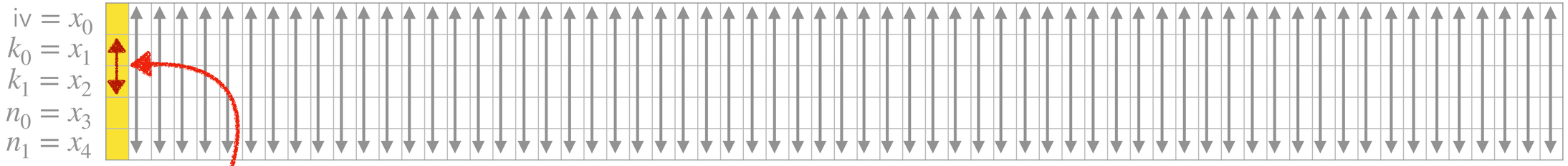
Active and Passive Side-Channel Key Recovery Attacks on Ascon

Keyvan Ramezanpour[1], Abubakr Abdulgadir[2], William Diehl[1], Jens-Peter Kaps[2], and Paul Ampadu[1]

[1] Virginia Tech, Blacksburg, VA 24061, USA{rkeyvan8,wdiehl,ampadu}@vt.edu
[2] George Mason University, Fairfax, VA 22033, USA {aabdulga,jkaps}@gmu.edu

- Cannot recover key with 40K traces! 😈

Selection function of [RAKDA20]



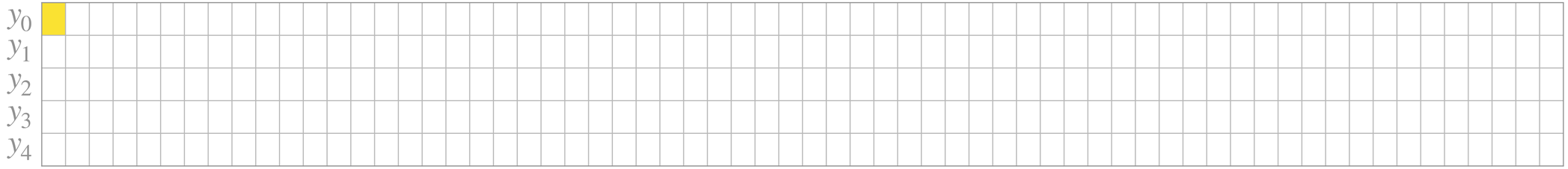
😈 guess 2 bits

$y_0 = x_4x_1 \oplus x_3 \oplus x_2x_1 \oplus x_2 \oplus x_1x_0 \oplus x_1 \oplus x_0$

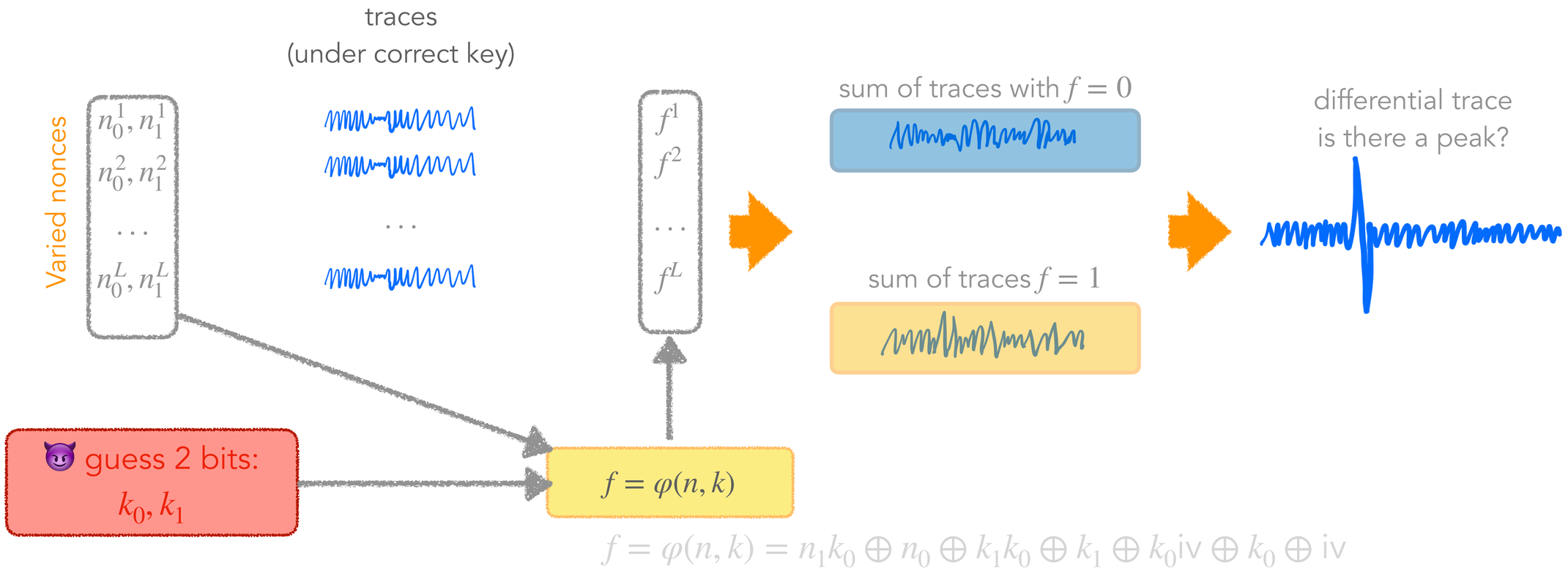
$y_0 = n_1k_0 \oplus n_0 \oplus k_1k_0 \oplus k_1 \oplus k_0\text{iv} \oplus k_0 \oplus \text{iv}$

[RAKDA20]: Choose this as selection function:

$$f = \varphi(n, k) = n_1k_0 \oplus n_0 \oplus k_1k_0 \oplus k_1 \oplus k_0\text{iv} \oplus k_0 \oplus \text{iv}$$



DPA of [RAKDA20]



Cannot recover key with 40K traces! 😈 **WHY?** 🤔

Attack of Samwel and Daemen [SD17]

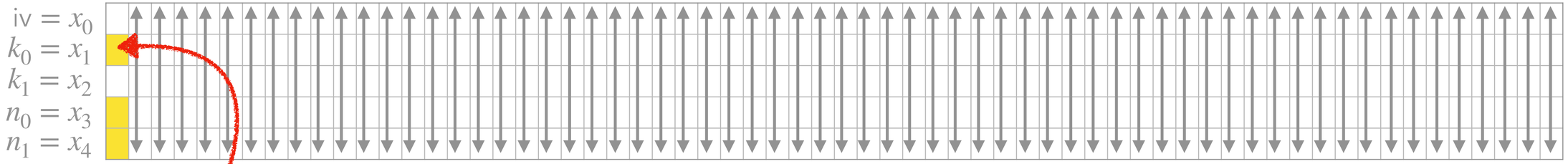
DPA on hardware implementations of Ascon and Keyak

Niels Samwel[1] and Joan Daemen[1,2]

[1] Digital Security Group, Radboud University Nijmegen
{n.samwel, joan}@cs.ru.nl
[2] ST Microelectronics

- Recover key successfully with ~1.5K traces! 😈

Selection function of [SD17]



😈 guess 1 bits

$y_0 = x_4x_1 \oplus x_3 \oplus x_2x_1 \oplus x_2 \oplus x_1x_0 \oplus x_1 \oplus x_0$

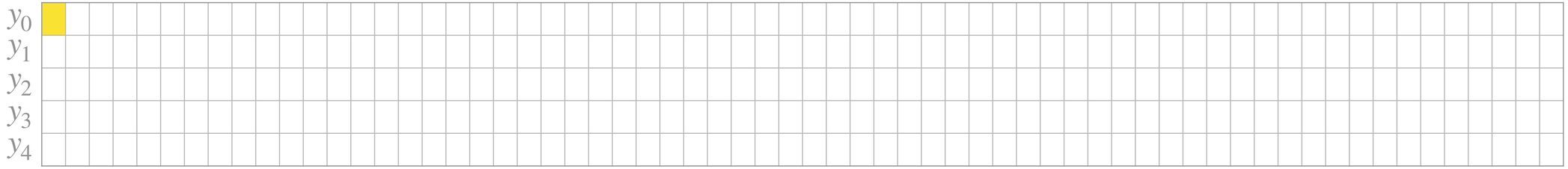
$y_0 = n_1k_0 \oplus n_0 \oplus k_1k_0 \oplus k_1 \oplus k_0\text{iv} \oplus k_0 \oplus \text{iv}$

$y_0 \sim k_0(n_1 \oplus 1) \oplus n_0$

[SD17]: Choose this as selection function:

$$f = \varphi(n, k) = k_0(n_1 \oplus 1) + n_0$$

Wait… How? 🤔



Selection function of [SD17]

$$y_0 = x_4x_1 \oplus x_3 \oplus x_2x_1 \oplus x_2 \oplus x_1x_0 \oplus x_1 \oplus x_0$$

$$y_0 = n_1k_0 \oplus n_0 \oplus k_1k_0 \oplus k_1 \oplus k_0\mathsf{iv} \oplus k_0 \oplus \mathsf{iv}$$

$$y_0 = k_0(n_1 \oplus 1) \oplus n_0 \oplus k_1k_0 \oplus k_0\mathsf{iv} \oplus k_1 \oplus \mathsf{iv}$$

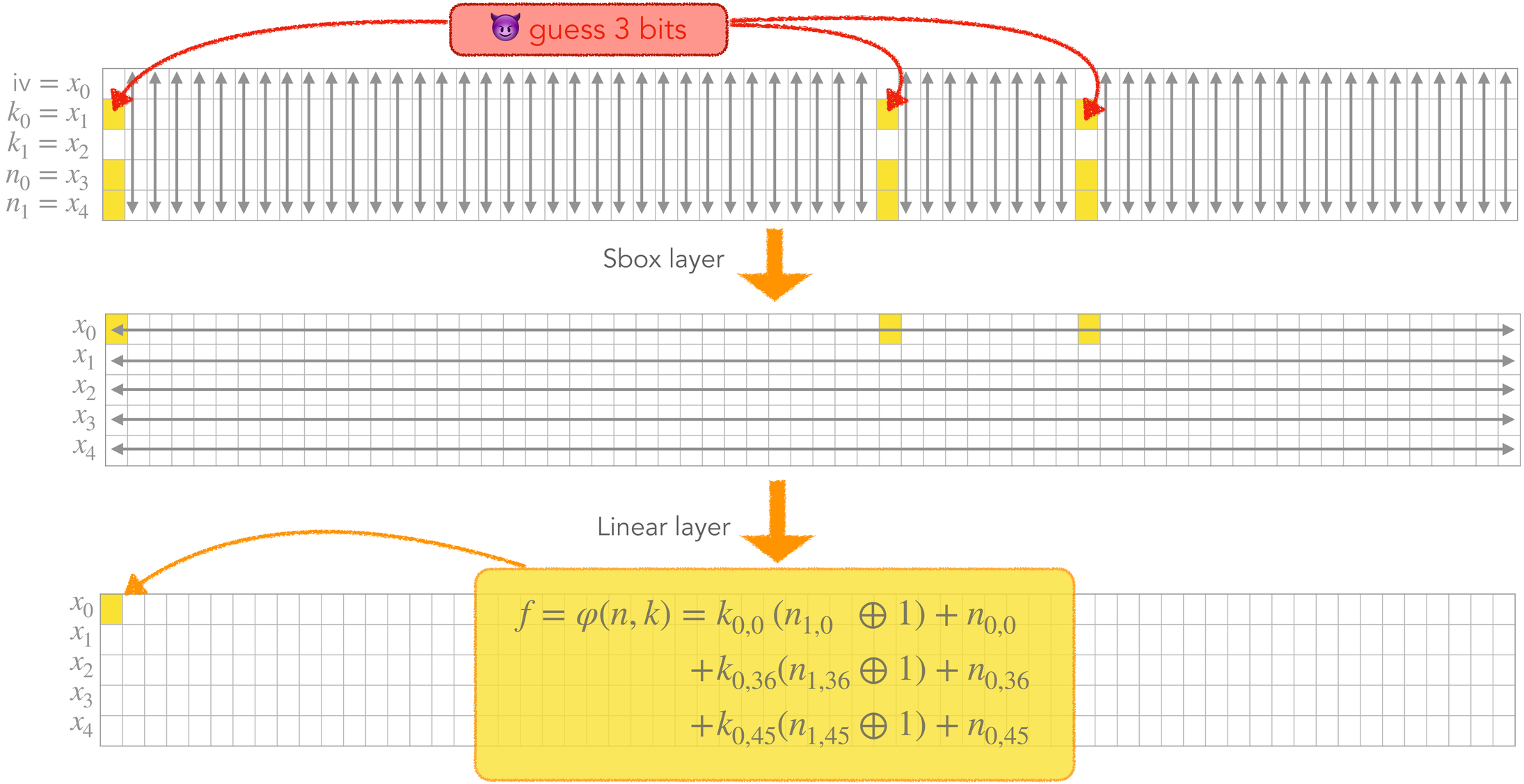
Magic happens 🪄: these are constants

→ contribute a *constant amount* to the activity of the register

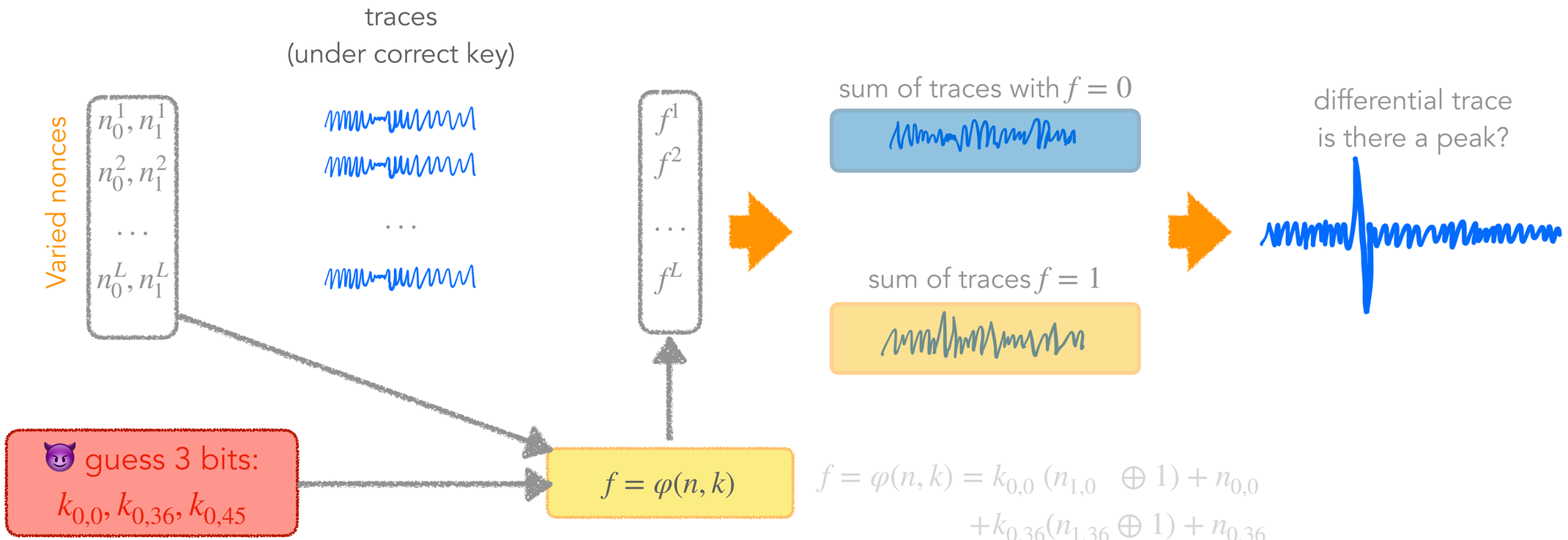
→ removed

$$y_0 \sim k_0(n_1 \oplus 1) \oplus n_0$$

Selection Function of [SD17]: Go Further...



DPA of [SD17]



Recover key successfully with ~1.5K traces! 😈 **WHY?** 🤔

Outlines

1. Background
   - Differential and Correlation Power Analysis
   - Ascon
2. Previous Attacks
   - [SD17] and [RADKA20]
3. Our work
   - Comparison of Previous Attacks
   - Correlation Power Analysis with Less Traces
4. Conclusion

Compare [SD17] and [RADKA20]

- [RADKA20]

  Cannot recover key with 40K traces! 😈 **WHY?** 🤔

- [SD17]

  Recover key successfully with ~1.5K traces! 😈 **WHY?** 🤔

- → Selection functions $f = \varphi(n, k)$ are different

[RADKA20]: Cannot recover key with 40K traces! 😈

$$f = \varphi(n, k) = n_1 k_0 \oplus n_0 \oplus k_1 k_0 \oplus k_1 \oplus k_0 \mathsf{iv} \oplus k_0 \oplus \mathsf{iv}$$

iv = 0 to simplify

$$f = \varphi(n, k) = n_1 k_0 \oplus n_0 \oplus k_1 k_0 \oplus k_1 \oplus k_0$$

varied nonce n_0, n_1	correct key $k_0^* = 1, k_1^* = 0$	f
00	10	1
01	10	0
10	10	0
11	10	1

varied nonce n_0, n_1	wrong key $k_0 = 1, k_1 = 1$	f
00	11	1
01	11	0
10	11	0
11	11	1

Cannot distinguish correct key guess and wrong key guess ⚠️

→ Bad choice of $f = \varphi(n, k)$

[SD17]: Recover key successfully with ~1.5K traces! 😈

$$f = \varphi(n, k) = k_0(n_1 \oplus 1) \oplus n_0$$

varied nonce n_0, n_1	correct key $k_0^* = 0$	f
00	0	0
01	0	0
10	0	1
11	0	1

varied nonce n_0, n_1	wrong key $k_0 = 1$	f
00	1	1
01	1	0
10	1	0
11	1	1

Able to distinguish correct key guess and wrong key guess

→ Good choice of $f = \varphi(n, k)$

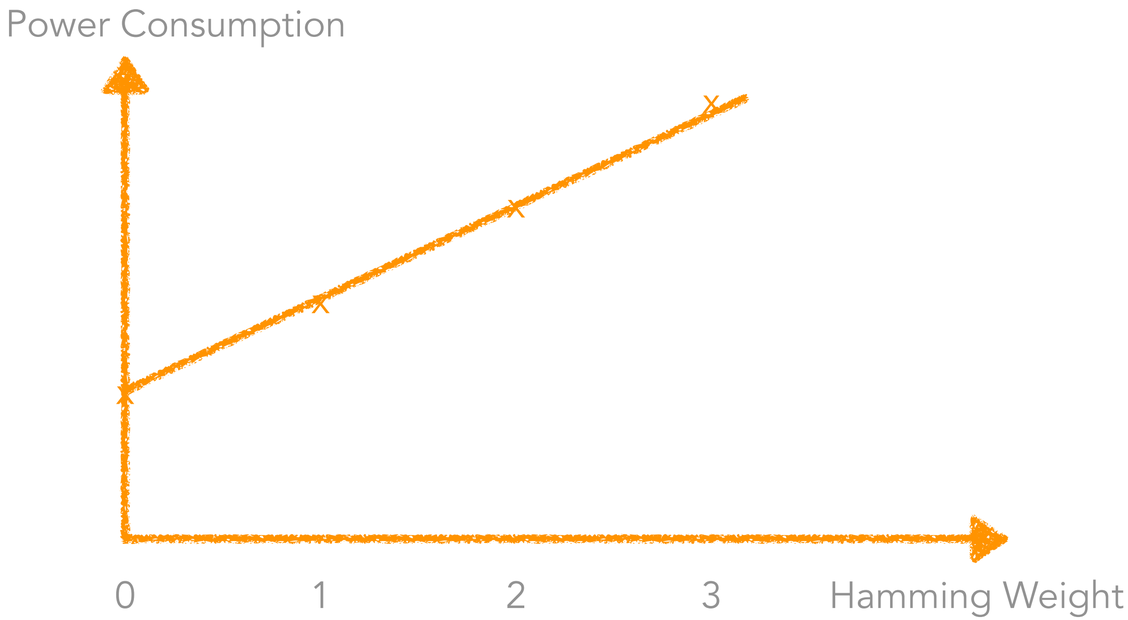
Outlines

1. Background
   - Differential and Correlation Power Analysis
   - Ascon
2. Previous Attacks
   - [SD17] and [RADKA20]
3. Our work
   - Comparison of Previous Attacks
   - Correlation Power Analysis with Less Traces
4. Conclusion

Remind: Correlation Power Analysis (CPA)

- Fact: Hamming Weight (HW) and power consumption have linear relation*

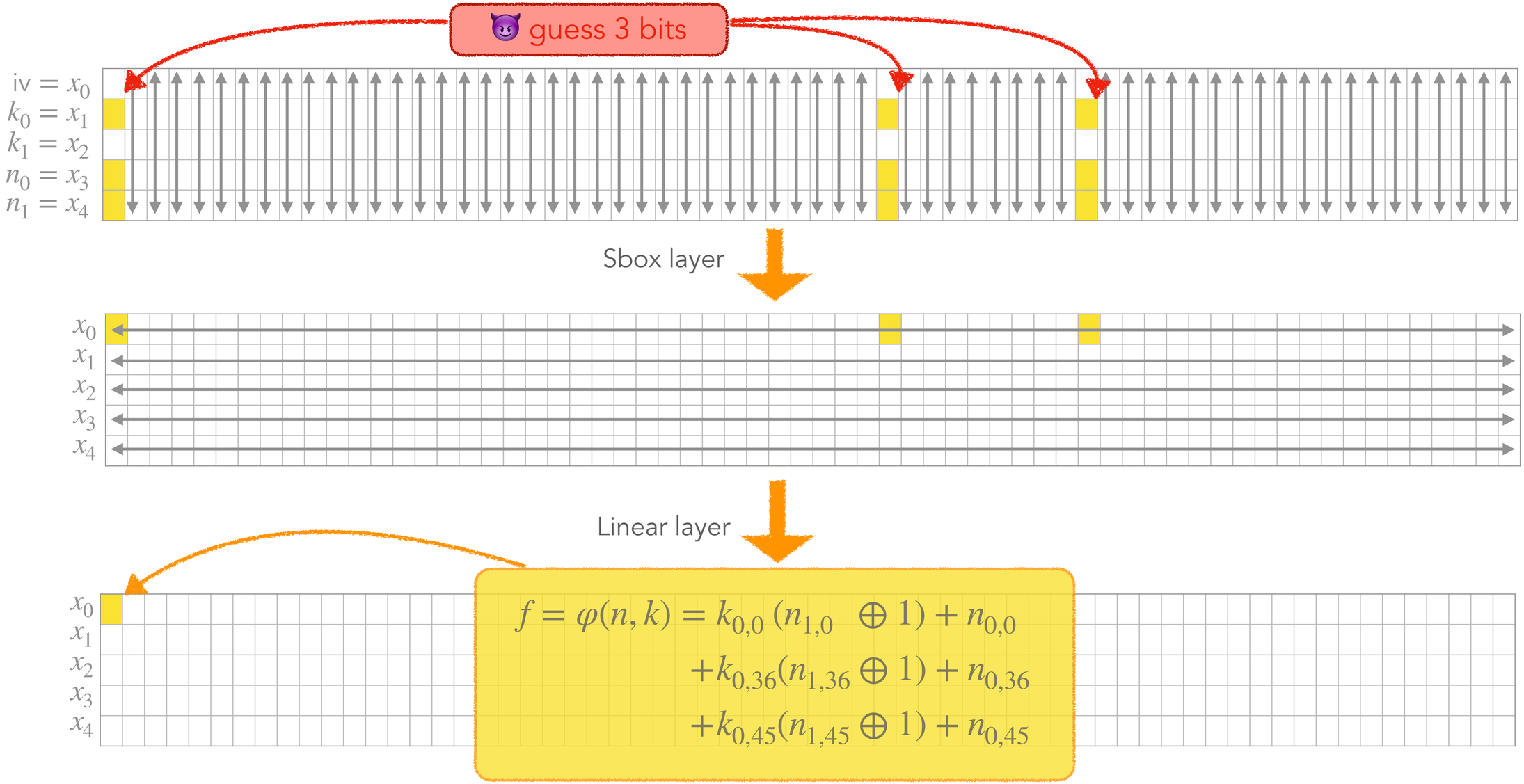
- Register transitions
  - HW = 3: consumes *most* power
    - 000 → 111
  - HW = 1: consumes *less* power
    - 000 → 001, 000 → 010, etc.
  - …



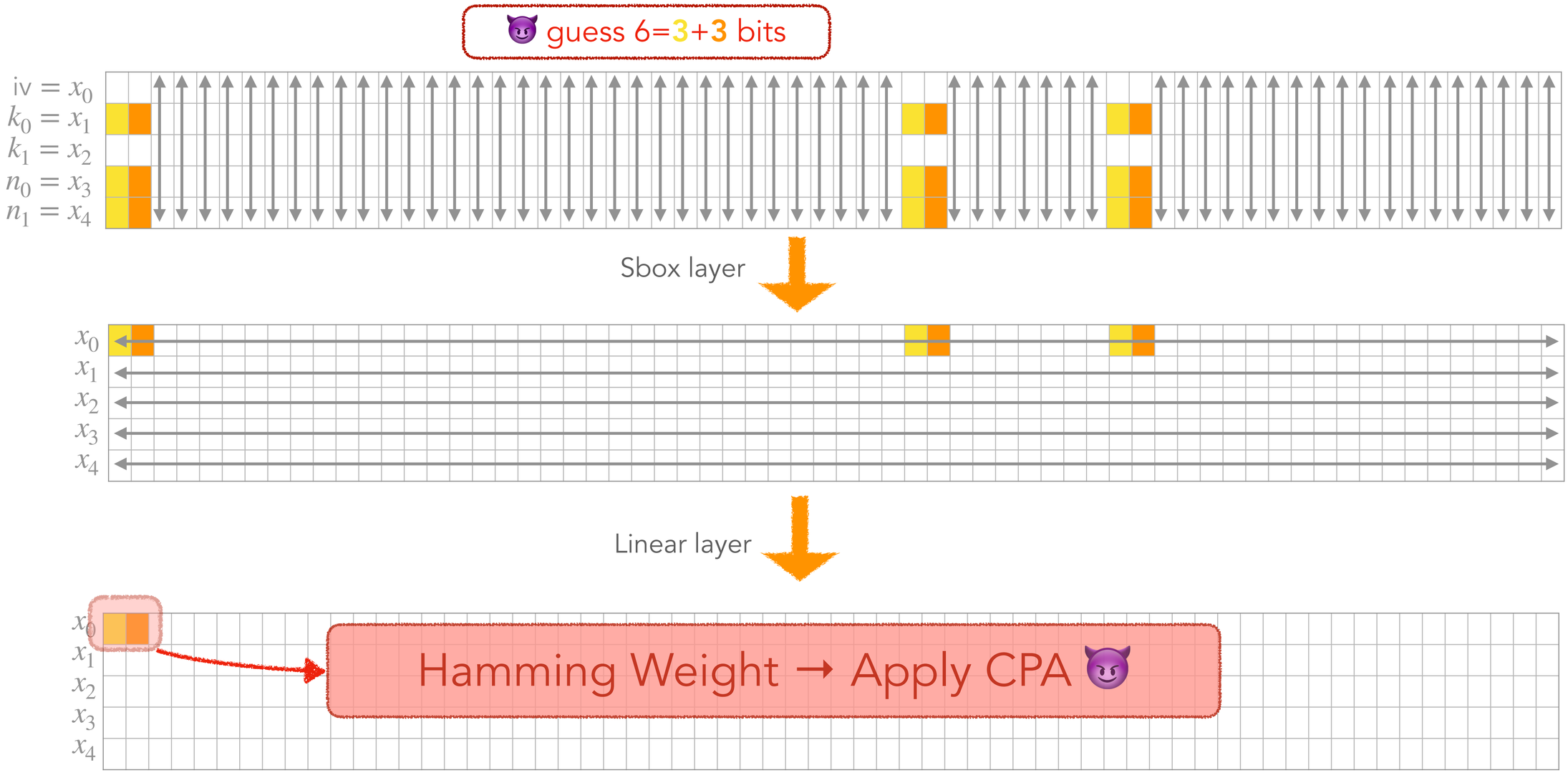
→ Need multiple-bit value of $f = \varphi(n, k)$ to have Hamming Weight

*Assume that each register is pre-charged at 0

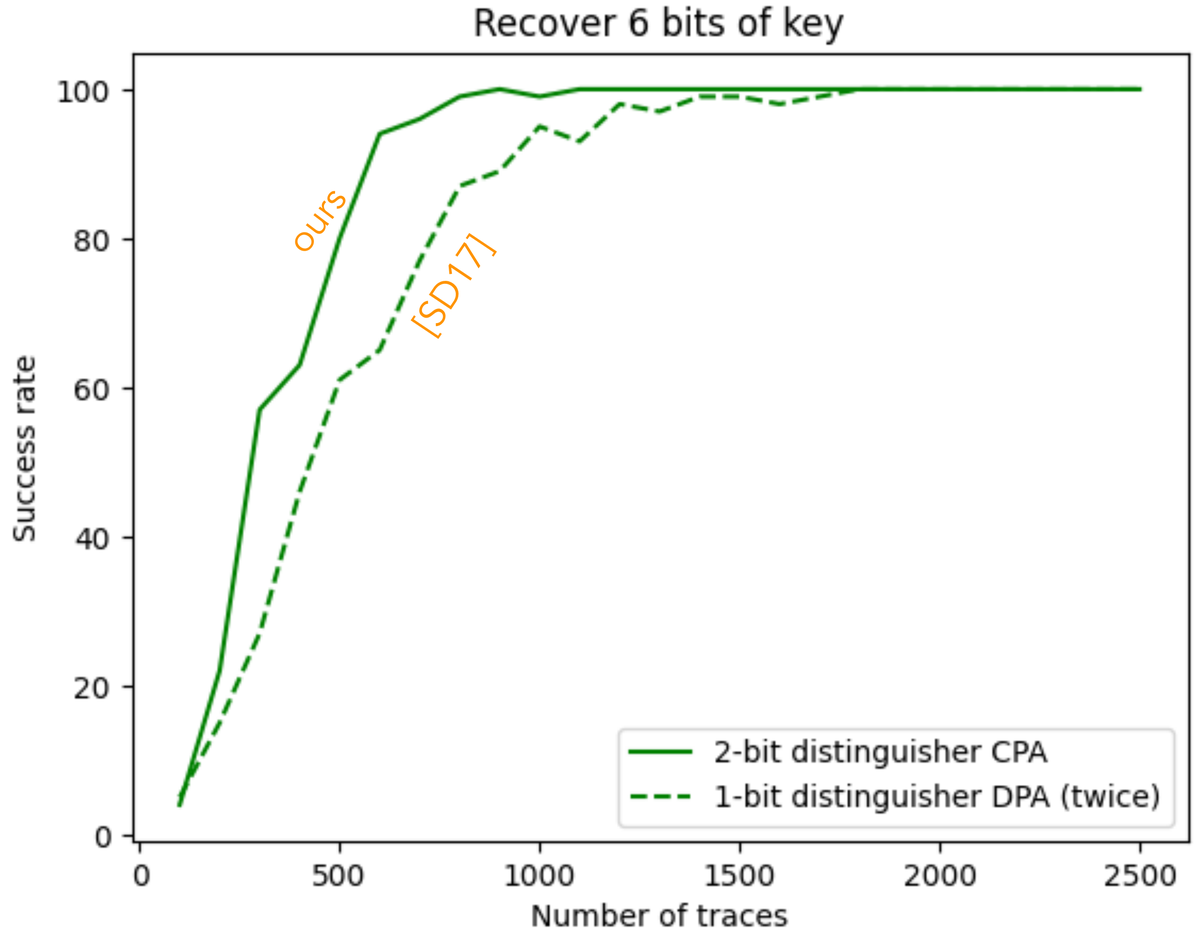
Remind: Selection Function of [SD17]



Extend Selection Function of [SD17]

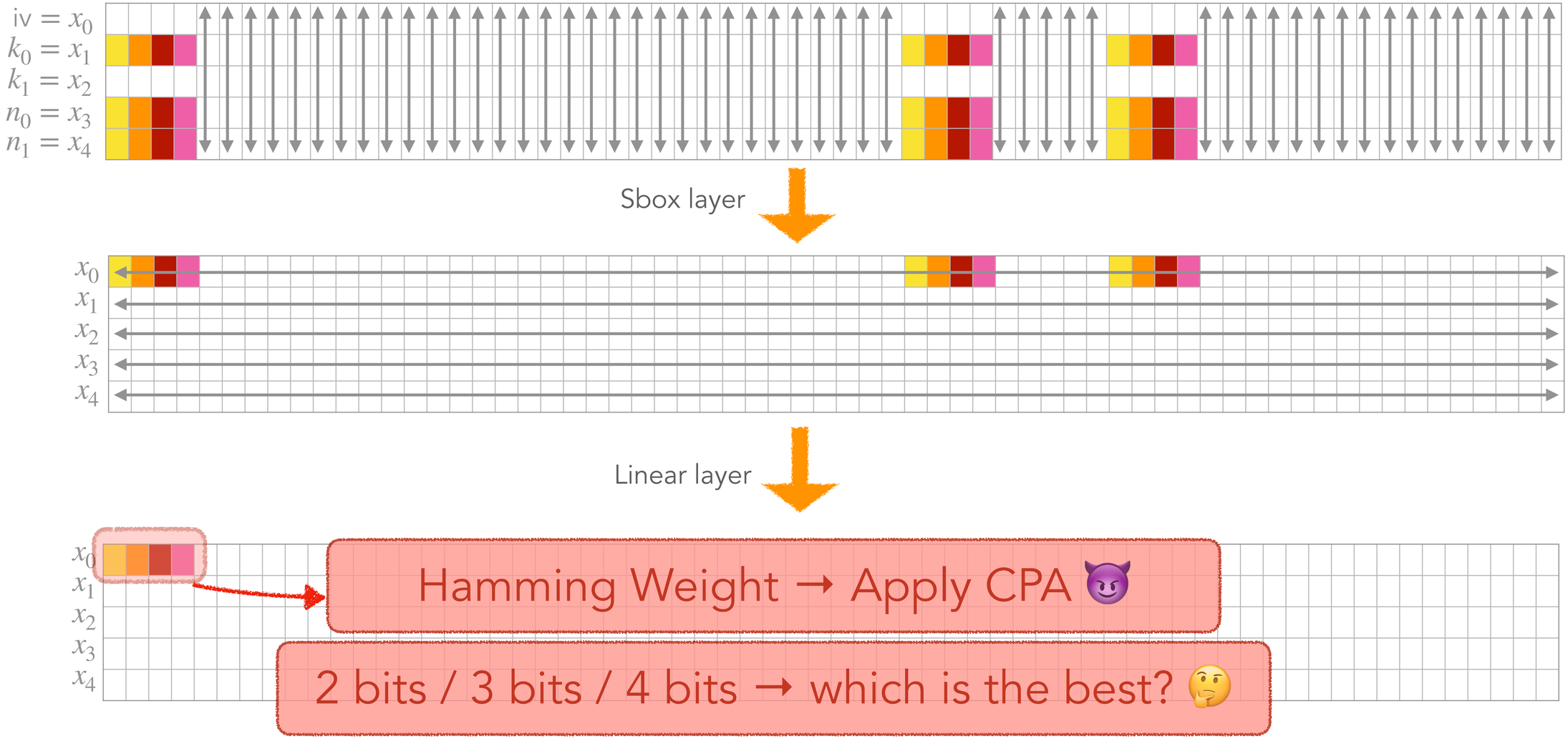


Our Result

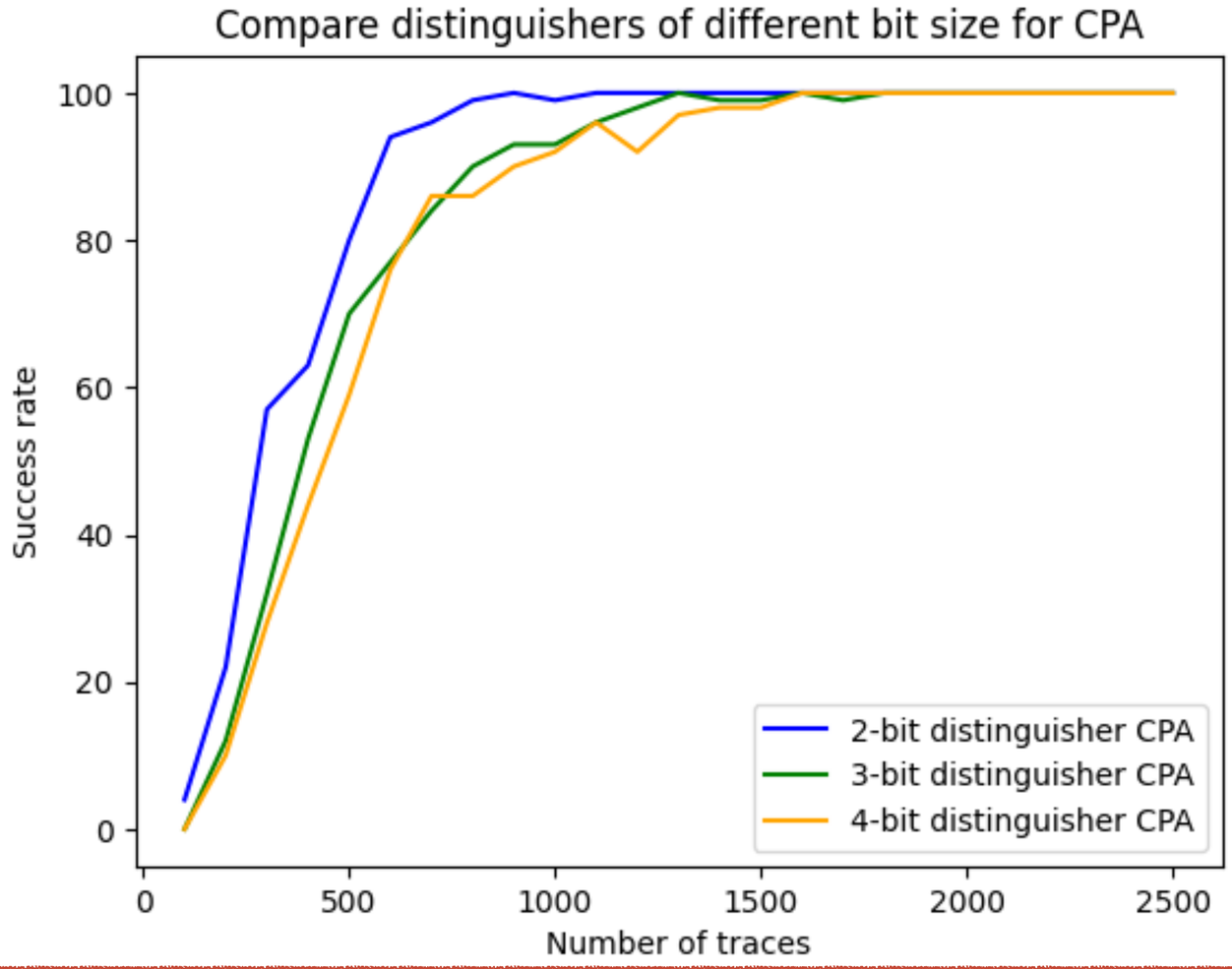


Our attack reduced number of traces ✅

Can we extend more?



Our Result



2-bit distinguisher is the best ✅

Outlines

1. Background
   - Differential and Correlation Power Analysis
   - Ascon
2. Previous Attacks
   - [SD17] and [RADKA20]
3. Our work
   - Comparison of Previous Attacks
   - Correlation Power Analysis with Less Traces
4. Conclusion

Conclusion

- Revisited DPA on Ascon
- Explained why
  - [RADKA20] DPA failed with 40K traces
  - [SD17] DPA succeeded with ~1.5K traces
- Extended [SD17] to
  - apply CPA and thus reduce number of traces
  - discover that 2-bit distinguisher is the best

Thank you!

Any questions? 🤔

References

- **[RADKA20]** Ramezanpour, Abdulgadir, Diehl, Kaps, Ampadu: “Active and Passive Side-Channel Key Recovery Attacks on Ascon”, NIST LWC Workshop 2020
- **[SD17]** Samwel, Daemen: “DPA on hardware implementation of Ascon and Keyak”, Computing Frontiers Conference 2017