RUHR-UNIVERSITÄT BOCHUM

Yet Another Size Record for AES:

A First-Order SCA Secure AES S-box Based on $GF(2^8)$ Multiplication

SPONSORED BY THE

Federal Ministry of Education and Research

Grant. Nr. 16KIS0666 SYSKIT_HW

Cardis 2018, Montpellier

Felix Wegener, Amir Moradi

Ruhr University Bochum, Horst Görtz Institute for IT-Security, Germany

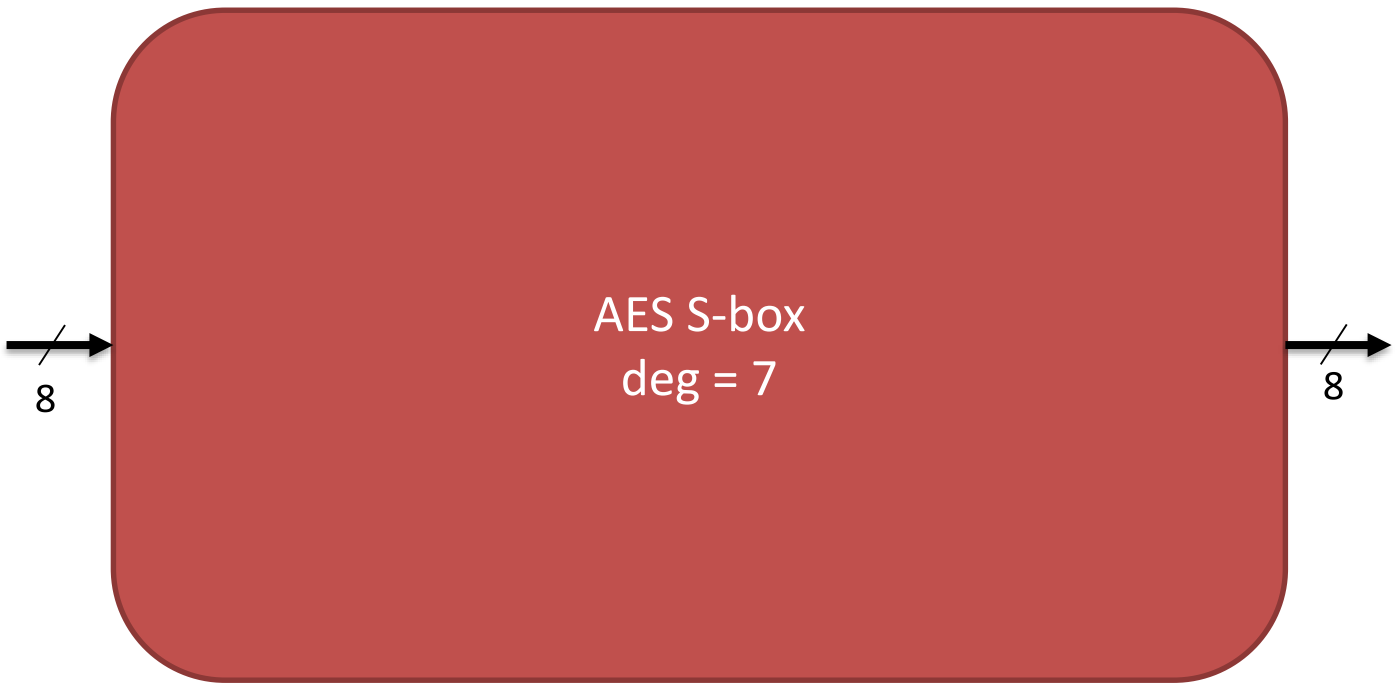
hgi Horst Görtz Institute for IT-Security

Problem:

How to find a small AES S-box implementation (with side-channel protection)?

AES S-box Implementations

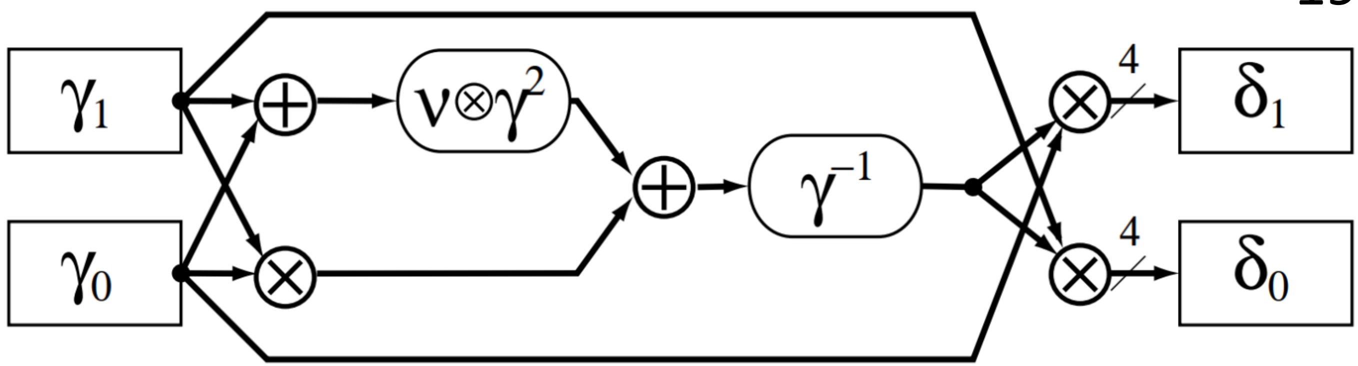
- Naive implementation:



AES S-box Implementations

- Canright:

195 Gates



Canright. *A Very Compact S-box for AES*. CHES 2005

- Boyar, Matthews, Peralta:

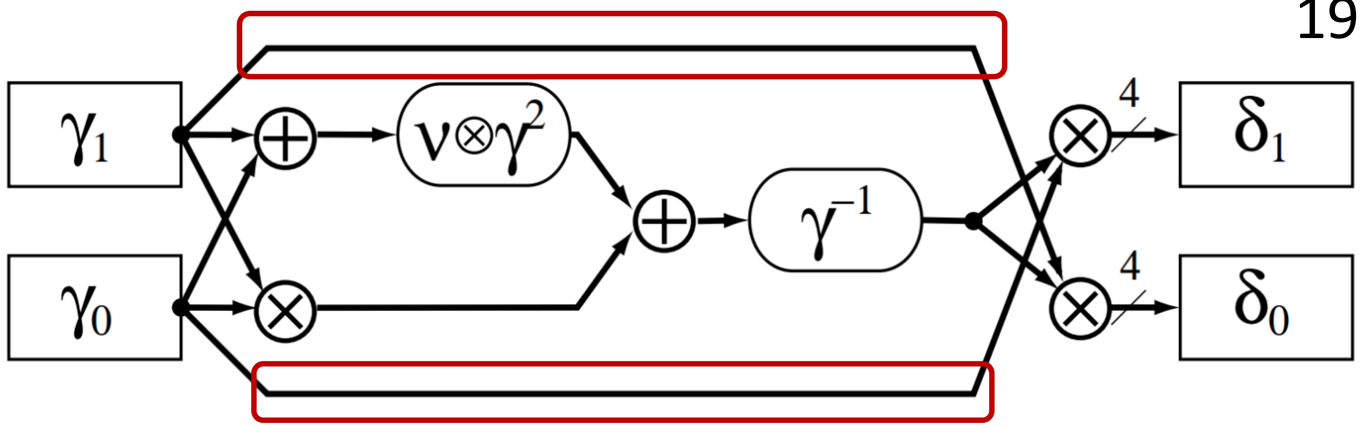
115 Gates

Boyar et al., *Logic Minimization Techniques with Applications to Cryptology*, J. Cryptology 2013

Issue I: Registers for Bypass Wires

- Canright:

195 Gates



Canright. *A Very Compact S-box for AES*. CHES 2005

- Boyar, Matthews, Peralta:

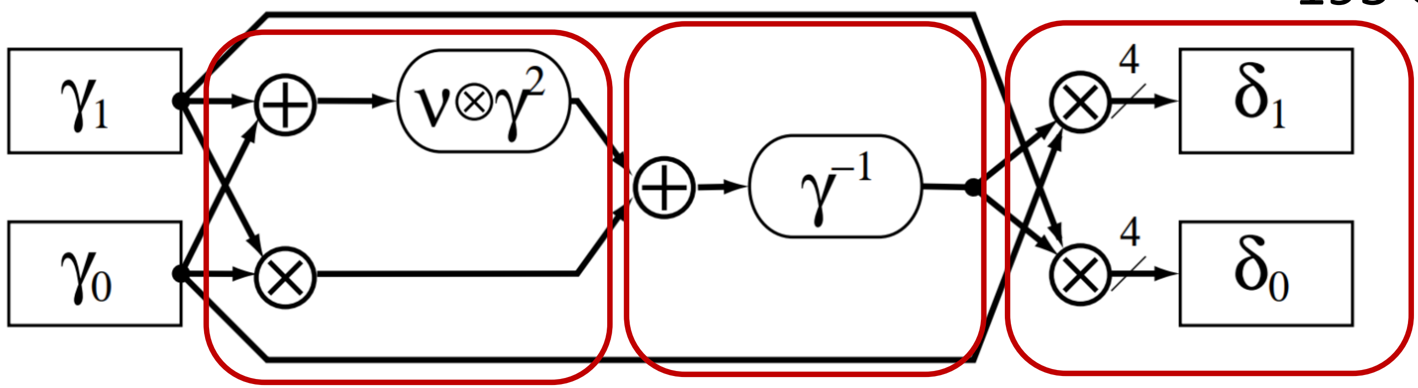
115 Gates

Boyar et al., *Logic Minimization Techniques with Applications to Cryptology*, J. Cryptology 2013

Issue II: No Serialization Possible

- Canright:

195 Gates



Canright. *A Very Compact S-box for AES*. CHES 2005

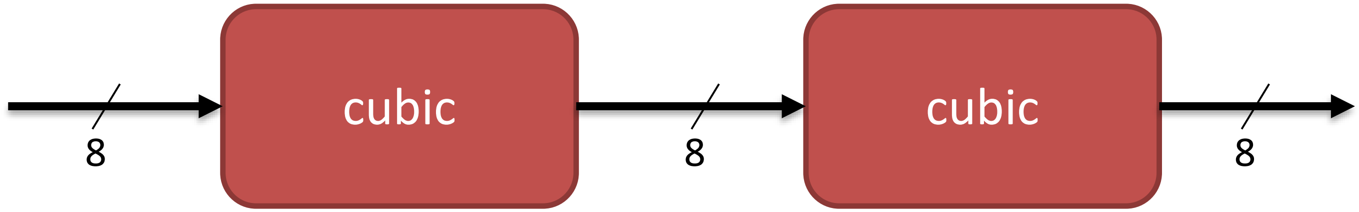
- Boyar, Matthews, Peralta:

115 Gates

Boyar et al., *Logic Minimization Techniques with Applications to Cryptology*, J. Cryptology 2013

A Different Structure

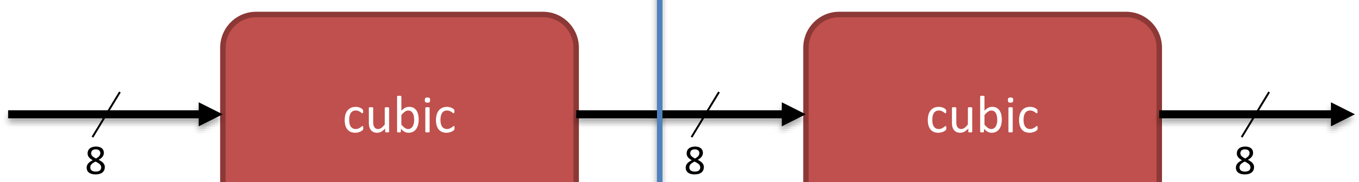
- In previous work:



Wegener, Moradi. *A first-order SCA resistant AES without fresh randomness*. COSADE 2018

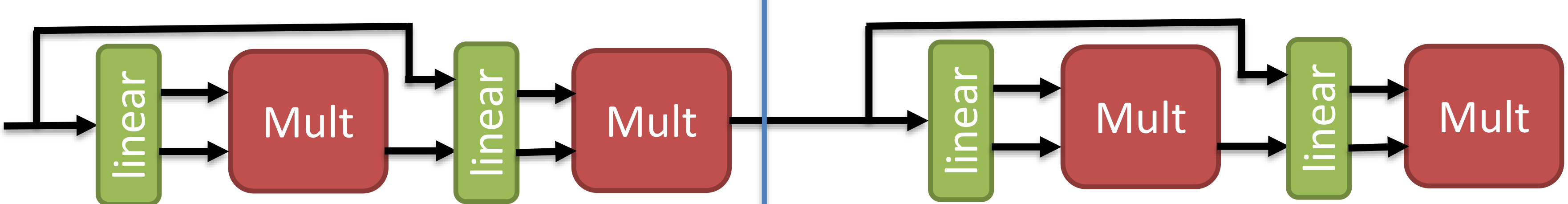
A Different Structure: Multiplication-based

- In previous work:



Wegener, Moradi. *A first-order SCA resistant AES without fresh randomness*. COSADE 2018

- This work:



Decomposition into Multiplications

Structure of AES S-box

- AES-Sbox (x): $Aff(x^{-1})$
- Inversion in $GF(2^8)$: $x^{-1} = x^{254}$

Structure of AES S-box

- AES-Sbox (x): $Aff(x^{-1})$
- Inversion in $GF(2^8)$: $x^{-1} = x^{254}$
- How many multiplications are necessary?
  → Find shortest multiplication chain

Multiplication Chain

- Start: $id = x^1$
- Step:
  - Square a previous element → cost = 0
  - Multiply two previous elements → cost = 1

Multiplication Chain

- Start: $id = x^1$
- Step:
  - – Square a previous element → cost = 0
  - – Multiply two previous elements → cost = 1
- Example chain for x^{13} :

Chain: x^1,

Cost: 0,

Multiplication Chain

- Start: $id = x^1$
- Step:
  - – Square a previous element → cost = 0
  - – Multiply two previous elements → cost = 1
- Example chain for x^{13} :

Chain:	x^1,	x^2,
Cost:	0,	0,

Multiplication Chain

- Start: $id = x^1$
- Step:
  - – Square a previous element → cost = 0
  - – Multiply two previous elements → cost = 1
- Example chain for x^{13} :

Chain:	x^1,	x^2,	x^4,
Cost:	0,	0,	0,

Multiplication Chain

- Start: $id = x^1$
- Step:
  - Square a previous element → cost = 0
  - Multiply two previous elements → cost = 1
- Example chain for x^{13} :

Chain:	x^1,	x^2,	x^4,	x^8,
Cost:	0,	0,	0,	0,

Multiplication Chain

- Start: $id = x^1$
- Step:
  - – Square a previous element → cost = 0
  - – Multiply two previous elements → cost = 1
- Example chain for x^{13} :

Chain:	x^1,	x^2,	x^4,	x^8,	x^{12} ,
Cost:	0,	0,	0,	0,	1,

Multiplication Chain

- Start: $id = x^1$
- Step:
  - Square a previous element → cost = 0
  - Multiply two previous elements → cost = 1
- Example chain for x^{13} :

Chain:	x^1,	x^2,	x^4,	x^8,	x^{12} ,	x^{13}
Cost:	0,	0,	0,	0,	1,	2

Multiplication Chain

- Start: $id = x^1$
- Step:
  - – Square a previous element → cost = 0
  - – Multiply two previous elements → cost = 1
- Example chain for x^{13} :

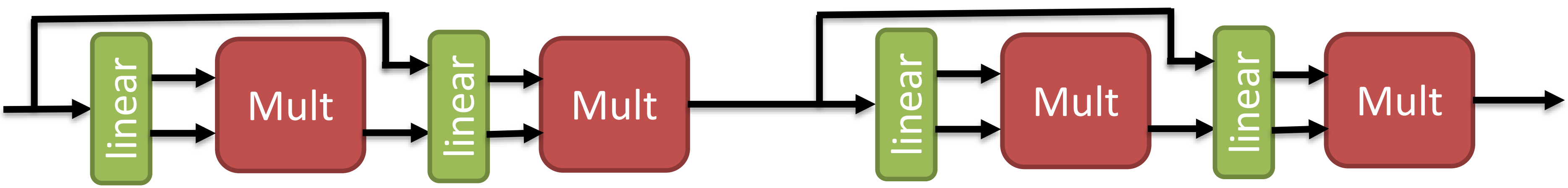
Chain:	x^1,	x^2,	x^4,	x^8,	x^{12} ,	x^{13}
Cost:	0,	0,	0,	0,	1,	2

- Lowest cost of chain for x^{254}: **4**

Multiplication Chain

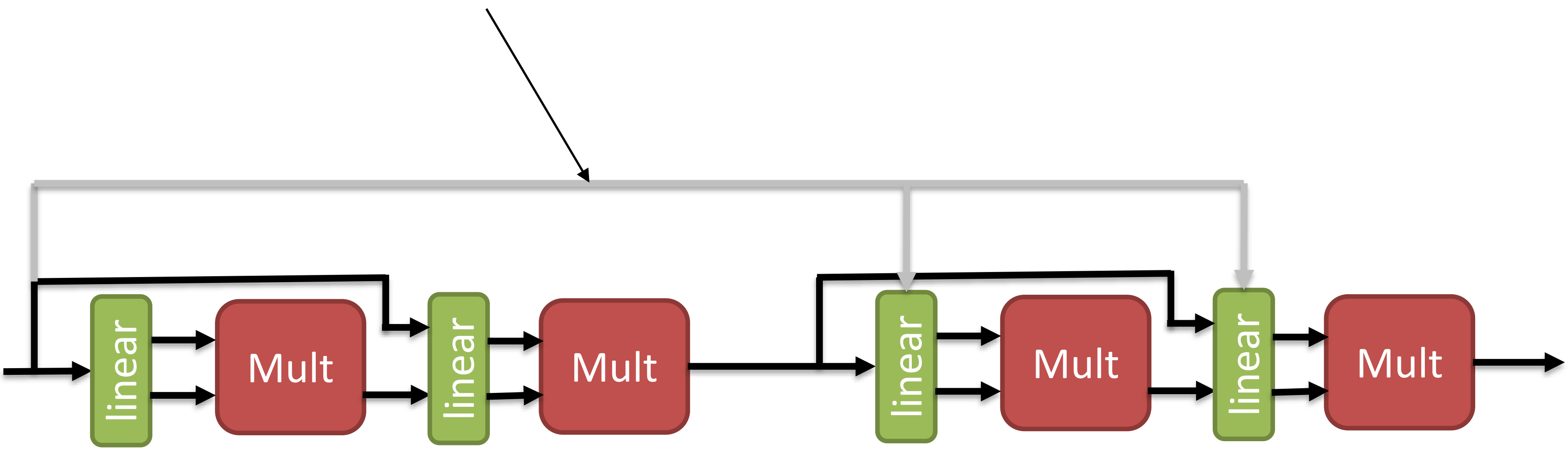
What is the "best" way to implement x^{254} with 4 multiplications?

Area Reduction Techniques



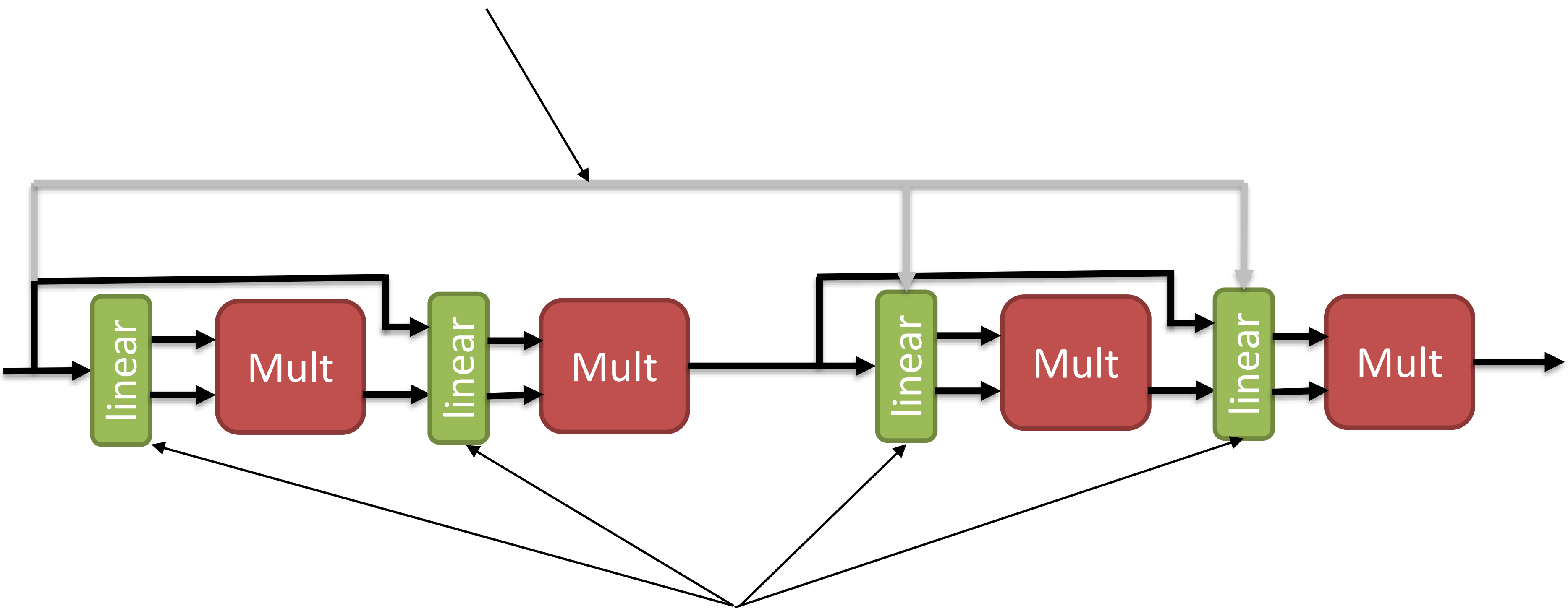
Area Reduction Techniques

- Limit bypass wires



Area Reduction Techniques

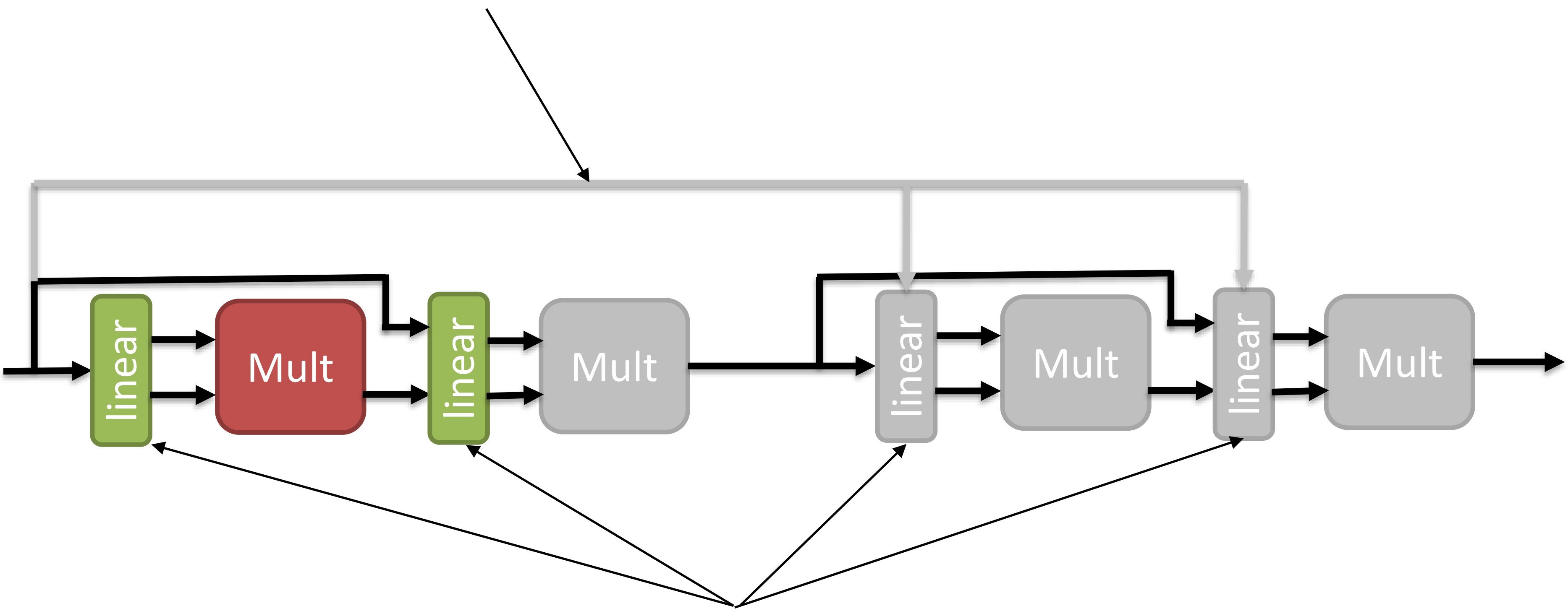
- Limit bypass wires



- Minimize linear components

Area Reduction Techniques

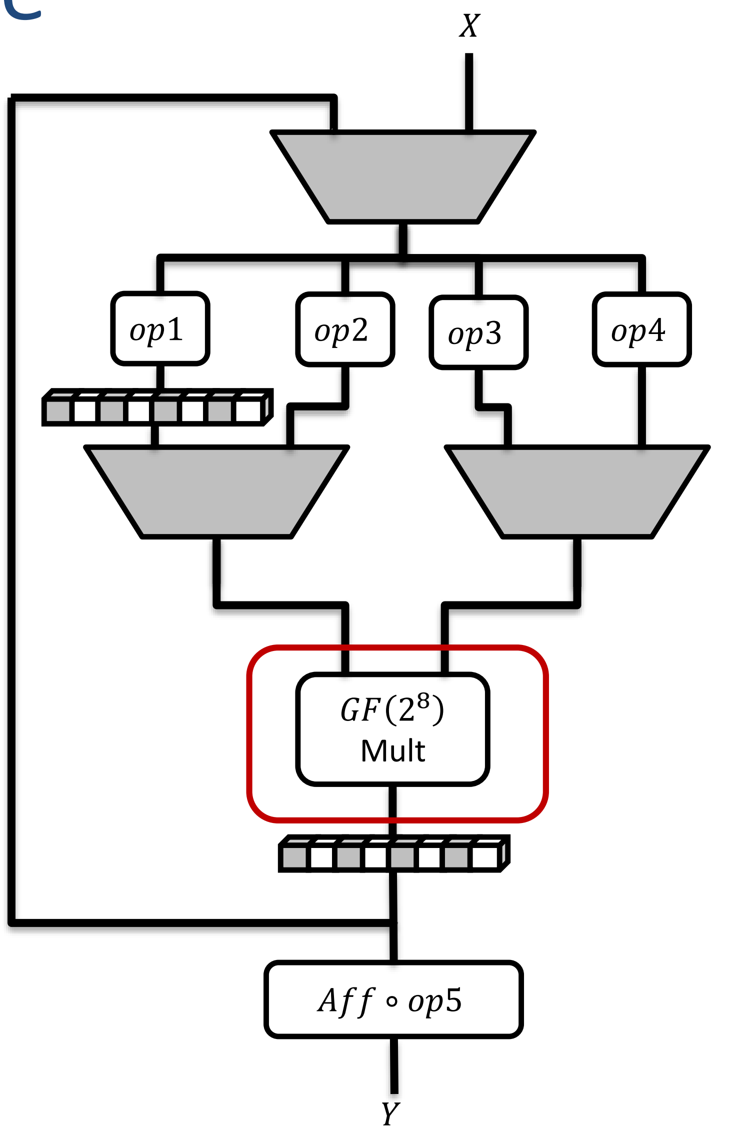
- Limit bypass wires



- Minimize linear components
- Serialize: One Multiplier instance

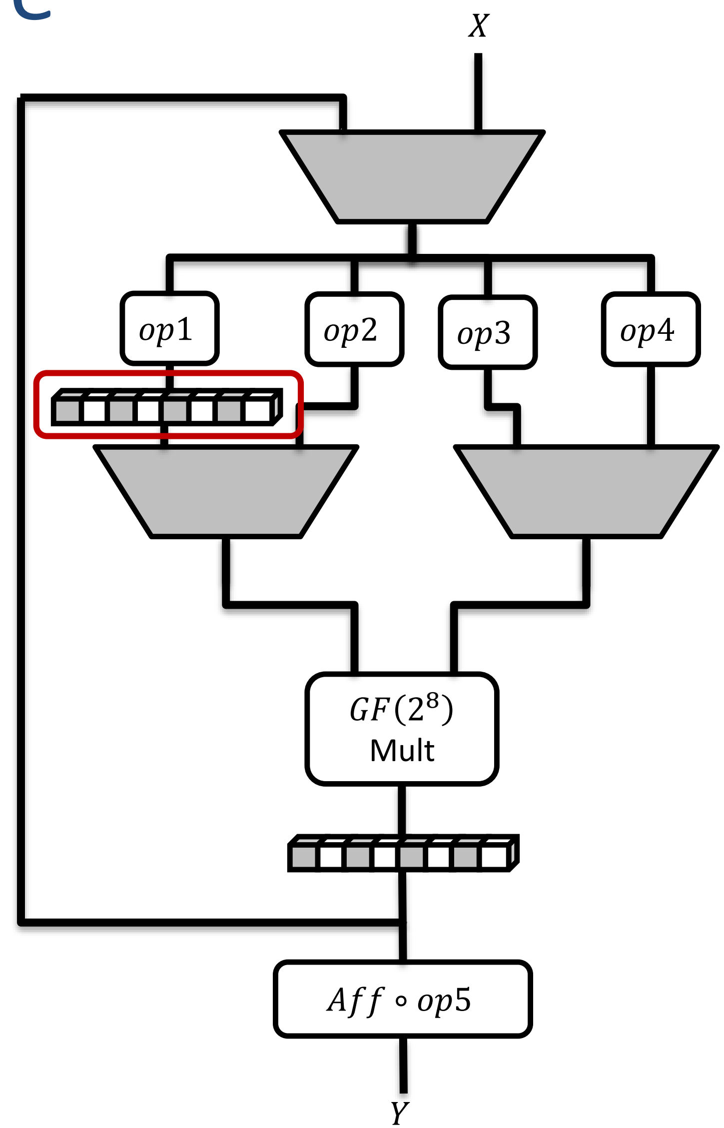
Our Design: High-Level Structure

- One multiplier instance



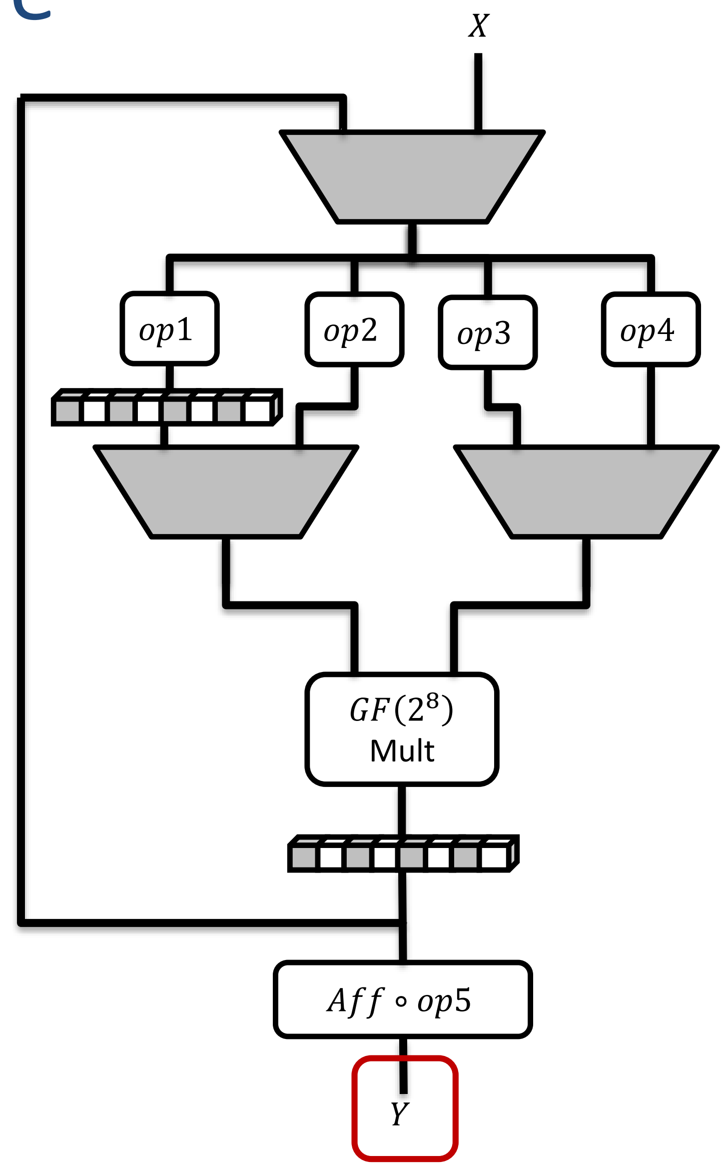
Our Design: High-Level Structure

- One multiplier instance
- One bypass wire



Our Design: High-Level Structure

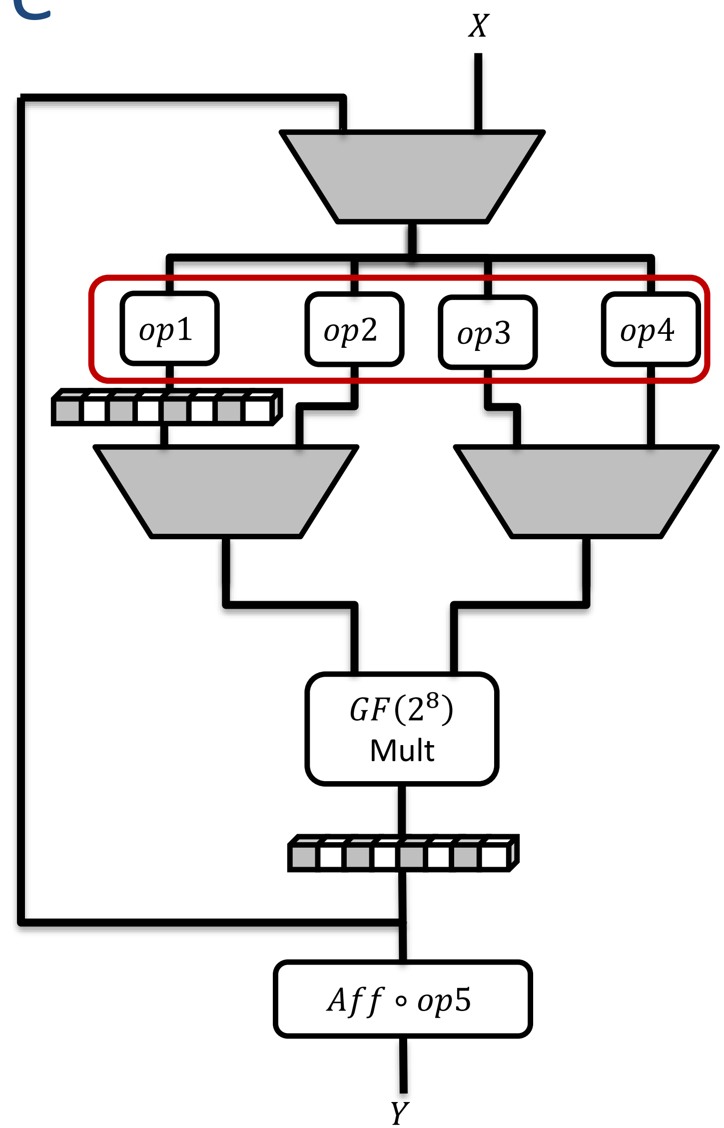
- One multiplier instance
- One bypass wire
- $Y = Sbox(X)$ (after 4 iterations)



Our Design: High-Level Structure

- One multiplier instance
- One bypass wire
- $Y = Sbox(X)$ (after 4 iterations)
- Linear power functions of form

$$op_i = x^{2^k}$$

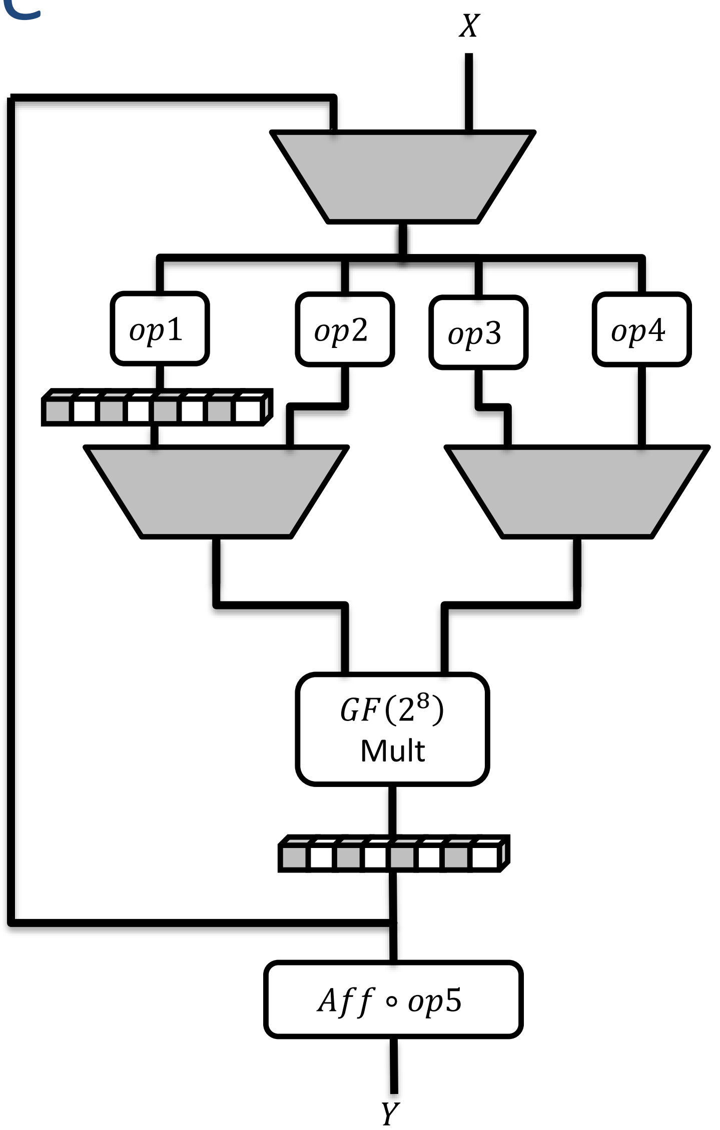


Our Design: High-Level Structure

- One multiplier instance
- One bypass wire
- $Y = Sbox(X)$ (after 4 iterations)
- Linear power functions of form

$$op_i = x^{2^k}$$

Goal: Minimize total area



Area Minimization

Two Steps:

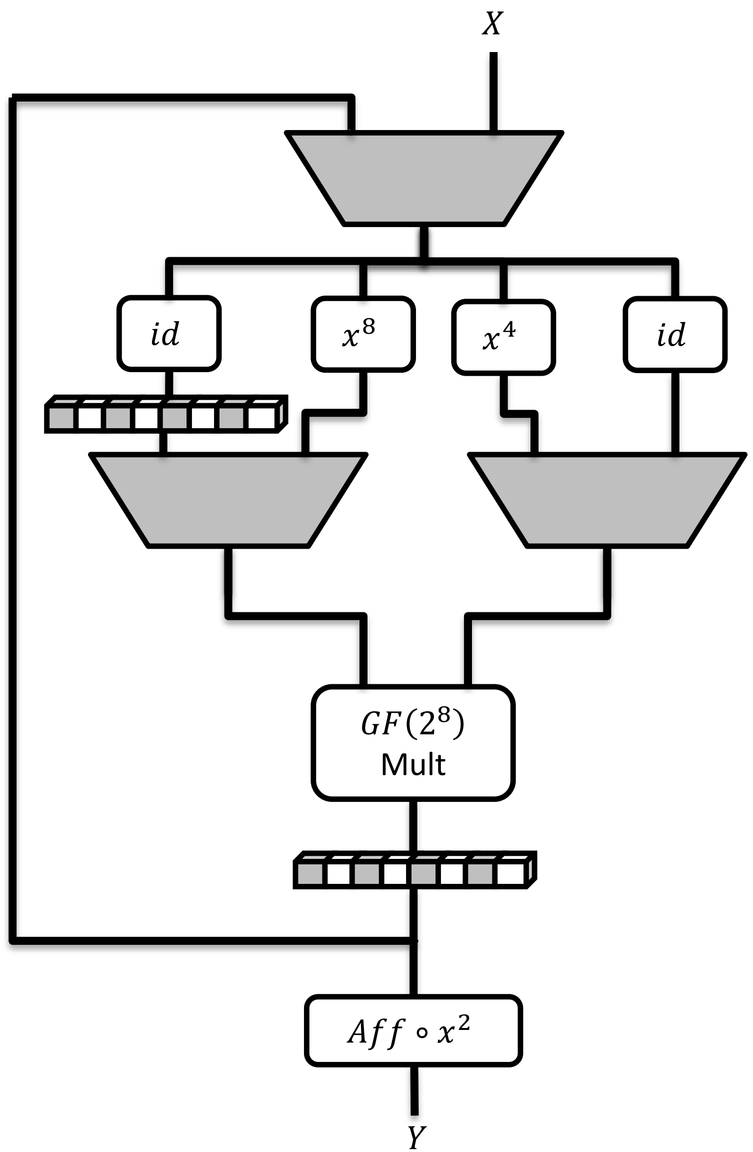
- Determine area of each linear component
- Choose op1, ..., op5 to minimize the total area

UMC 180 nm

Function	Area (GE)
x^{128}	23.7
x^{16}	33.3
x^{2}	22.7
x^{32}	33.3
x^{4}	31.7
x^{64}	29.7
x^{8}	32.0
Aff $\circ\, x^{1}$	41.7
Aff $\circ\, x^{128}$	40.7
Aff $\circ\, x^{16}$	36.3
Aff $\circ\, x^{2}$	40.3
Aff $\circ\, x^{32}$	36.7
Aff $\circ\, x^{4}$	36.3
Aff $\circ\, x^{64}$	29.7
Aff $\circ\, x^{8}$	34.0

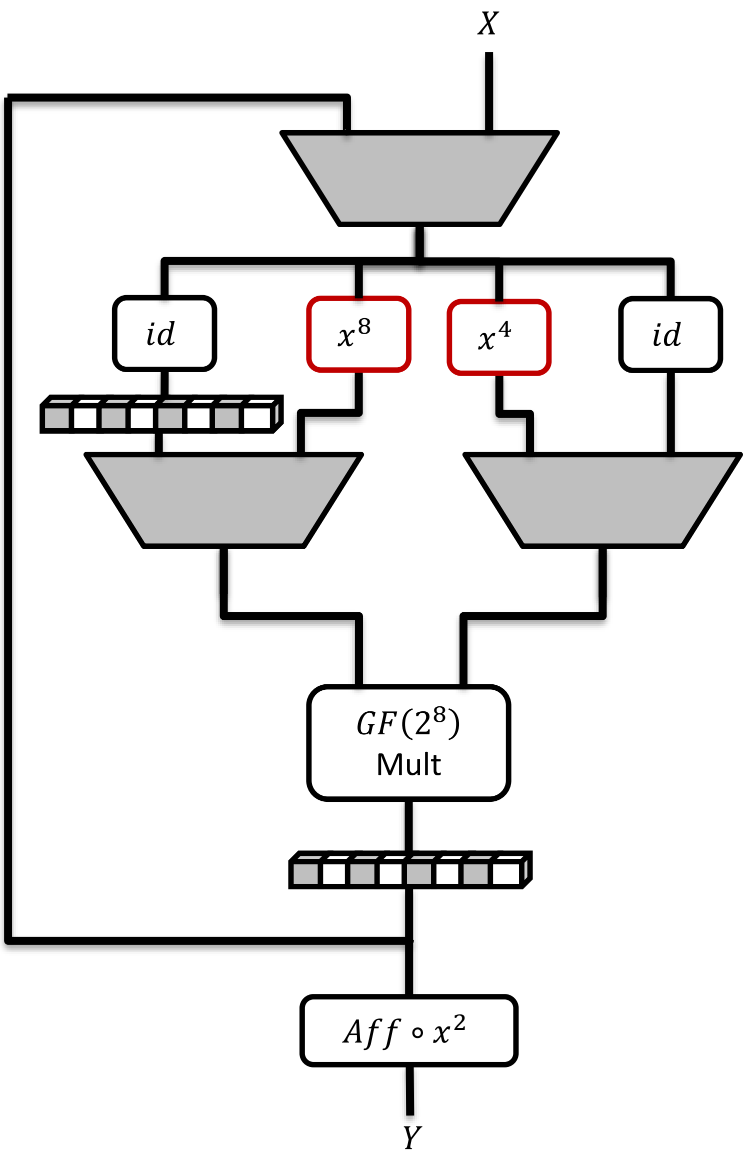
$x^{128} \,\|\|\, x^{16}$	52.3
$x^{128} \,\|\|\, x^{2}$	41.3
$x^{128} \,\|\|\, x^{32}$	49.0
$x^{128} \,\|\|\, x^{4}$	50.7
$x^{128} \,\|\|\, x^{64}$	43.7
$x^{128} \,\|\|\, x^{8}$	47.0
$x^{16} \,\|\|\, x^{2}$	44.7
$x^{16} \,\|\|\, x^{4}$	54.3
$x^{16} \,\|\|\, x^{8}$	54.3
$x^{32} \,\|\|\, x^{16}$	49.7
$x^{32} \,\|\|\, x^{2}$	45.0
$x^{32} \,\|\|\, x^{4}$	52.3
$x^{32} \,\|\|\, x^{8}$	53.0
$x^{4} \,\|\|\, x^{2}$	45.7
$x^{64} \,\|\|\, x^{16}$	53.7
$x^{64} \,\|\|\, x^{2}$	48.3
$x^{64} \,\|\|\, x^{32}$	53.0
$x^{64} \,\|\|\, x^{4}$	53.7
$x^{64} \,\|\|\, x^{8}$	51.7
$x^{8} \,\|\|\, x^{2}$	44.0
$x^{8} \,\|\|\, x^{4}$	52.0

Area Minimal Choice



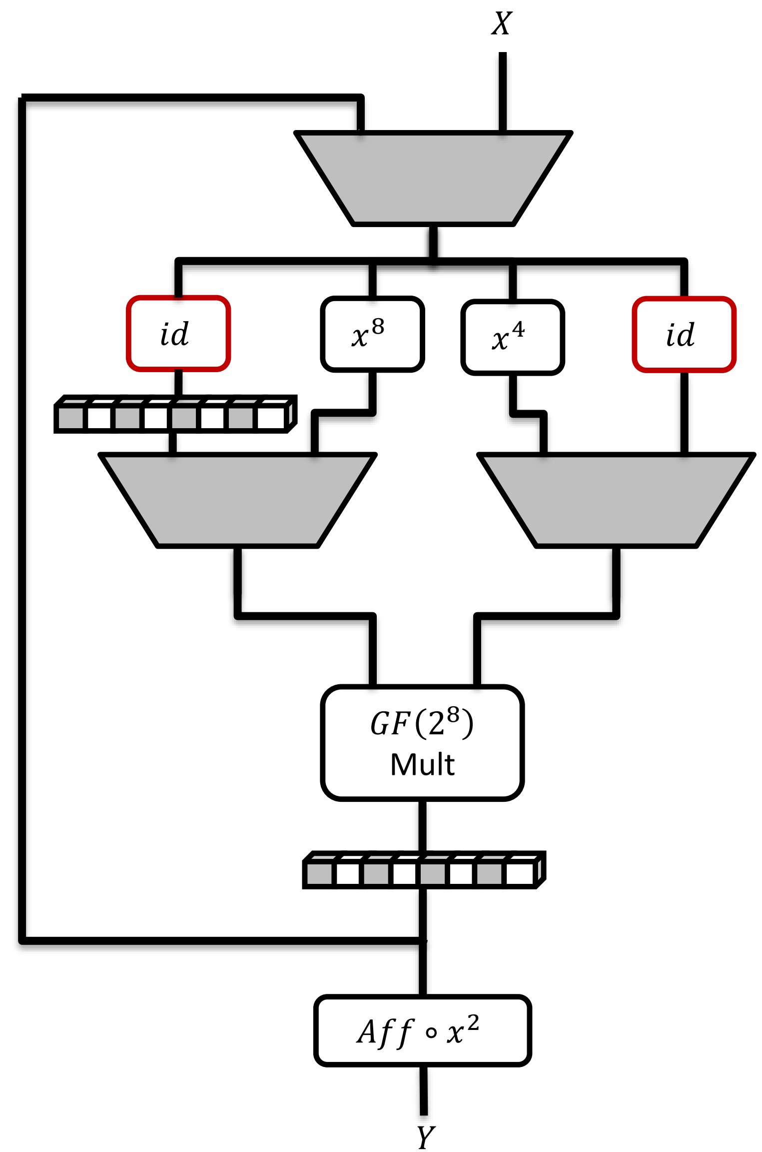
Area Minimal Choice

- Iteration 1:
  $x^{12} = Mult(x^8, x^4)$



Area Minimal Choice

- Iteration 1:
  $x^{12} = Mult(x^8, x^4)$
- Iteration 2:
  $x^{13} = Mult(x^1, x^{12}) =: z$



Area Minimal Choice

- Iteration 1:

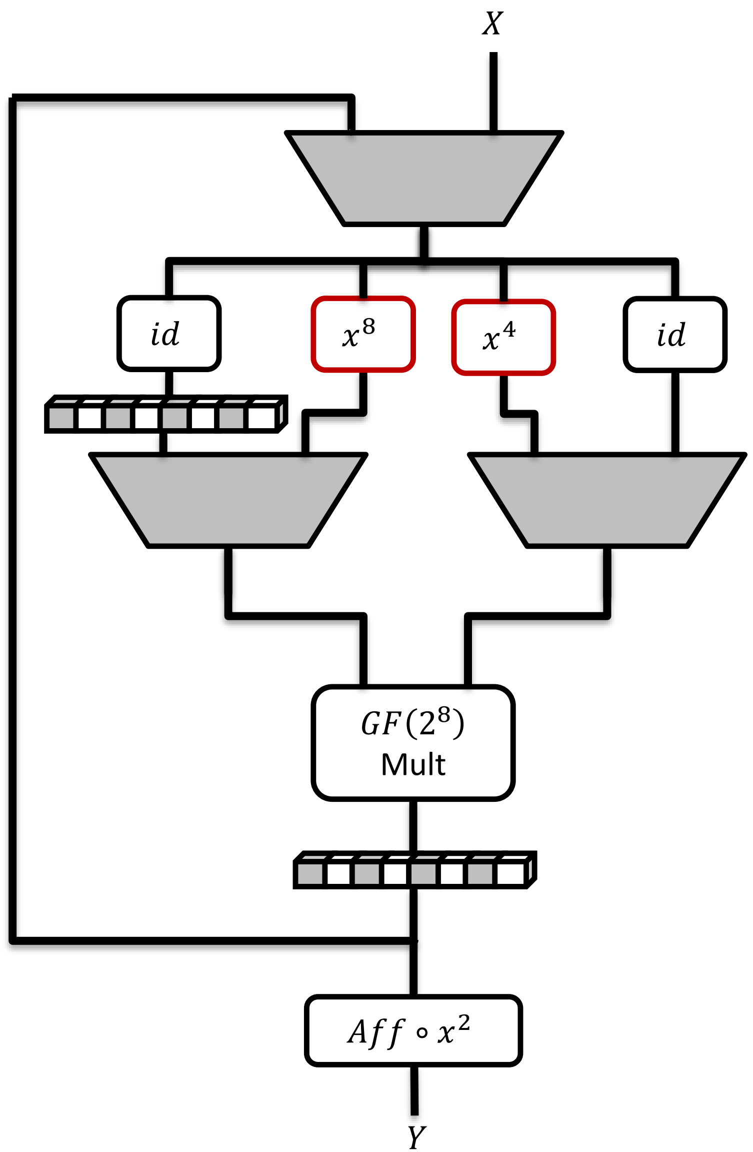
  $x^{12} = Mult(x^8, x^4)$

- Iteration 2:

  $x^{13} = Mult(x^1, x^{12}) =: z$

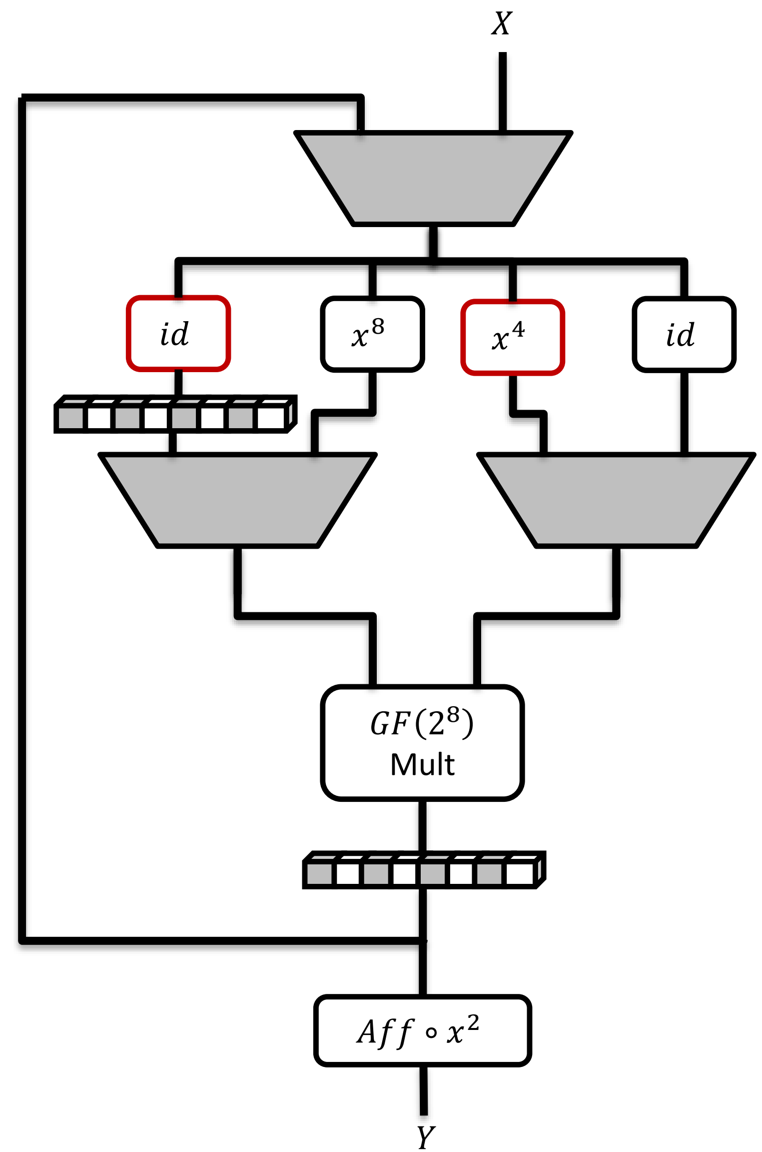
- Iteration 3:

  $z^{12} = Mult(z^8, z^4)$



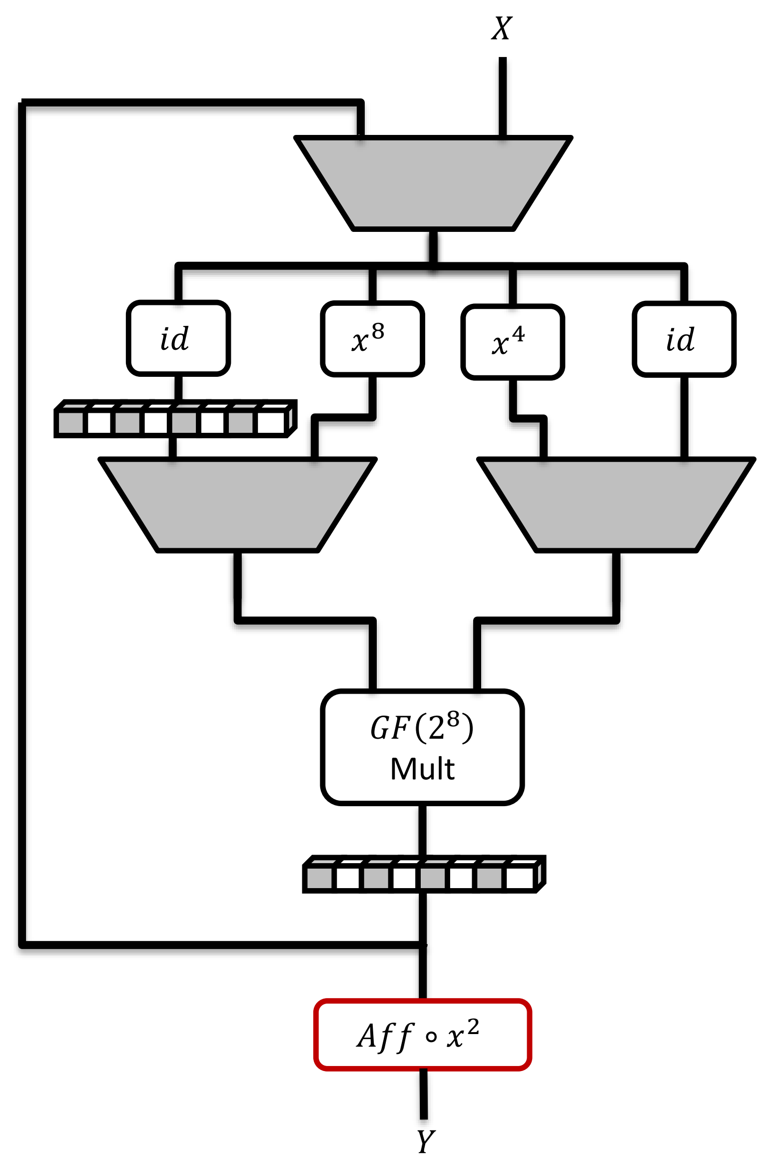
Area Minimal Choice

- Iteration 1:
  $x^{12} = Mult(x^8, x^4)$
- Iteration 2:
  $x^{13} = Mult(x^1, x^{12}) =: z$
- Iteration 3:
  $z^{12} = Mult(z^8, z^4)$
- Iteration 4:
  $z^{49} = Mult(z^1, z^{48})$



Area Minimal Choice

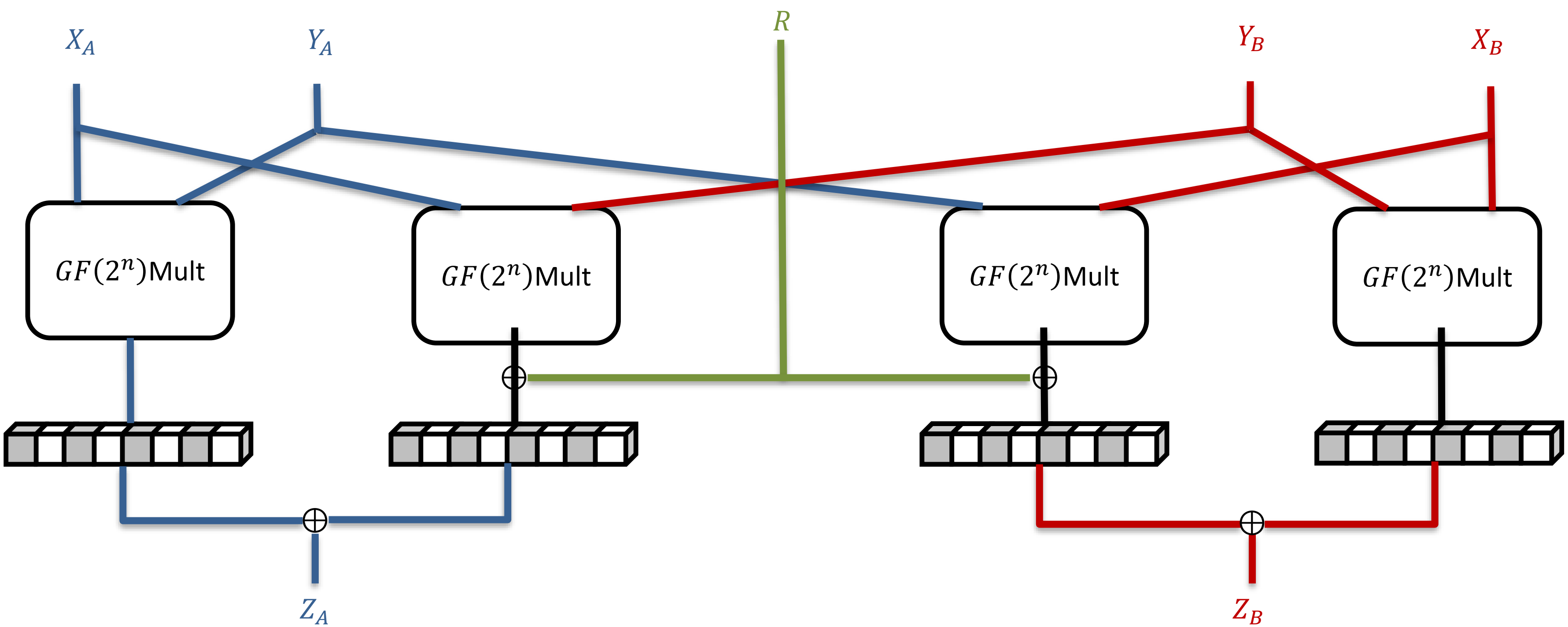
- Iteration 1:
  $x^{12} = Mult(x^8, x^4)$
- Iteration 2:
  $x^{13} = Mult(x^1, x^{12}) =: z$
- Iteration 3:
  $z^{12} = Mult(z^8, z^4)$
- Iteration 4:
  $z^{49} = Mult(z^1, z^{48})$
- Output:
  $Y = Aff(x^{13 \cdot 49 \cdot 2}) = Aff(x^{254})$



Achieving SCA Security

Domain-oriented Masking

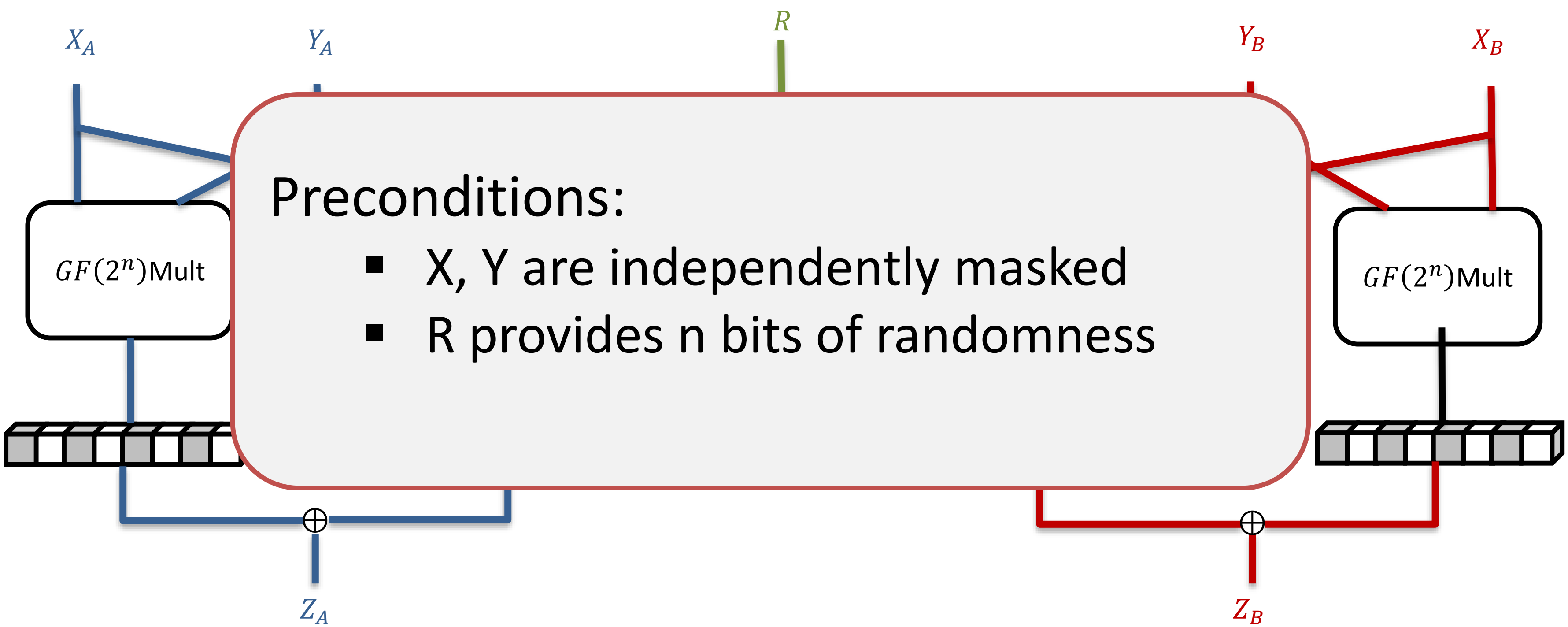
First-order DOM-independent multiplier:



Groß et al. *Domain-Oriented Masking: Compact Masked Hardware Implementations with Arbitrary Protection Order*, CCS 2016

Domain-oriented Masking

First-order DOM-independent multiplier:

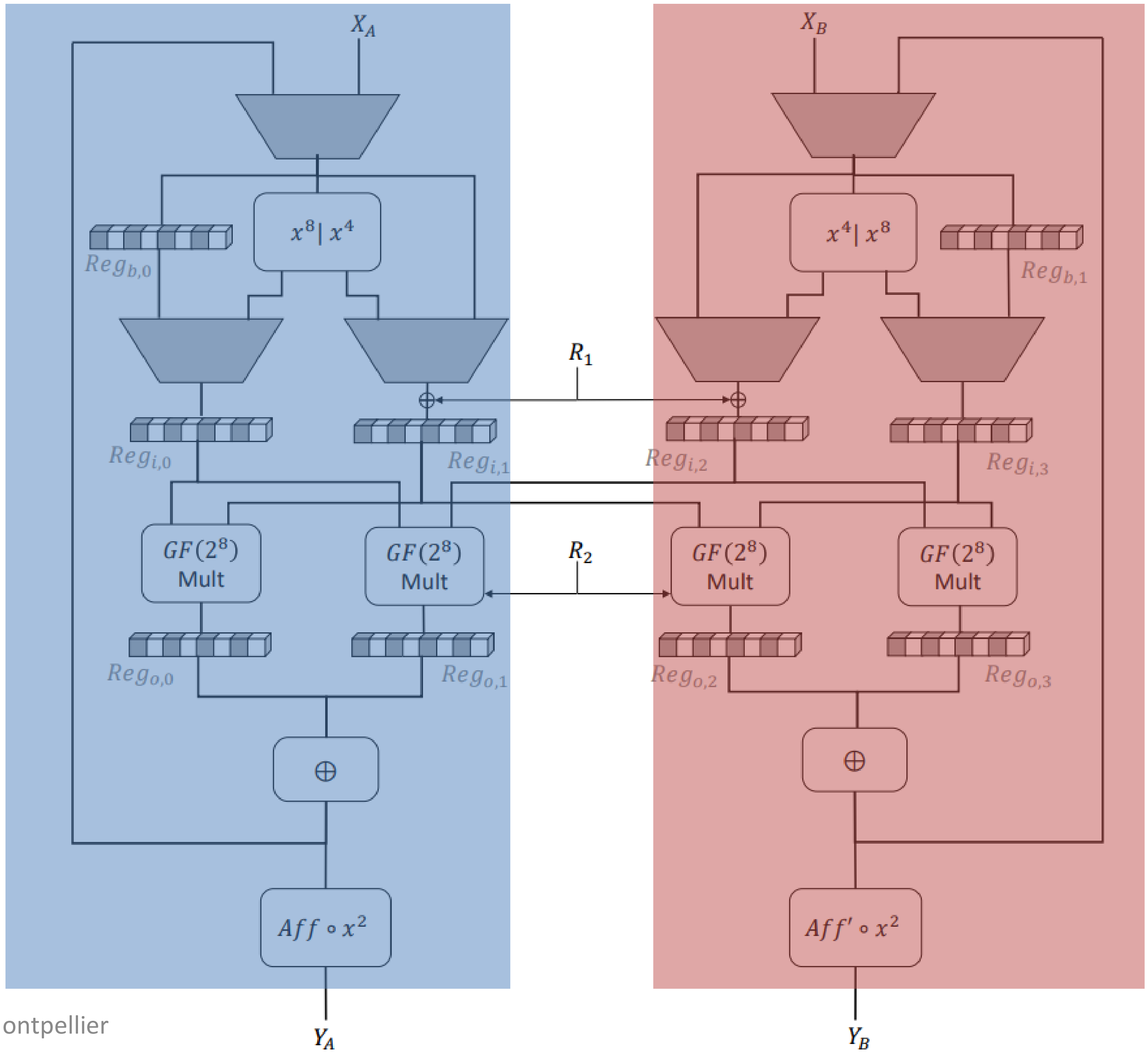


Preconditions:

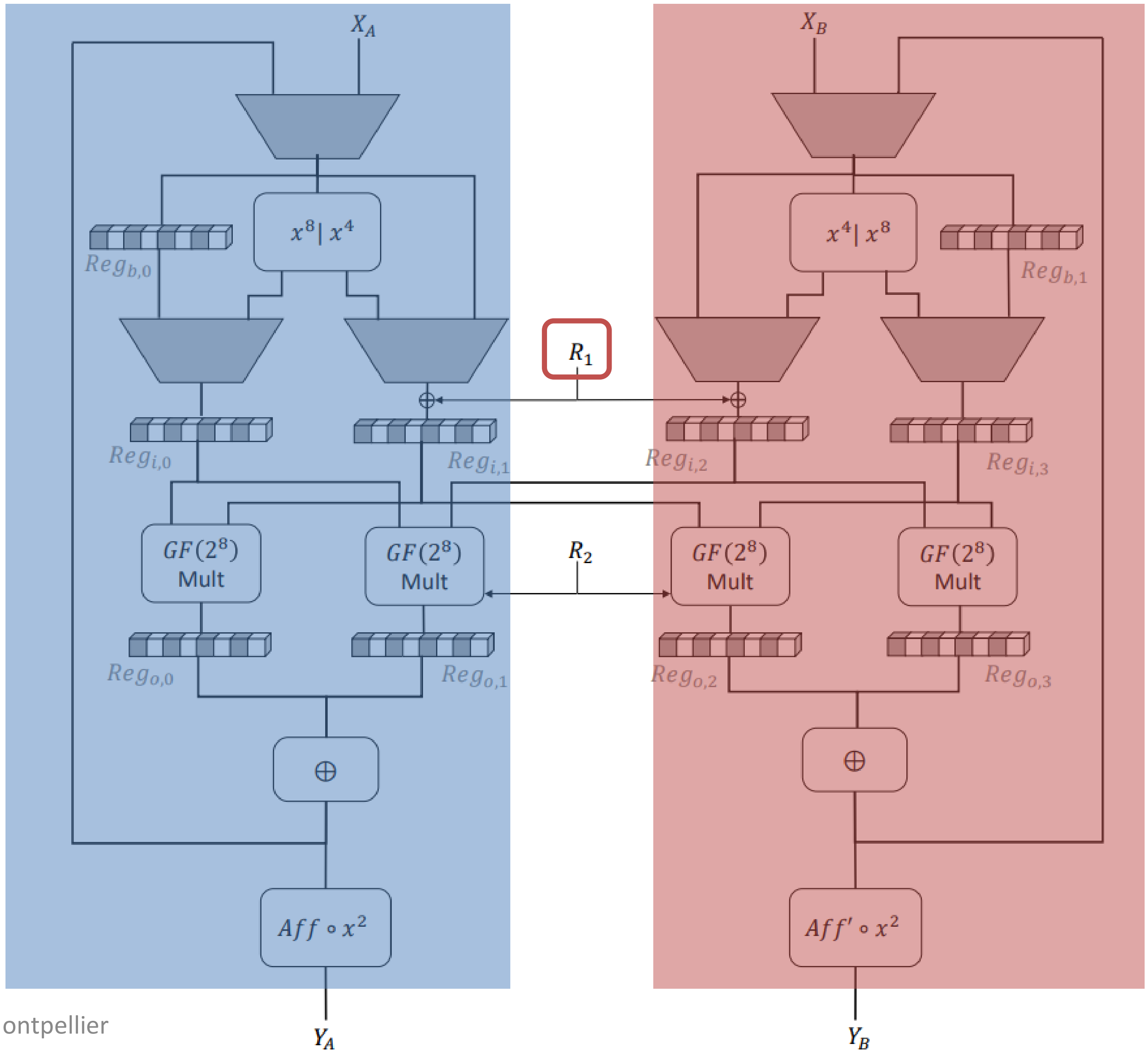
- X, Y are independently masked
- R provides n bits of randomness

Groß et al. *Domain-Oriented Masking: Compact Masked Hardware Implementations with Arbitrary Protection Order*, CCS 2016

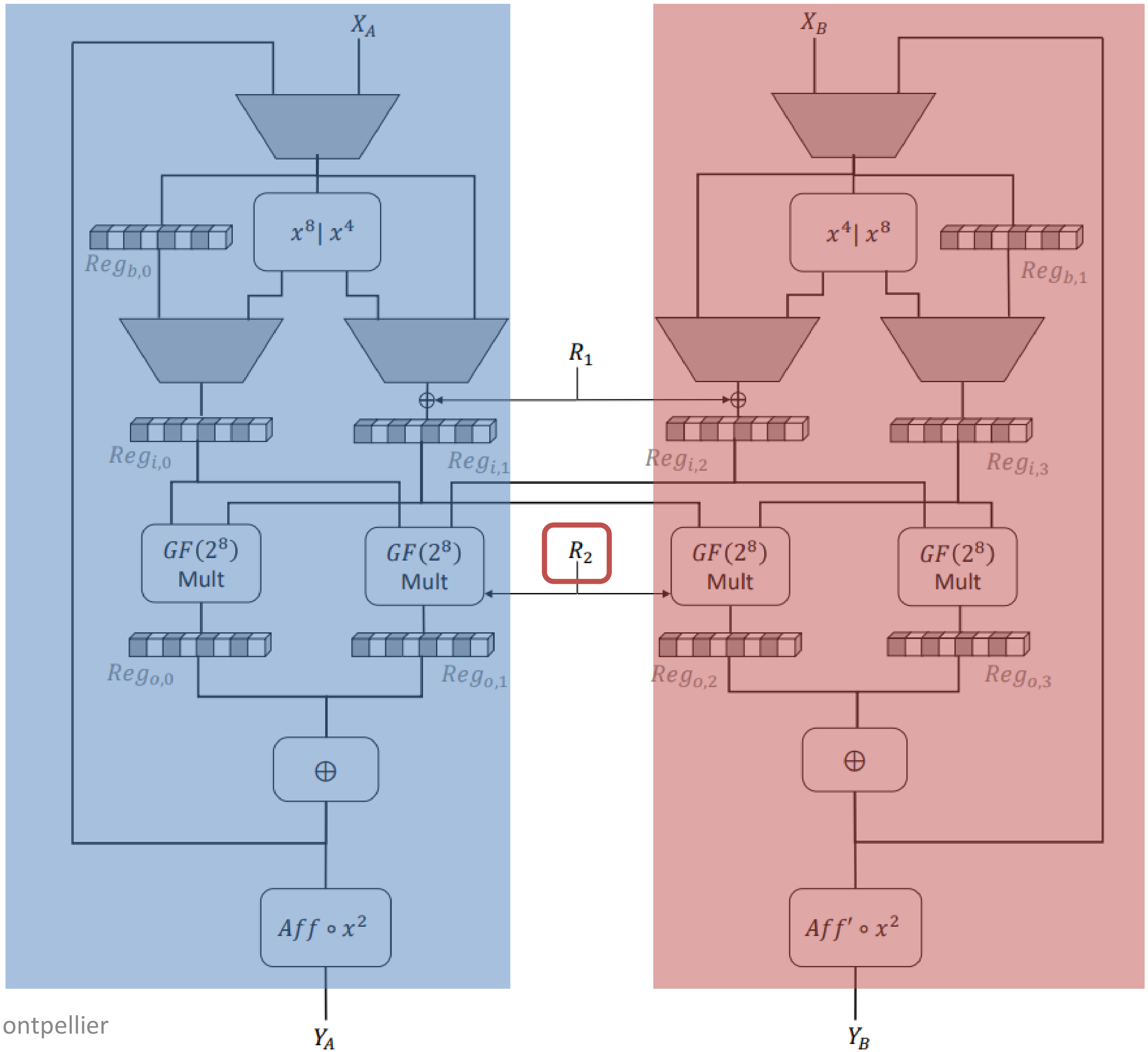
First-order Secure Design (Generic)



First-order Secure Design (Generic)



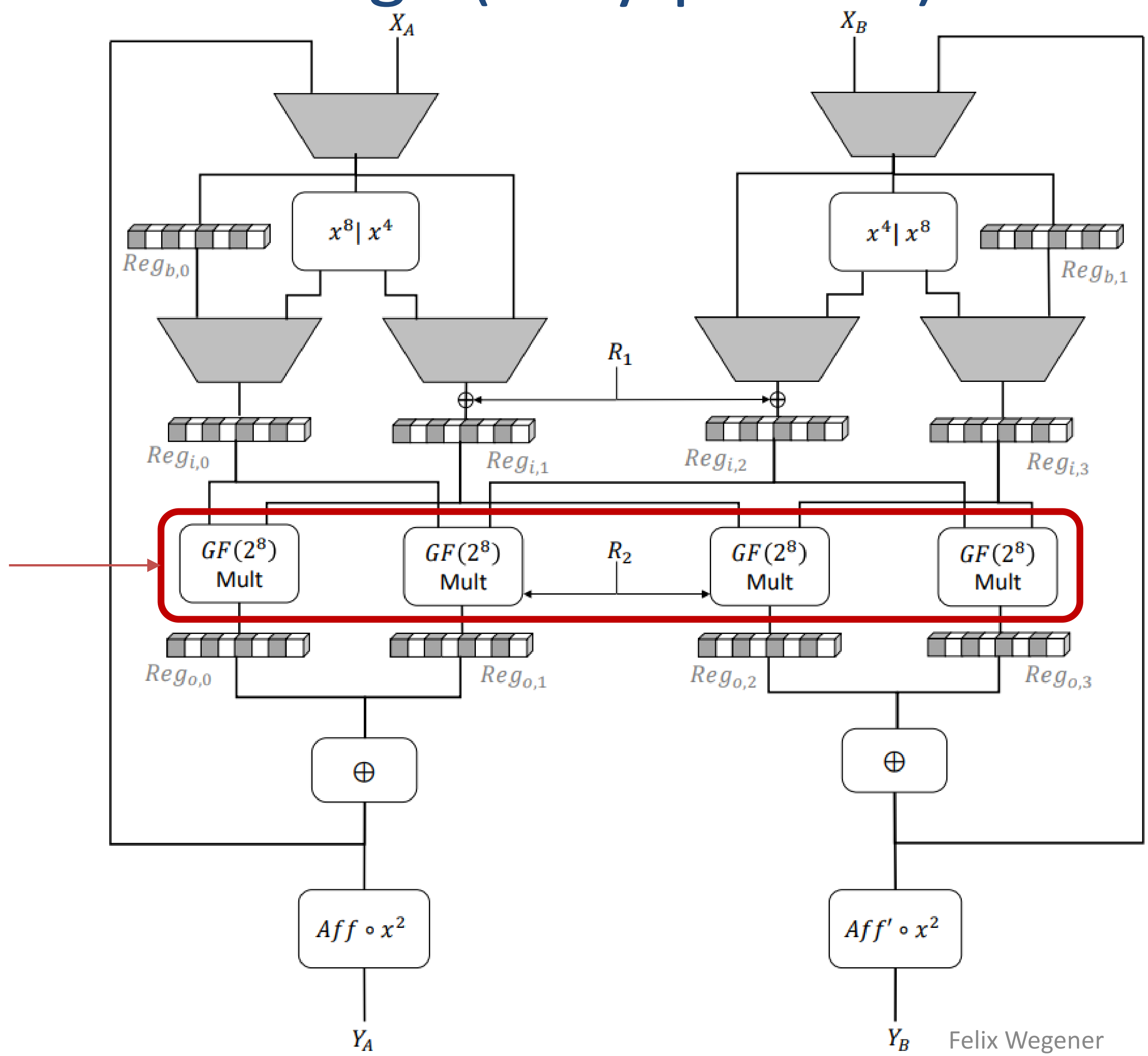
First-order Secure Design (Generic)



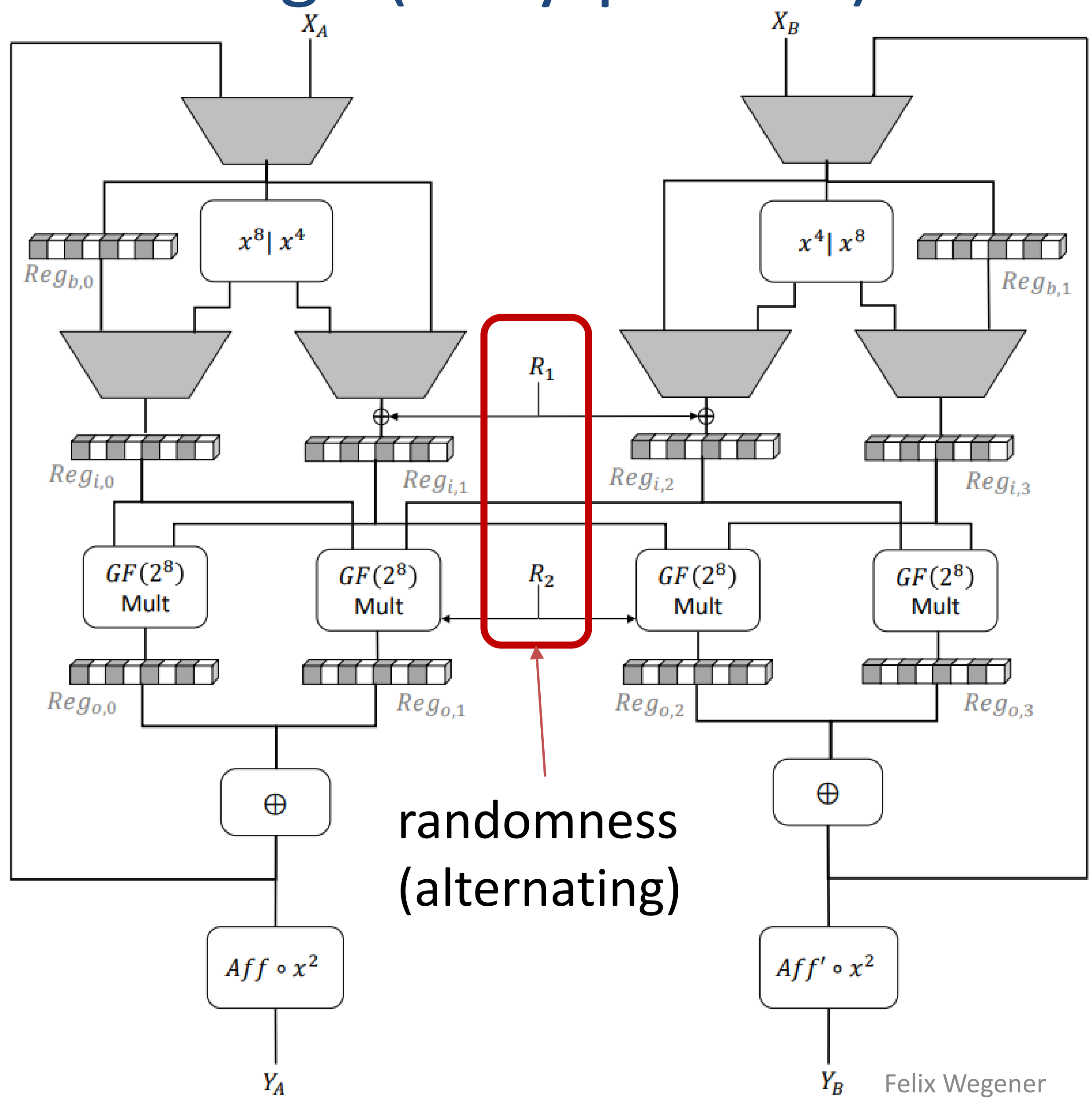
Design I: Fully-Parallel Multiplier

First-order Secure Design (Fully-parallel)

1 clock cycle



First-order Secure Design (Fully-parallel)



First-order Secure Design (Fully-parallel)

Latency:

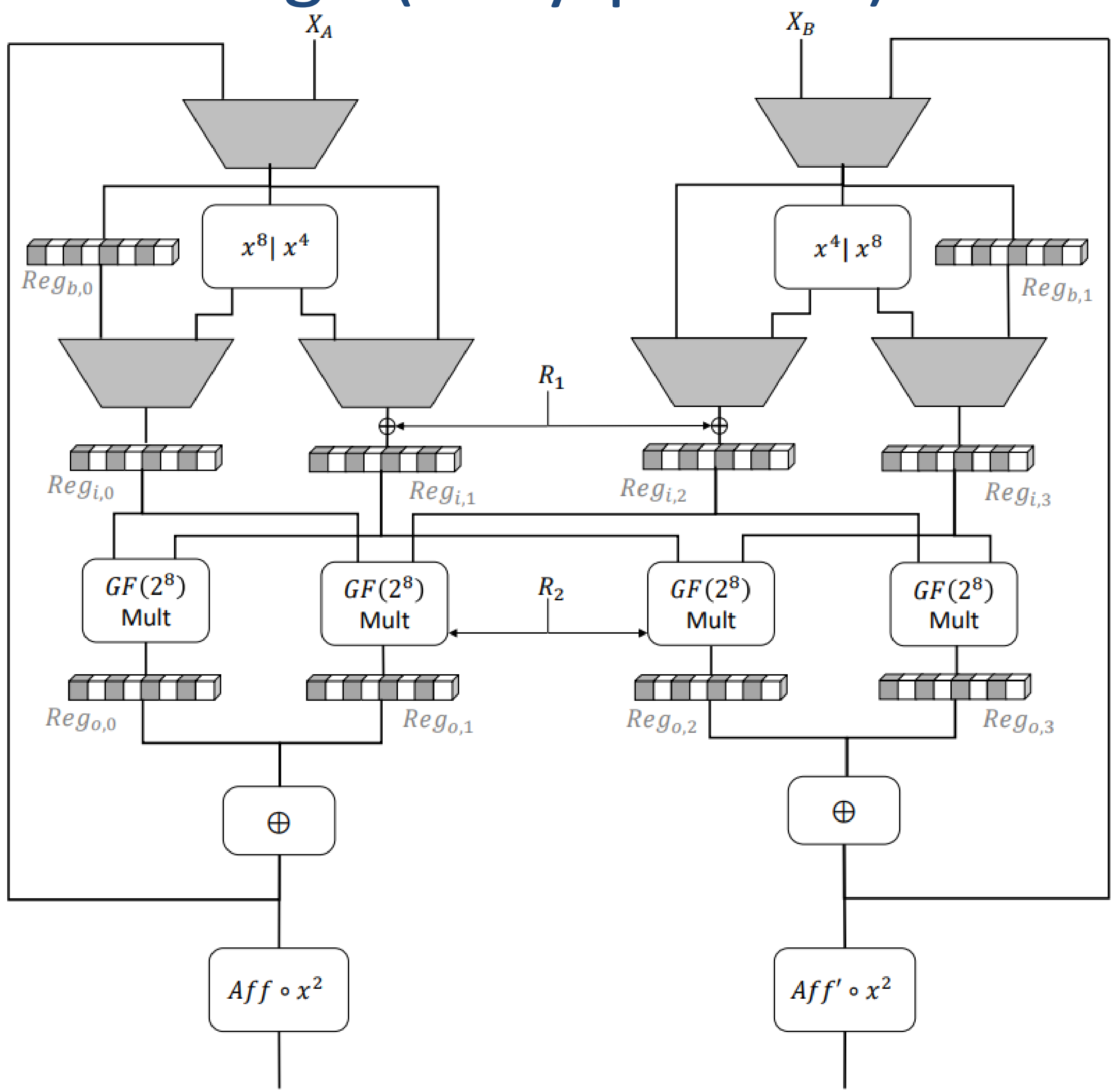
8 cycles

Randomness:

8 bits / cyc.

Area:

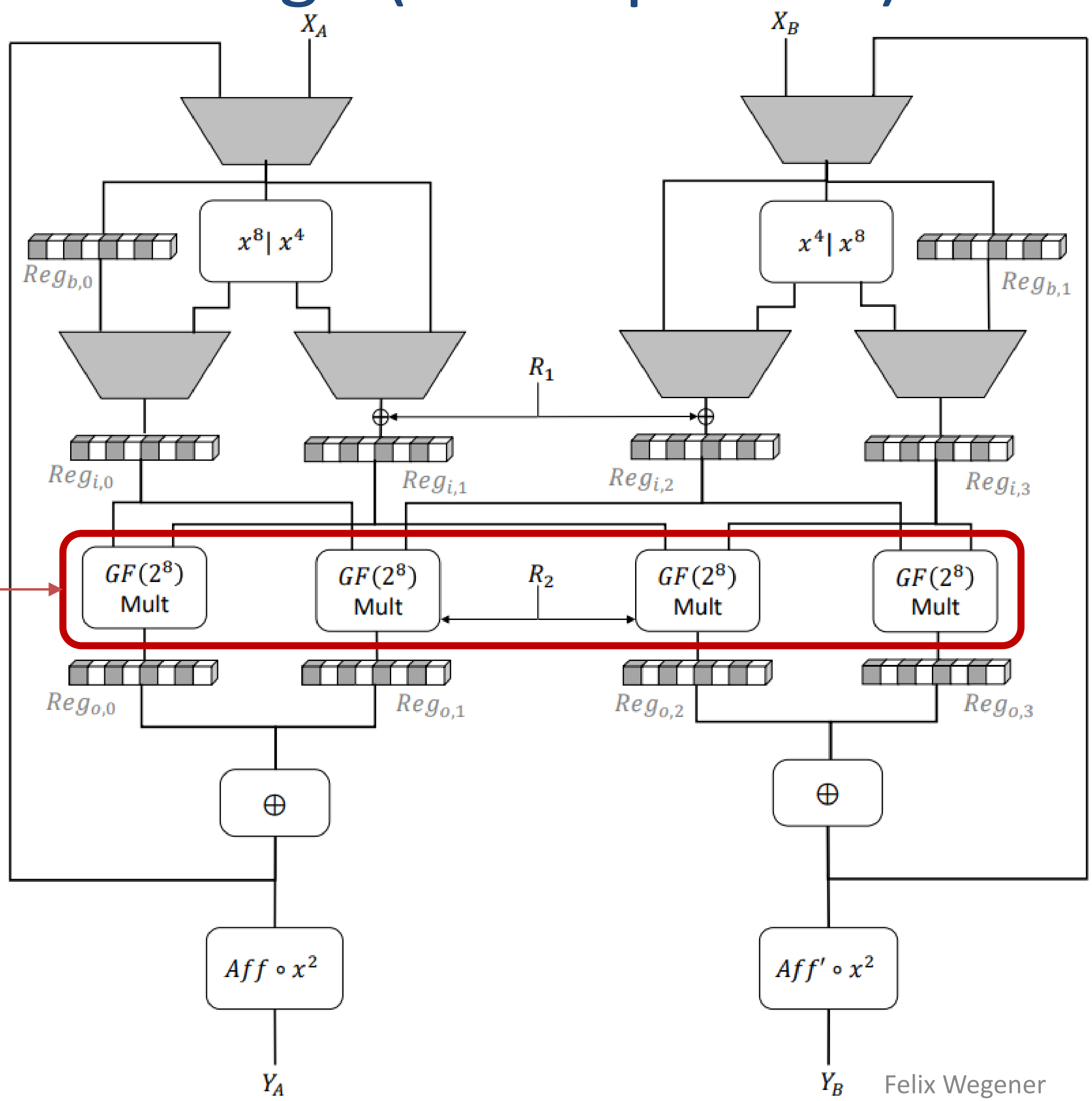
2321 GE



Design II: Serial-Parallel Multiplier

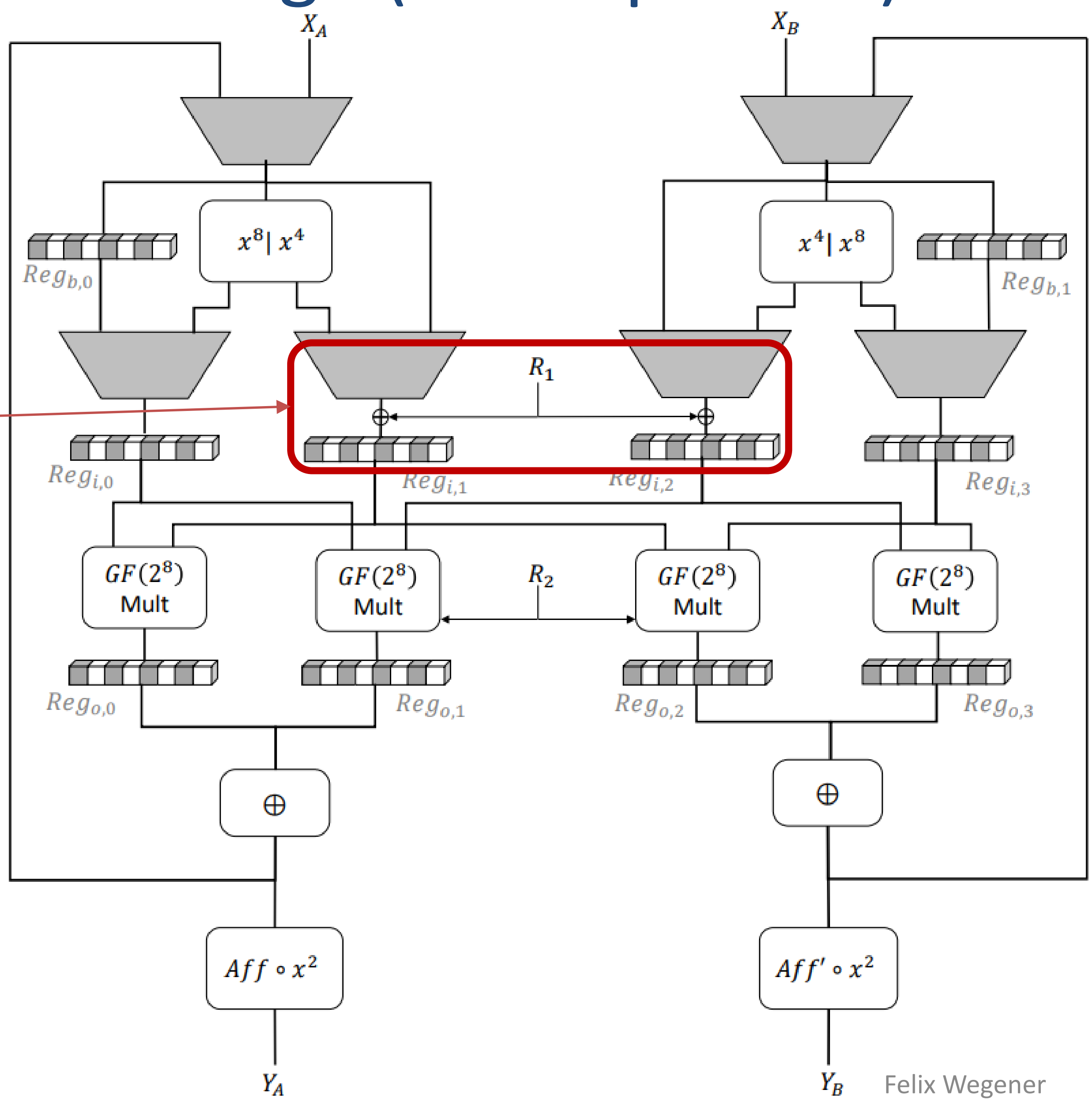
First-order Secure Design (Serial-parallel)

8 clock cycles



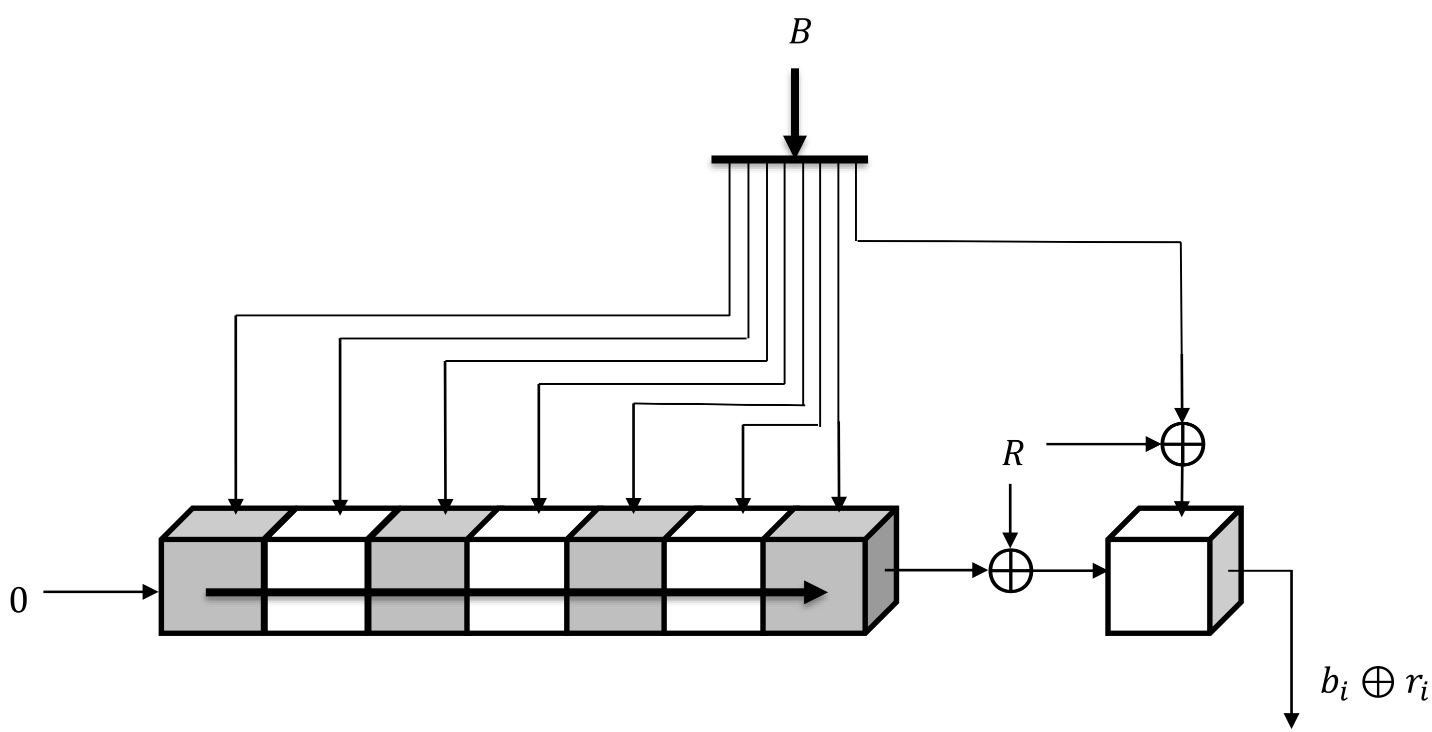
First-order Secure Design (Serial-parallel)

Restoring independence



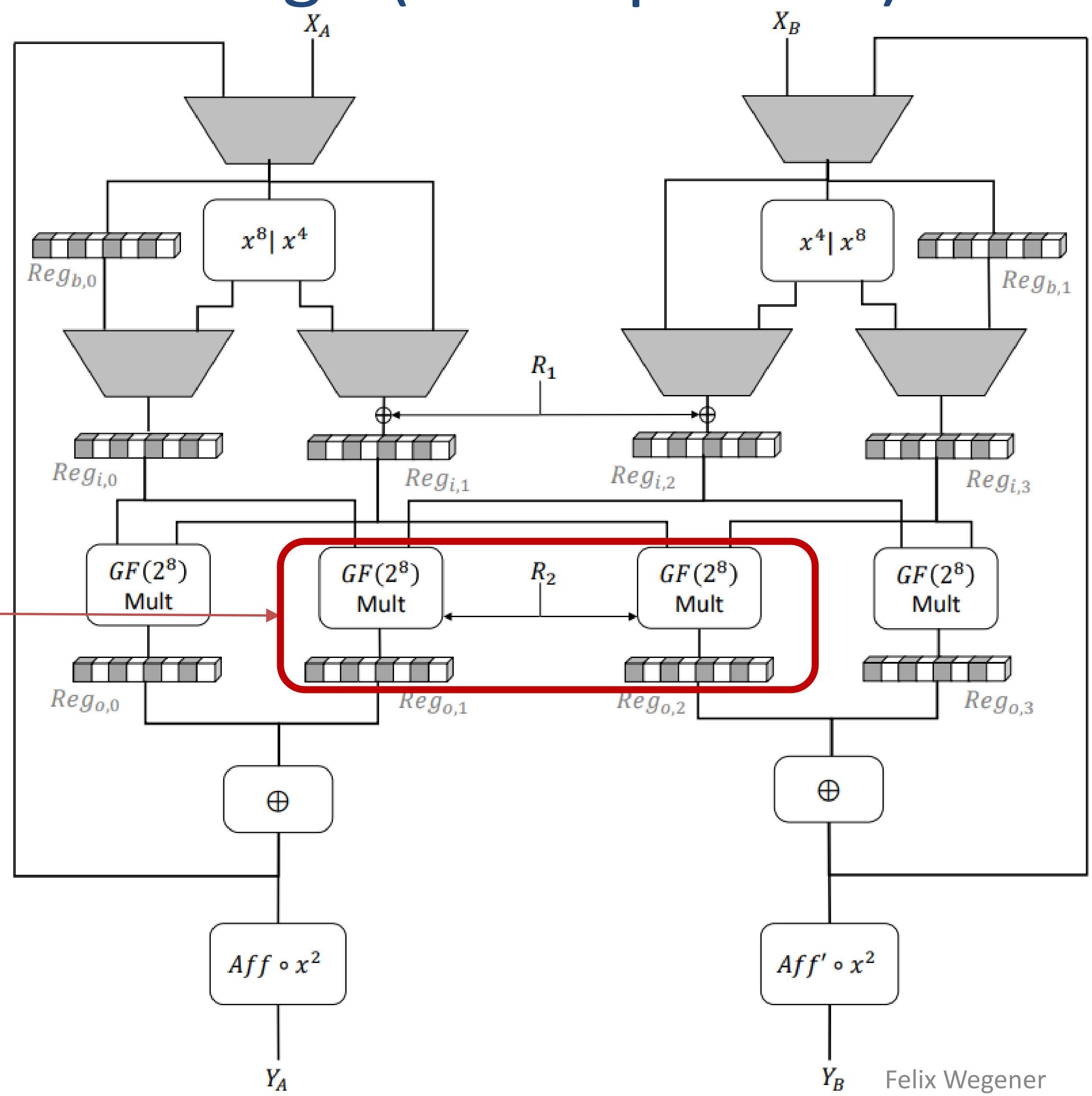
Restoring Independence

- Goal: 1 bit of randomness / cycle
  - Different path for MSB
  - Re-masked value from Register



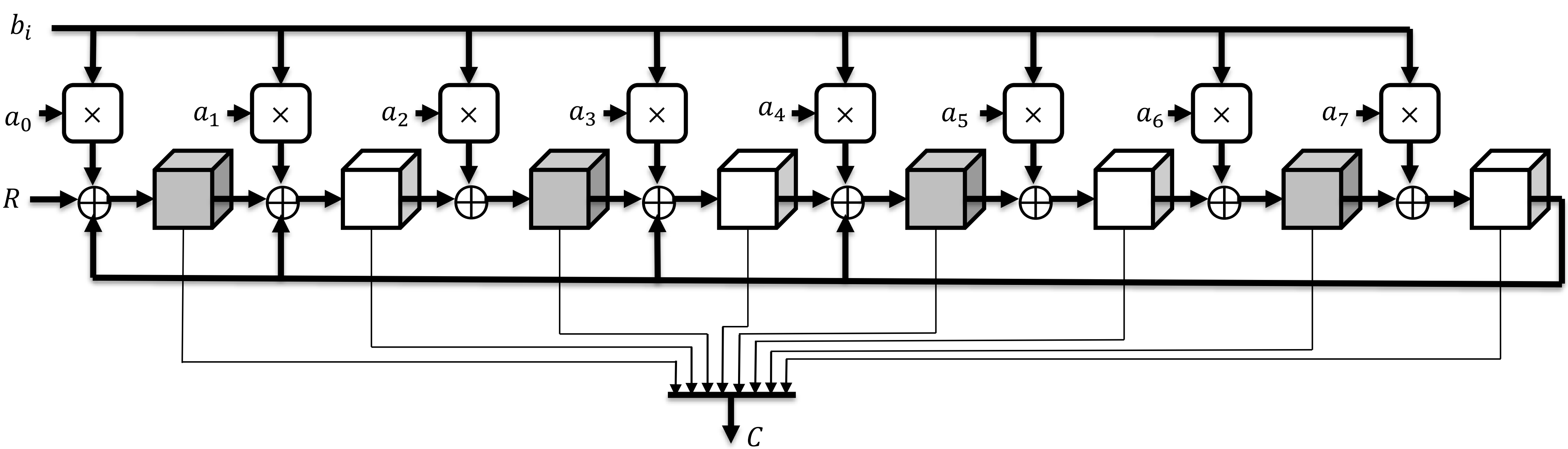
First-order Secure Design (Serial-parallel)

Cross-domain remasking



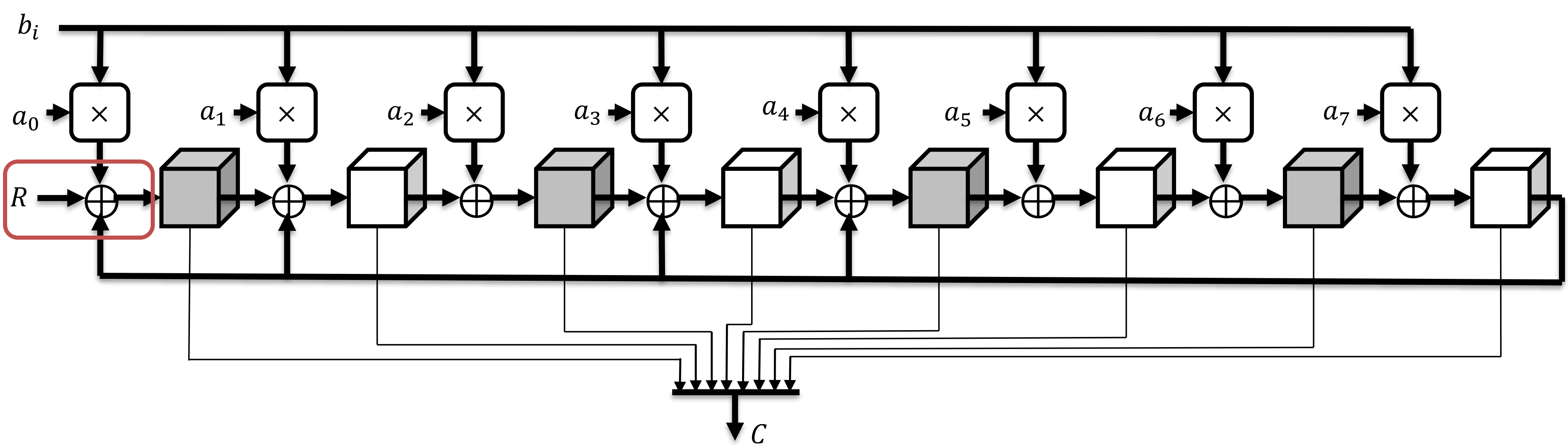
Serial-Parallel Multiplier

- Inputs:
  - a: 8 bits parallel
  - b: 1 bit serial



Serial-Parallel Multiplier

- Inputs:
  - a: 8 bits parallel
  - b: 1 bit serial
- Inject 1 random bit over 8 cycles

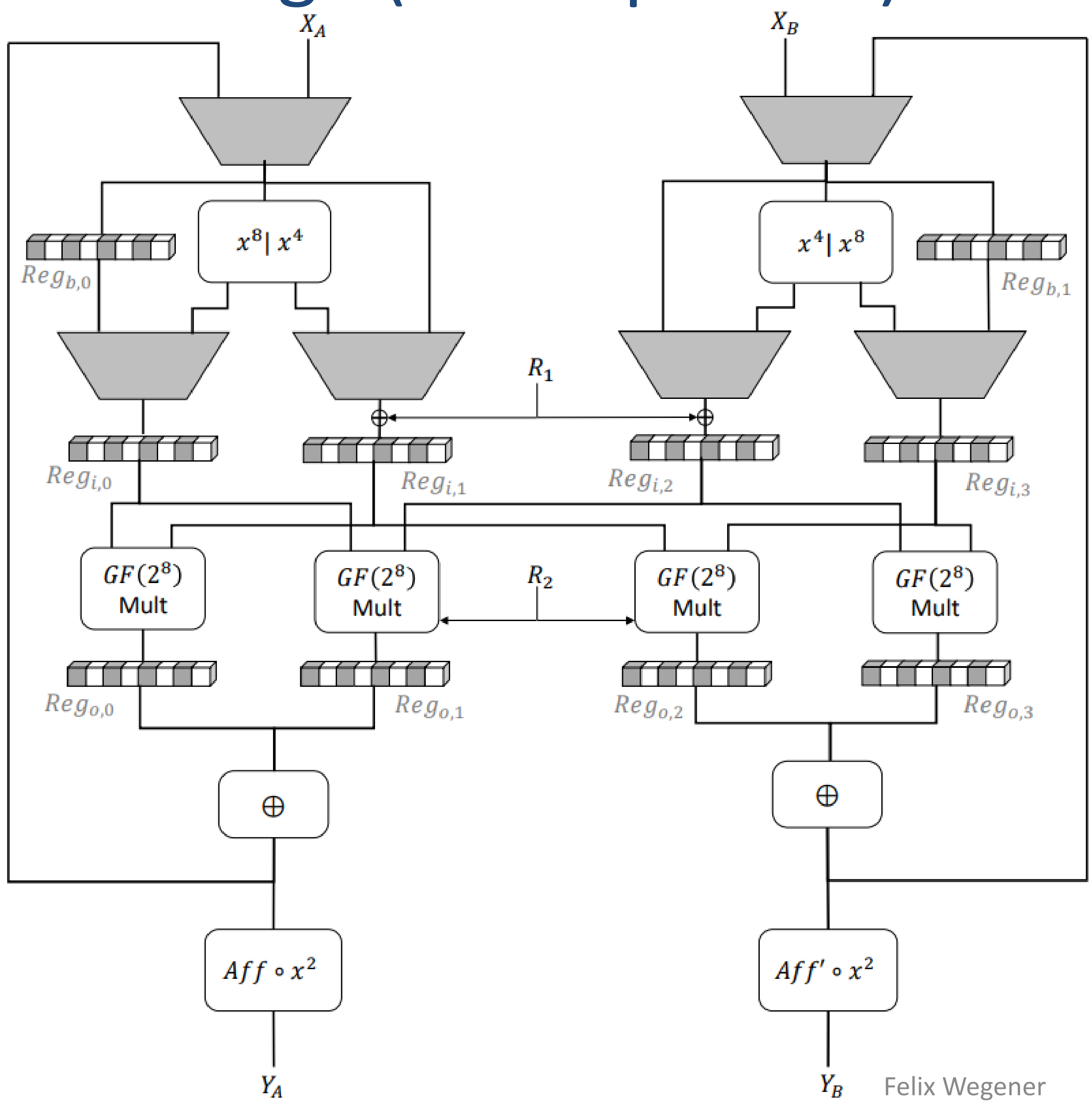


First-order Secure Design (Serial-parallel)

Latency:
36 cycles

Randomness:
2 bits / cyc.

Area:
1378 GE

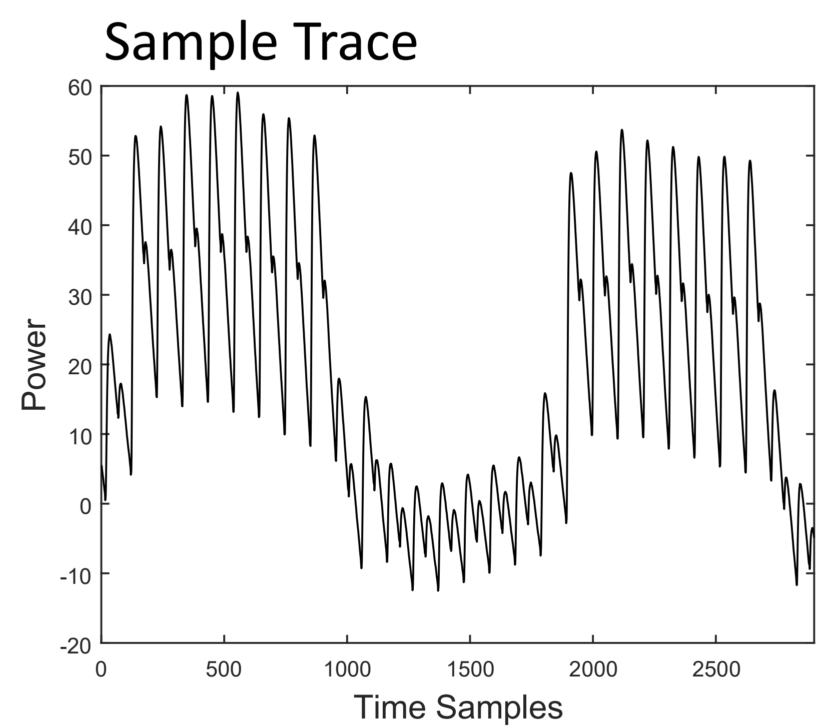


Side-Channel Evaluation

SCA Evaluation: Method and Setup

- MC-DPA evaluation
- Sequential execution of S-box
  - First: Derive Power Model
  - Second: CPA

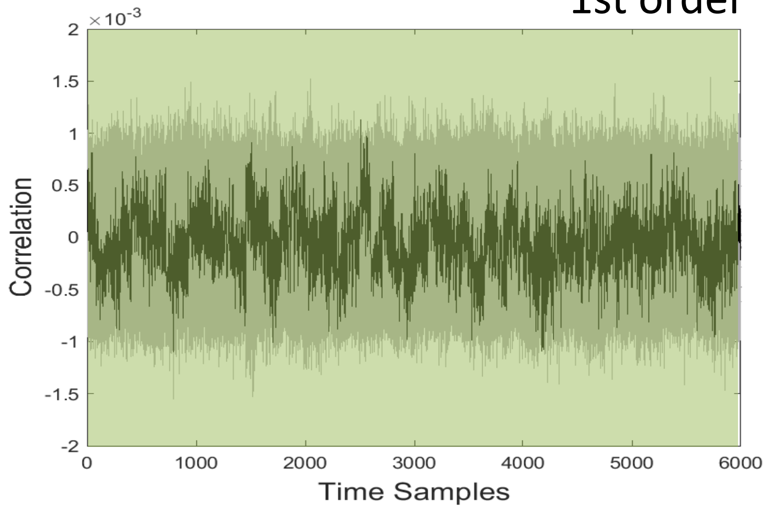
Setup:
- Sakura-G board @ 6Mhz
- Picoscope 6000 @ 625 MS/s
- No. traces: 10 million



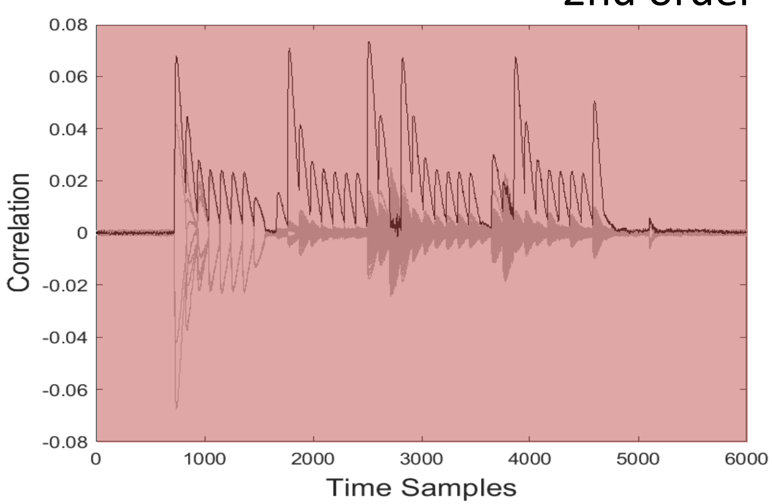
SCA Evaluation: Results

Serial Design

1st order

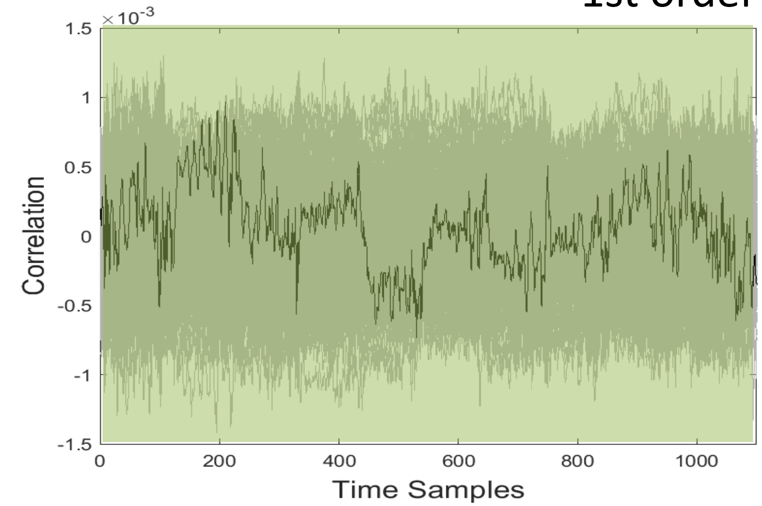


2nd order

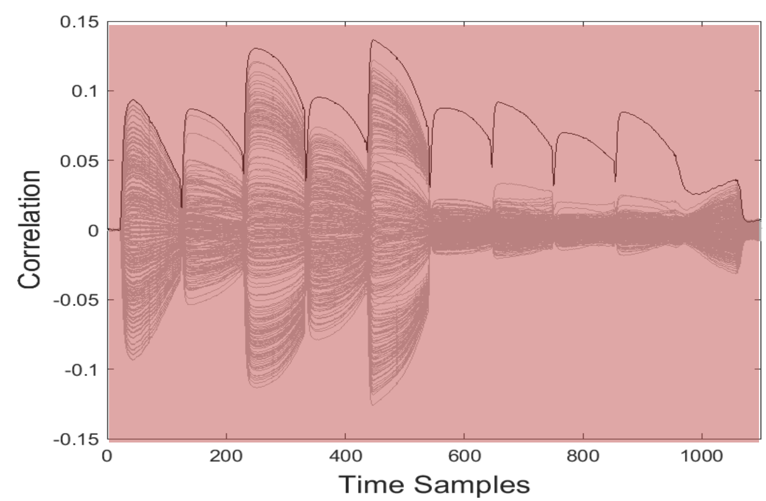


Parallel Design

1st order



2nd order



Comparison: Unprotected Designs

Design	Latency (cycles)	Crit. Path (ns)	Size (GE)
Boyar et al.	1	5.6	205
Serial Design (unprotected)	32	1.5	520

Comparison: Protected Designs

Design	Shares	Latency (cycles)	Crit. Path (ns)	Rand/Cyc (bits)	Size (GE)
Bilgin et al.	3	3	N/A	16	2224
Cnudde et al.	2	6	N/A	46	1872
Groß et al.	2	8	N/A	18	2600
Ueno et al.	2	5	1.5	56	1656
Former Work	4	16	3.3	0	4200
Parallel Design	**2**	**8**	**1.6**	**8**	**2321**
Serial Design	**2**	**36**	**1.5**	**2**	**1378**

Summary

- New first-order secure AES S-box designs:
  - Parallel Design: Interesting trade-off
  - Serial Design:
    - **Smallest** first-order secure AES S-box
    - Only 2 bits of randomness per cycle
- Methodology:

Smallest unprotected design

$$\not\Rightarrow$$

Smallest protected design

Thanks!
any questions?

felix.wegener@rub.de

Ruhr University Bochum, Horst Görtz Institute for IT-Security, Germany