13th April 2017, Paris, France
Constructive Side-Channel Analysis and Secure Design (COSADE)

Toward More Efficient DPA-Resistant AES Hardware Architecture Based on Threshold Implementation

Rei Ueno, Naofumi Homma, and Takafumi Aoki

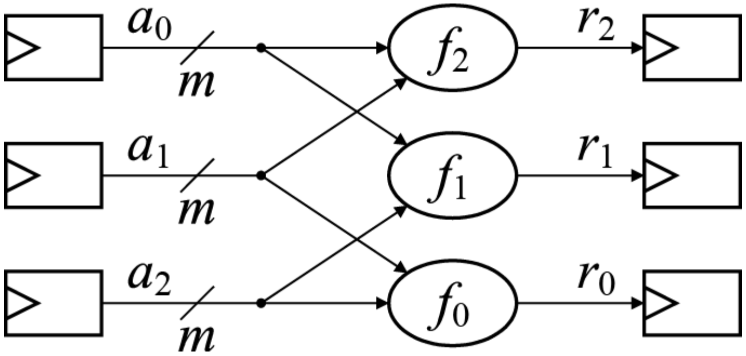
Tohoku University

Threshold Implementation (TI)

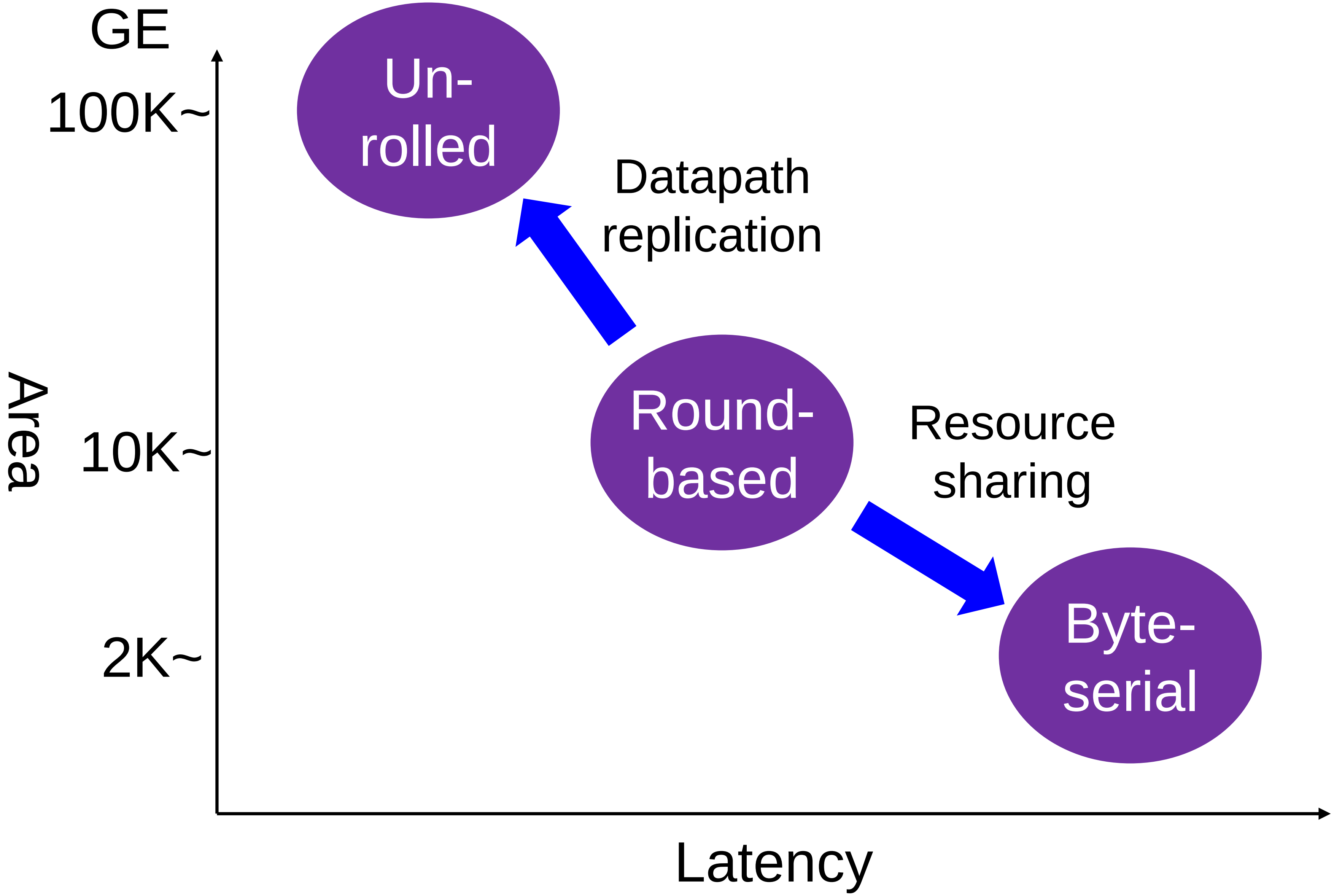
- Achieve provable security considering glitches
- Masking-based countermeasure

$$a = a_1 + a_2 + \cdots + a_i + \cdots + a_s$$

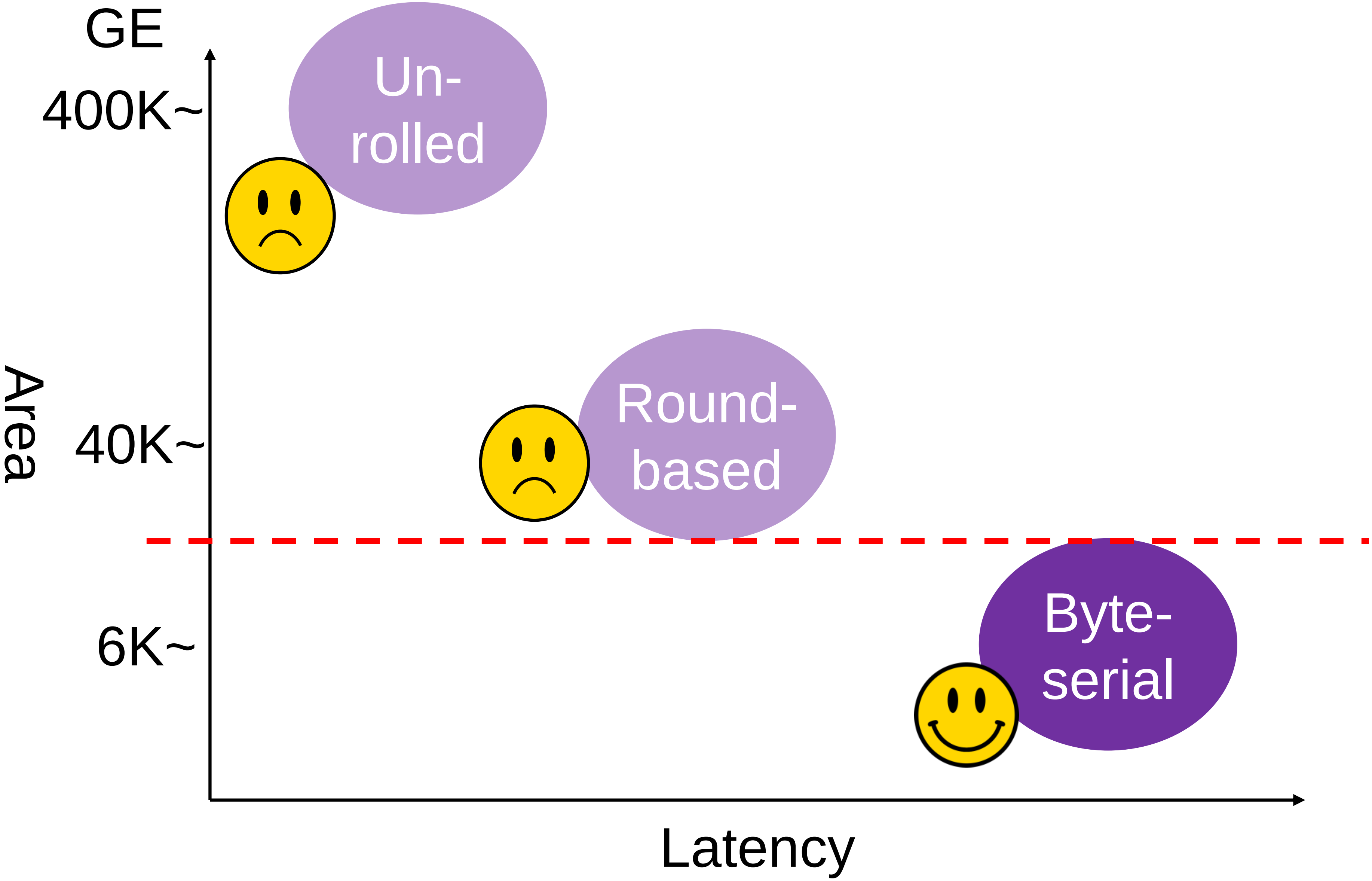
  - a: secret value, a_i: share
  - Exploits pipelining to avoid propagating glitches
  - dth-order TI defeats dth-order DPAs



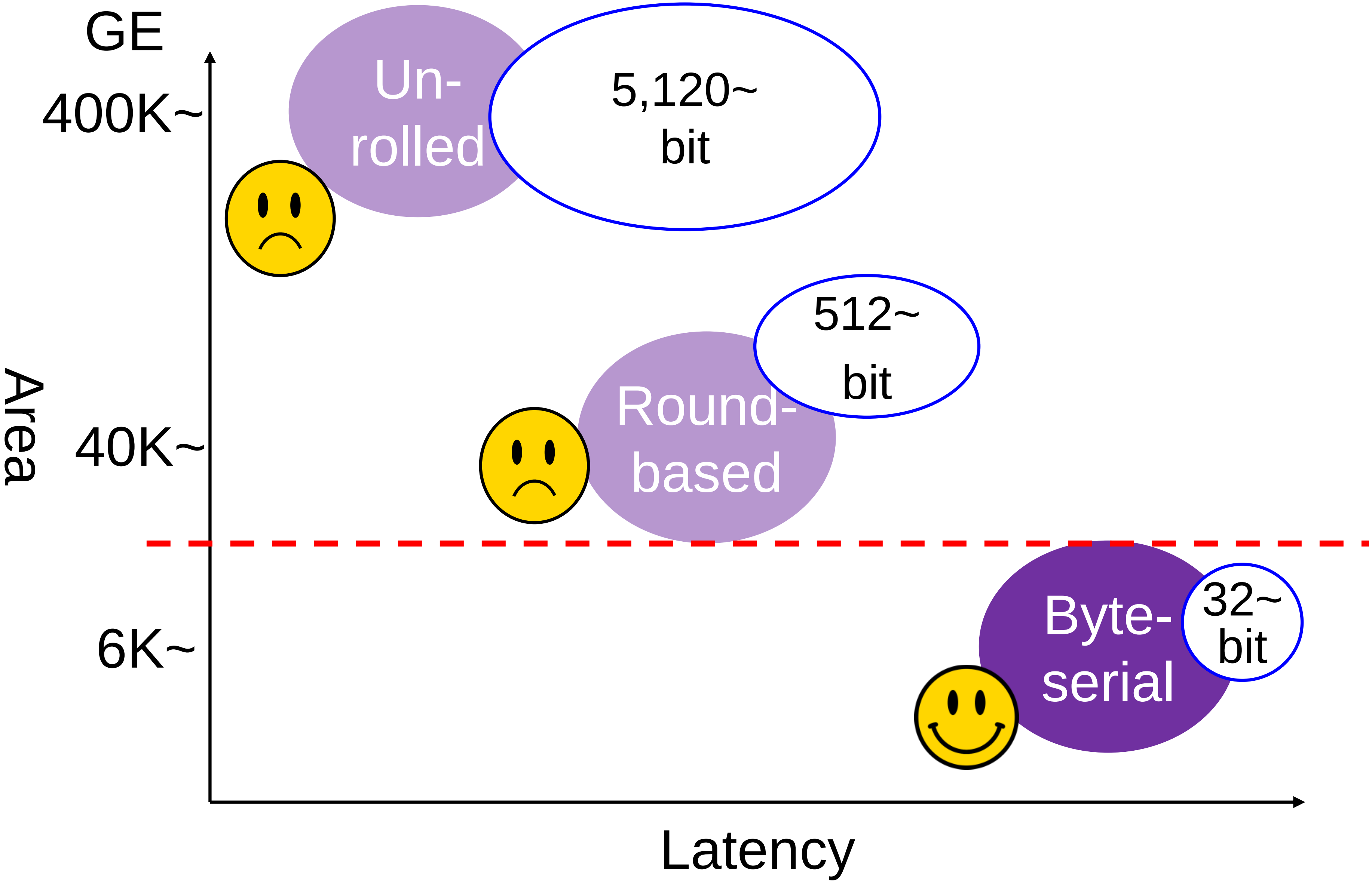
(Unprotected) AES hardware architectures



TI-based AES architectures



Randomness in TI-based AES architectures



This work

- New TI-based S-box
  - Combine algebraic characteristic of AES S-box with state-of-the-art TI construction ($d + 1$ input share TI)
  - Achieve 25% smaller area than conventional ones

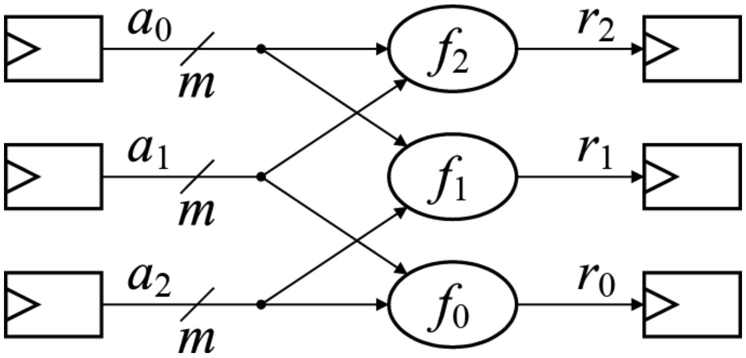
- Efficient byte-serial AES HW architecture for TI
  - Resister-retiming for low latency encryption
  - Achieve 11-21% lower latency without area overhead

Outline

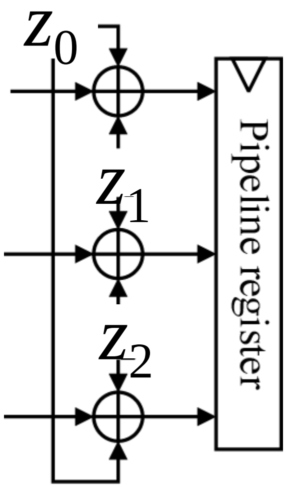
- Introduction
- TI-based AES S-box
- AES HW architecture for TI
- Experimental leakage evaluation
- Concluding remarks

Conditions for *d*th-order TI

- Correctness
- *d*th-order non-completeness
- Uniformity (or mask refreshing)



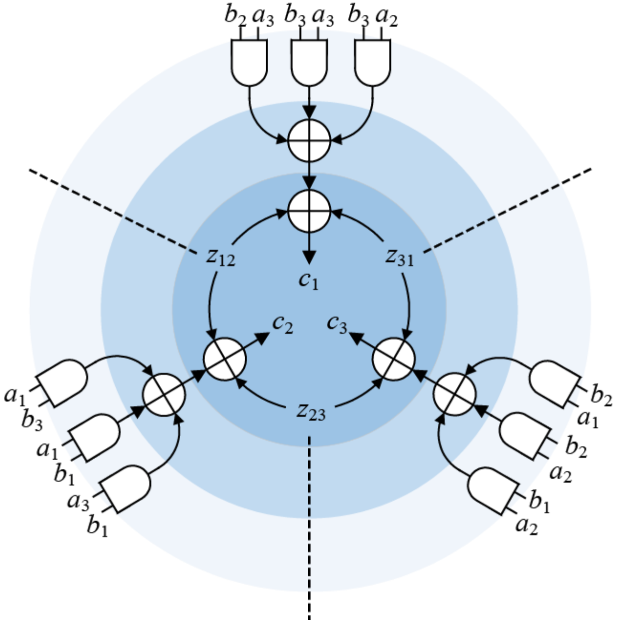
First-order
non-complete circuit

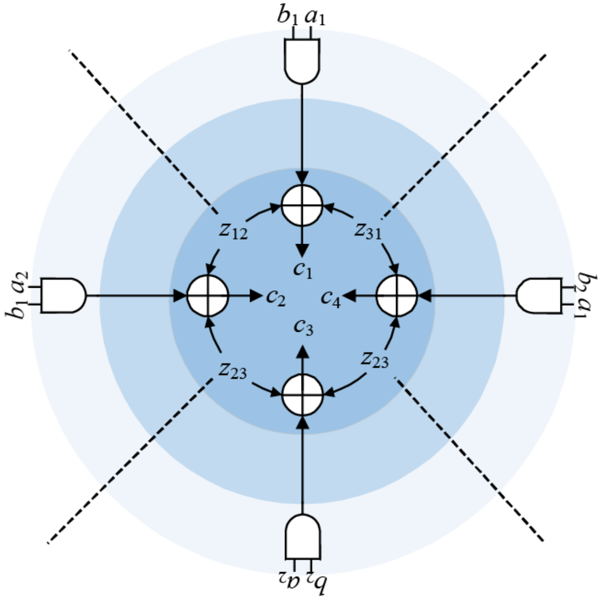


Ring-refreshing

Two constructions of TI-based circuit

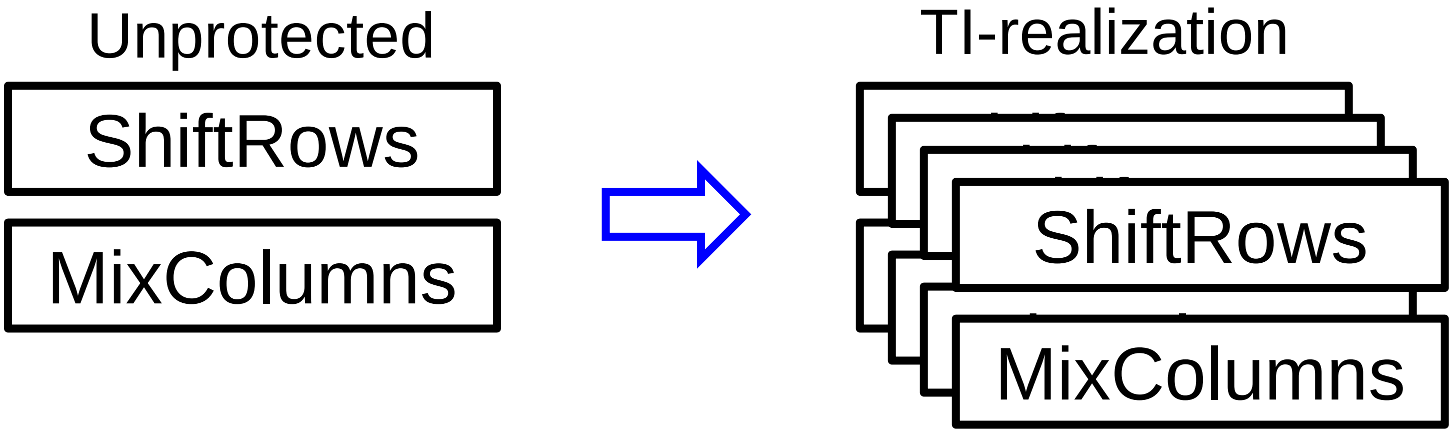
- TI with $td + 1$ input shares (t : algebraic degree)
  - Less registers and randomness
  - Efficiently applied to some practical 4-bit S-boxes
- TI with $d + 1$ input shares
  - Smaller area, but more registers and randomness
  - Input shares must be independent of each other





TI-based AES

- Linear functions are easily realized



Unprotected

ShiftRows

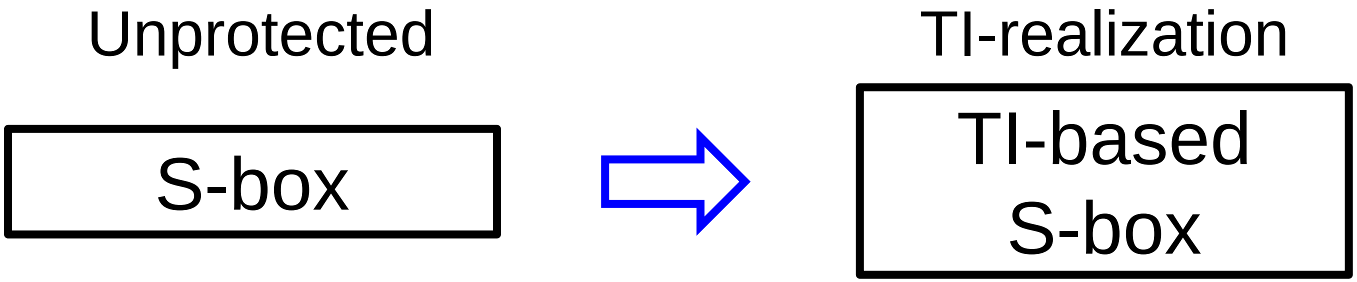
MixColumns

TI-realization

ShiftRows

MixColumns

- For non-linear function (i.e., S-box)?
  - Inversion determines security-order and performance



Unprotected

S-box

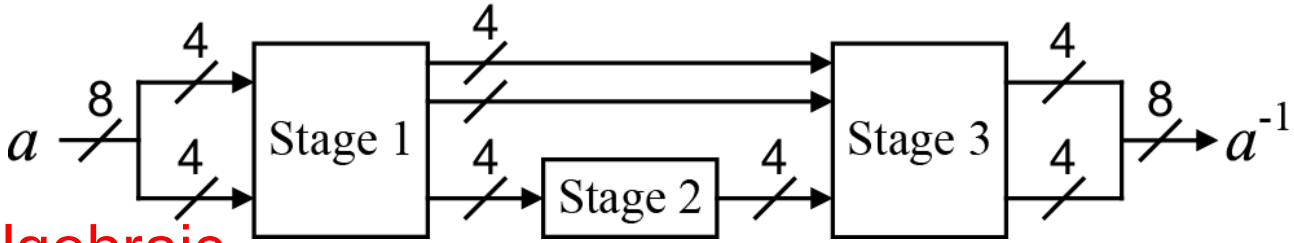
TI-realization

TI-based S-box

TI-based inversion circuits

- Efficiently decomposed into three stages based on tower-field arithmetic

Tower-field inversion



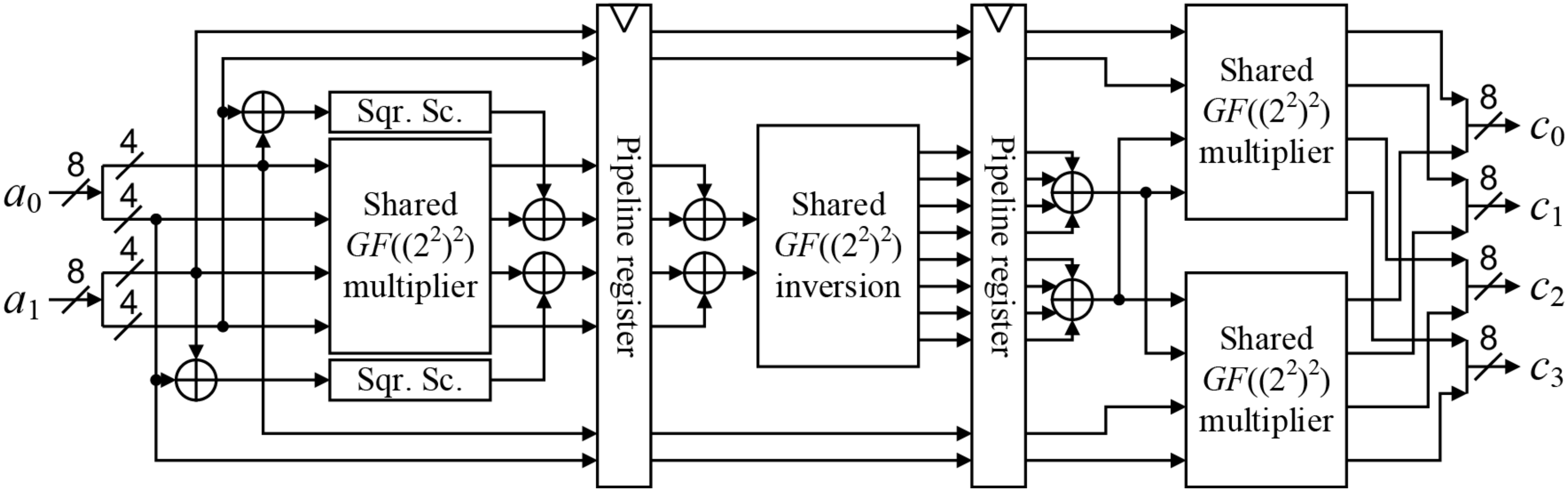
Algebraic degree 2 3 2

- Three designs w.r.t. Stage 2 and TI constructions

	Moradi et al., Eurocrypt 2011	Bilgin et al., TCAD 2015	Cnudde et al., CHES 2016
Stage 2	2-stage pipelined	Non-pipelined	2-stage pipelined
Input shares	$td + 1$	$td + 1$	$d + 1$
Area (GE)	4,244	2,224	1,872

Proposed TI-based inversion

- TI with $d + 1$ input shares
- Non-pipelined Stage 2



a_0
a_1
8
8
4
4
4
4
Sqr. Sc.
Shared $GF((2^2)^2)$ multiplier
Sqr. Sc.
Pipeline register
Shared $GF((2^2)^2)$ inversion
Pipeline register
Shared $GF((2^2)^2)$ multiplier
Shared $GF((2^2)^2)$ multiplier
8
8
8
8
c_0
c_1
c_2
c_3

- Non-pipelined Stage 2 can be efficiently optimized using OR (NOR) gates and factoring
  - a xor b xor (a and b) = a or b
  - Difficult to be applied to pipelined Stage 2

First-order TI-based S-boxes

- Logic synthesis with area optimization
  - Tool: Synopsys Design Compiler
  - Technology: TSMC 65 nm standard CMOS

Synthesis results

	Moradi+, Eurocrypt 2011	Bilgin+, TCAD 2015	Cnudde+, CHES 2016	This work
Area (GE)	4,244	2,224	1,872	1,342
Latency	5	4	6	5
Area-Latency product	21,220	8,896	11,232	6,710
Randomness [bit]	44	32	56	64

- 25% more compact and efficient

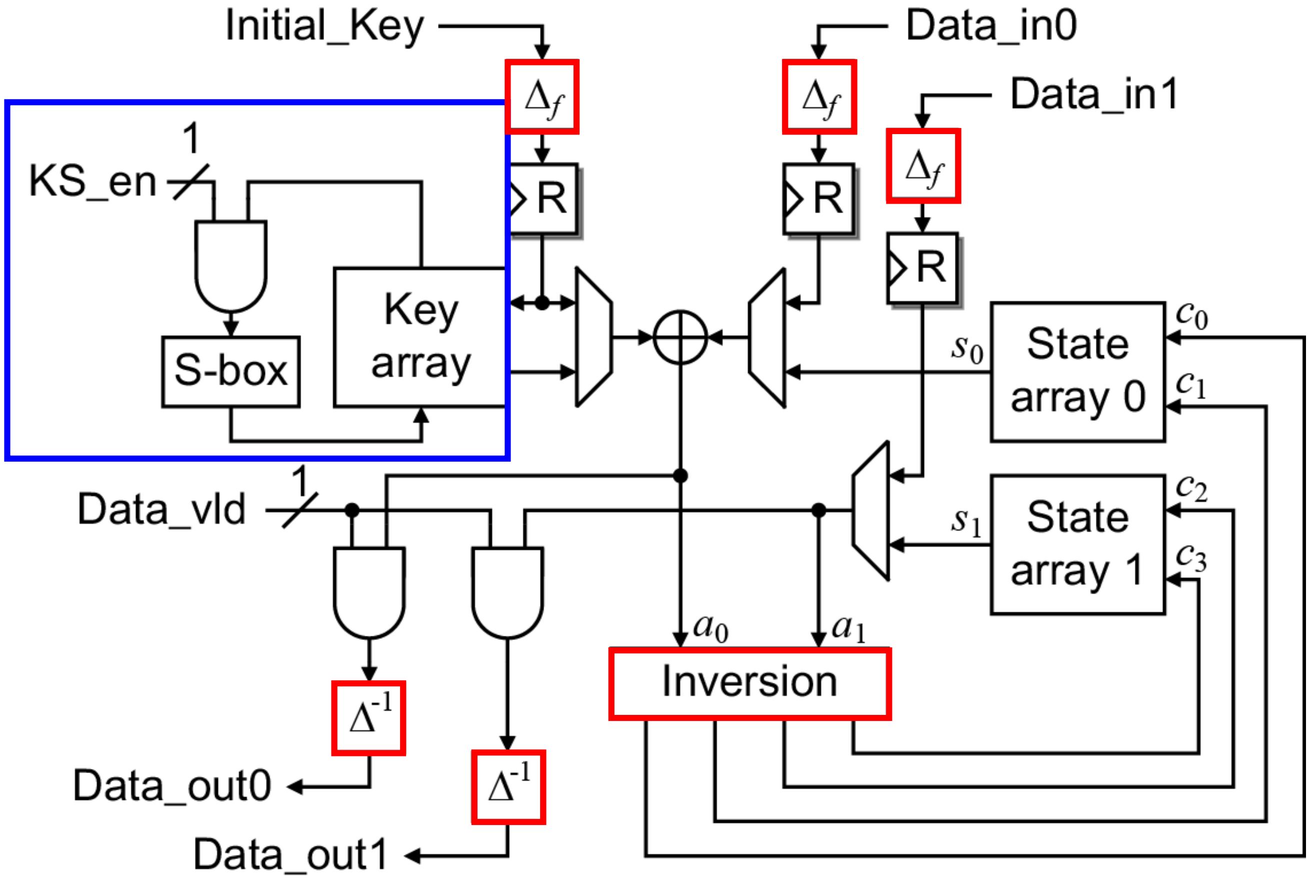
Outline

- Introduction
- TI-based AES S-box
- AES HW architecture for TI
- Experimental leakage evaluation
- Concluding remarks

Criteria

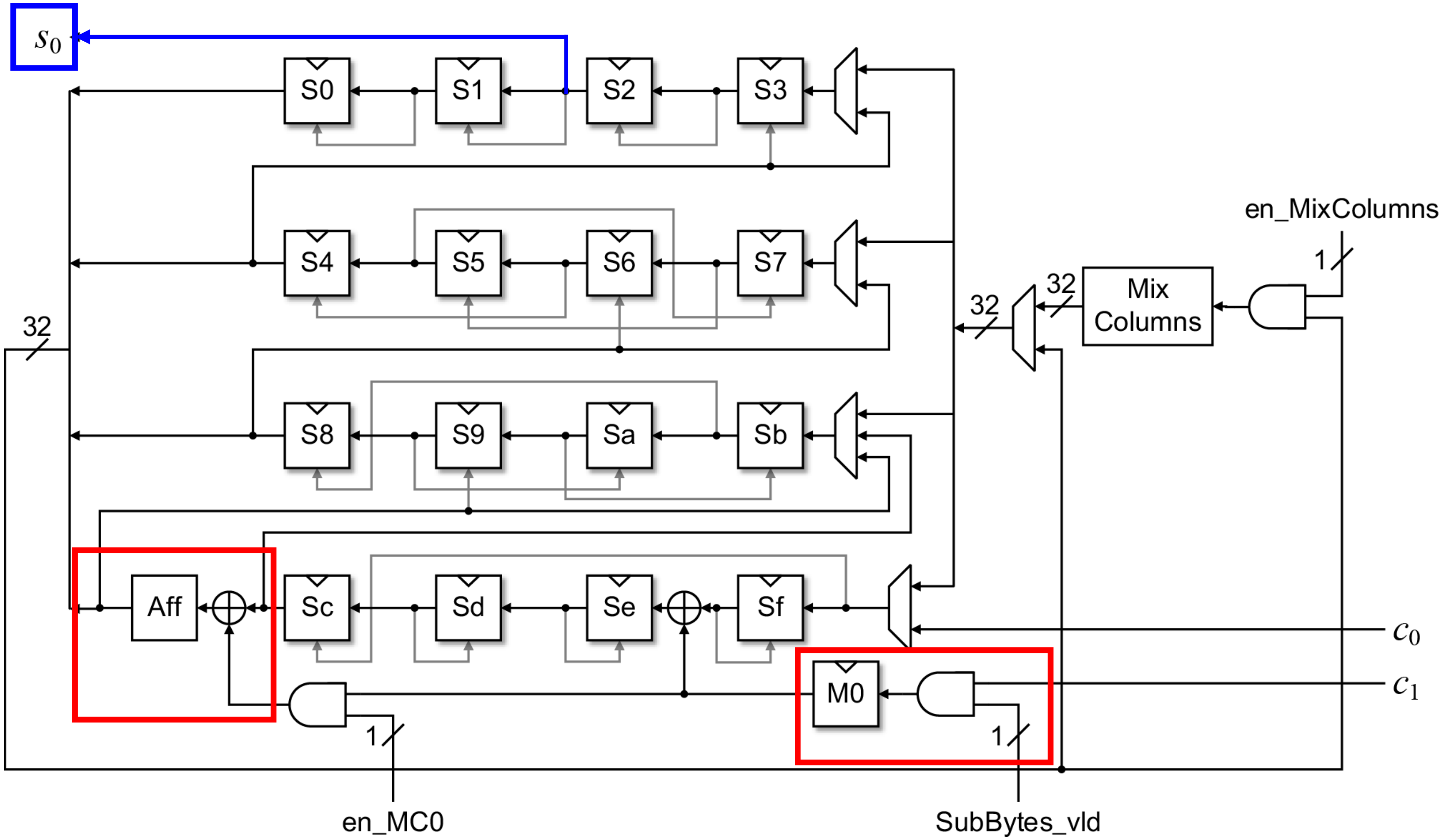
- Only one TI-based inversion
- 20 clock cycles per round
  - 16 clocks for SubBytes
  - 4 clocks for SubWords in key scheduling
- Latency caused by pipelining should have impact on only first round
  - Exploit parallelism
  - With few additional modules and few path selectors

Proposed AES HW architecture



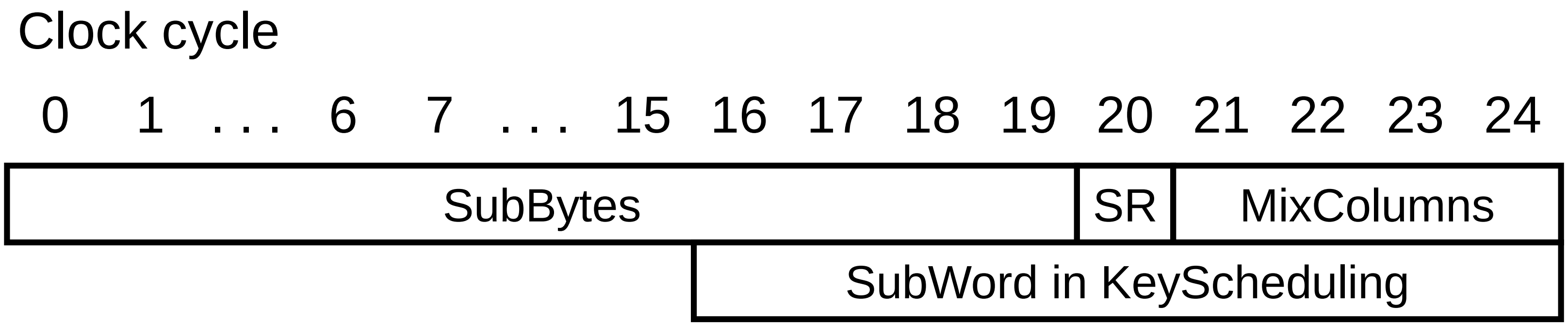
- Separated Inversion, Isomorphism, and Affine
- TI is not applied to key scheduler

State Array



- Register-retiming for parallel computation
  - Last four affine transformations after ShiftRows
  - Output to TI-based inversion during MixColumns

Timing diagram



Clock cycle

0 1 . . . 6 7 . . . 15 16 17 18 19 20 21 22 23 24

SubBytes

SR

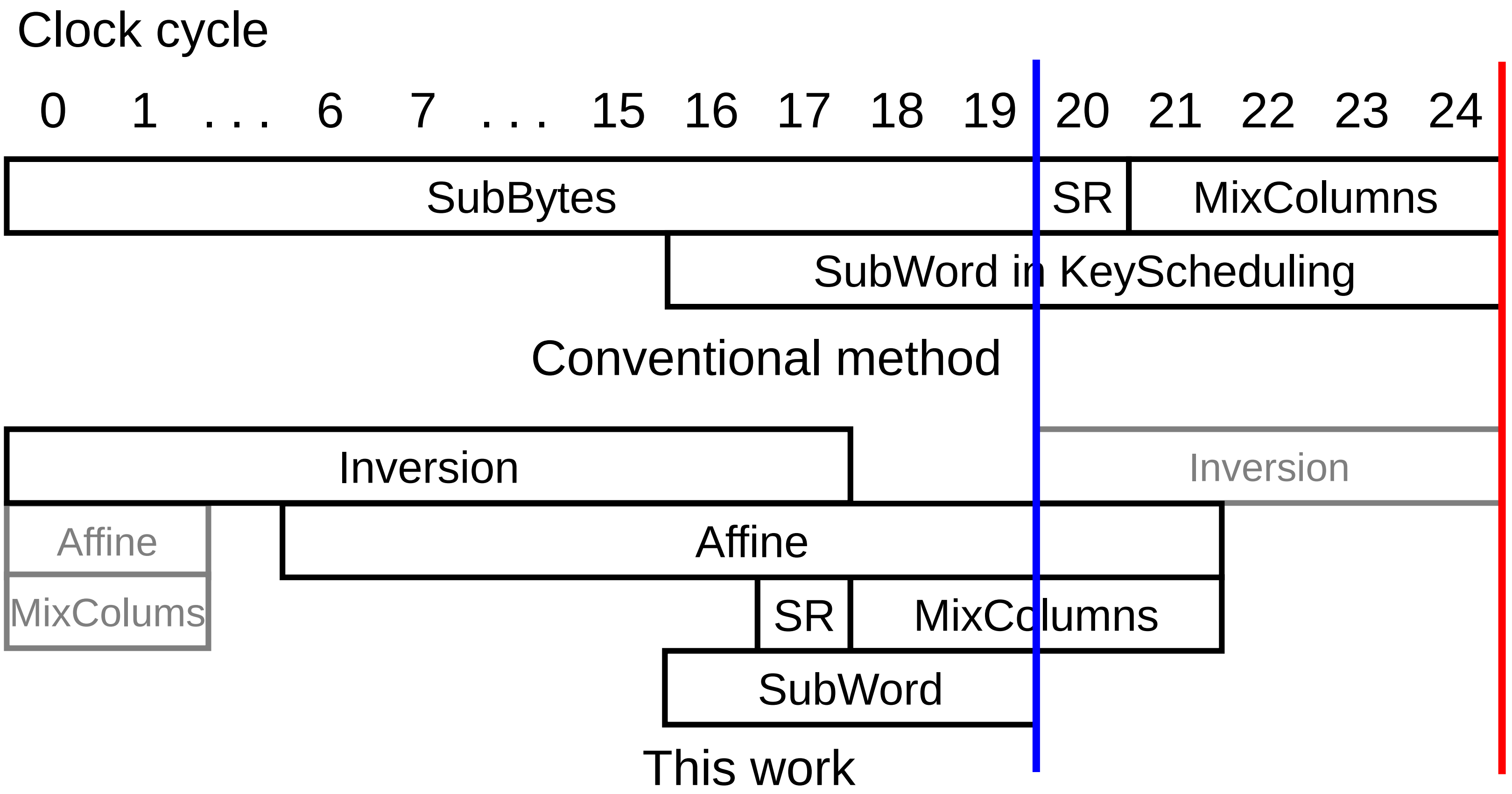
MixColumns

SubWord in KeyScheduling

Conventional method

- 20 (16 + pipeline-latency) clocks for SubBytes
- Distinct clocks for ShiftRows (SR) and MixColumns

Timing diagram



- 20 clock cycles per round instead of 25
  - Decompose SubBytes into Inversion and Affine
  - Parallel execution of SR, Inversion, Affine, MixColumns

Performance evaluation

	Moradi+, Eurocrypt 2011	Bilgin+, TCAD 2015	Cnudde+, CHES 2016	This work
Area [GE]	11,114	8,119	6,681	6,334
Latency	266	246	276	219
Power [uW]	24.12 (3.14*)	No data		3.06
Area-Latency product	2,956 K	1,997 K	1,844 K	1,387 K
Power-Latency product	6,416.92 (835.24*)	No data		672.33
Process [nm]	180			65

* Multiplied by square of process rate

- 11-21% lower latency and 25% higher efficient

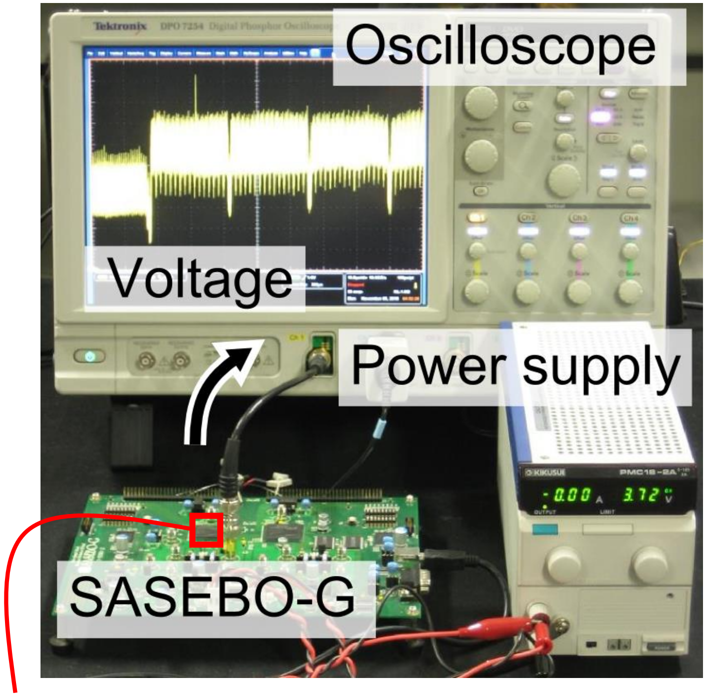
Outline

- Introduction
- TI-based AES S-box
- AES HW architecture for TI
- Experimental leakage evaluation
- Concluding remarks

Experimental evaluation of DPA-resistance

- Implement proposed AES on FPGA to perform Test Vector Leakage Assessment (TVLA)

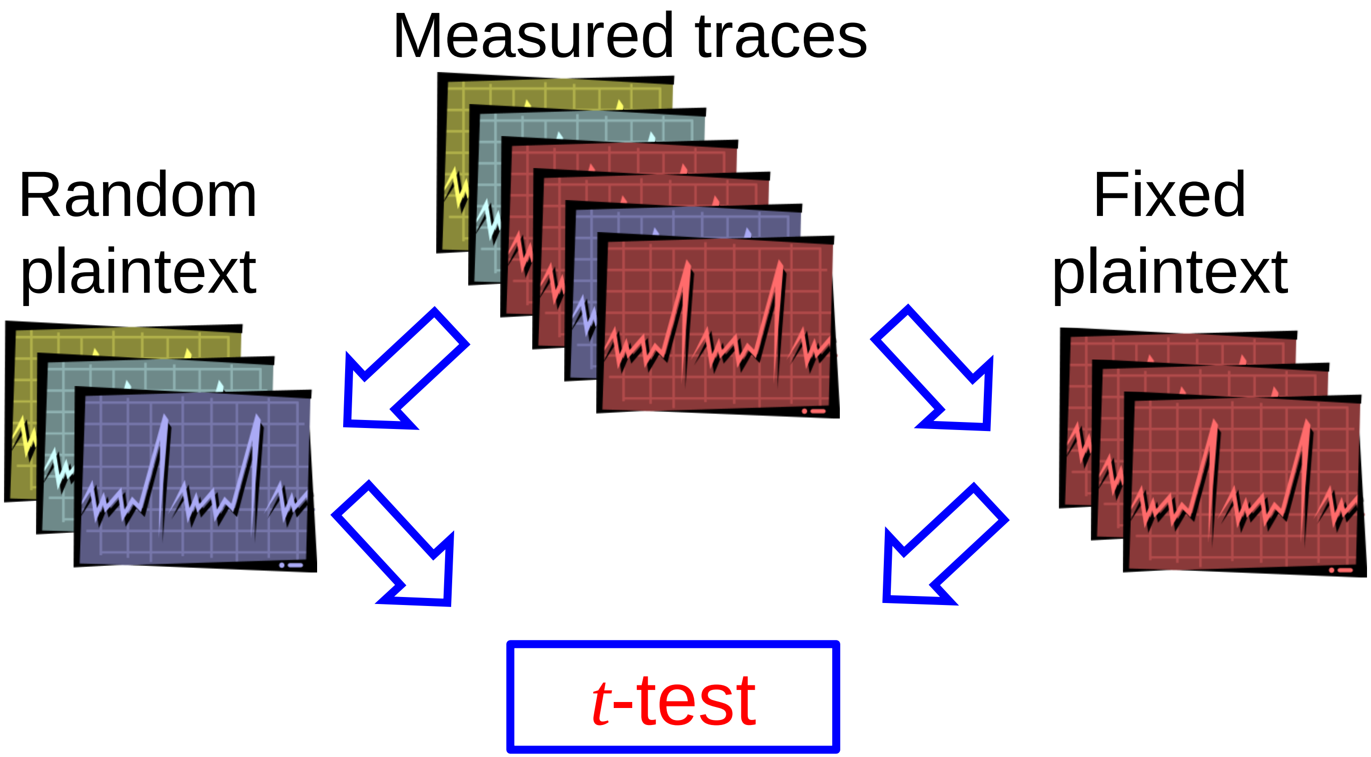
Setup overview



Proposed AES on FPGA

Experimental setup	
Board	SASEBO-G
FPGA	Xilinx Virtex PRO II
Frequency	24MHz
Oscilloscope	Tektronix DPO7254
Sampling rate	1GS/s
# of traces	500,000

Test Vector Leakage Assessment (TVLA)



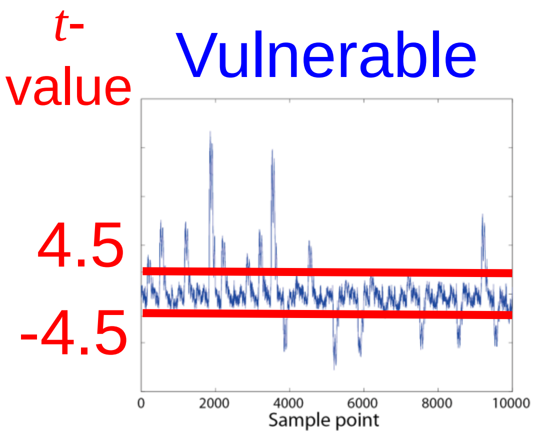
Measured traces

Random plaintext

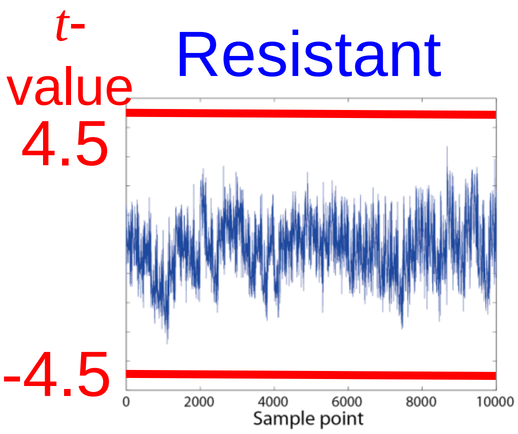
Fixed plaintext

t-test

Vulnerable

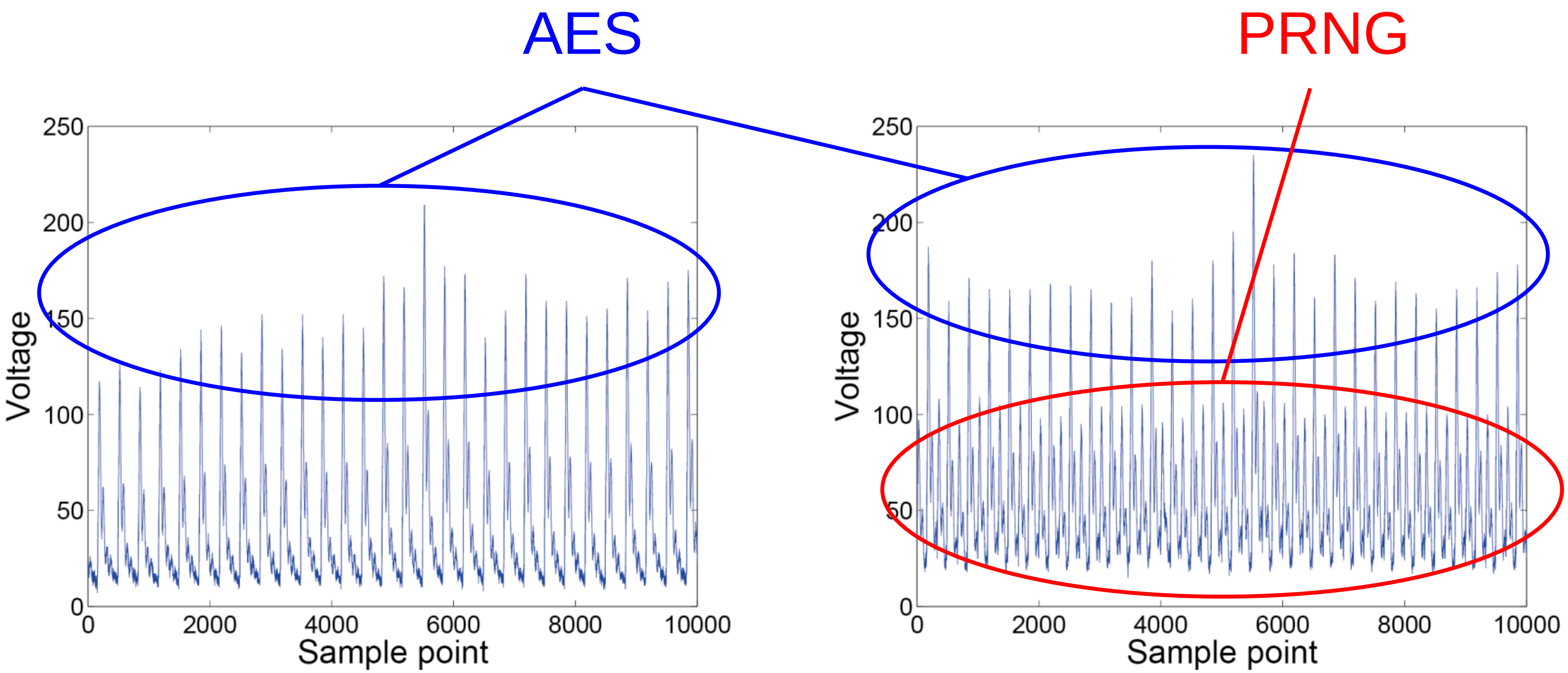


Resistant



Measured trace

- Random number for TI-based S-box is generated using LFSR-based PRNG



AES

PRNG

PRNG off

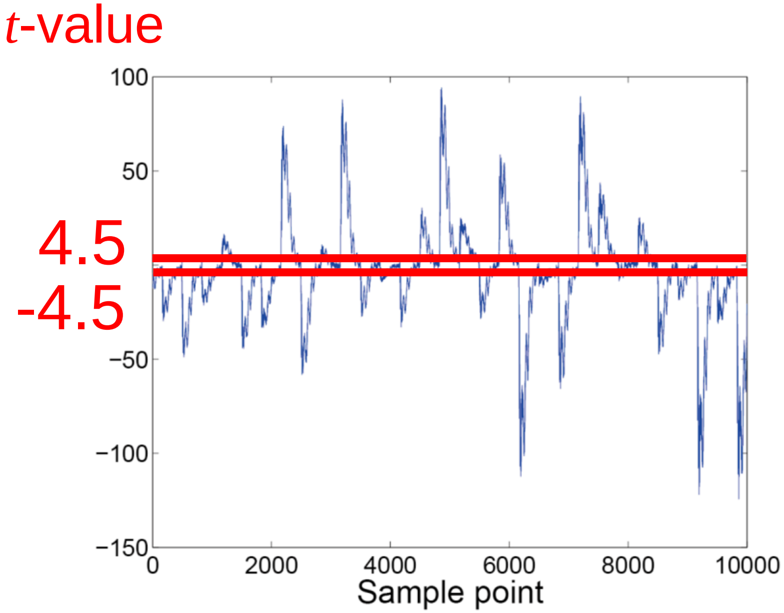
(Unprotected)

PRNG on

(Protected)

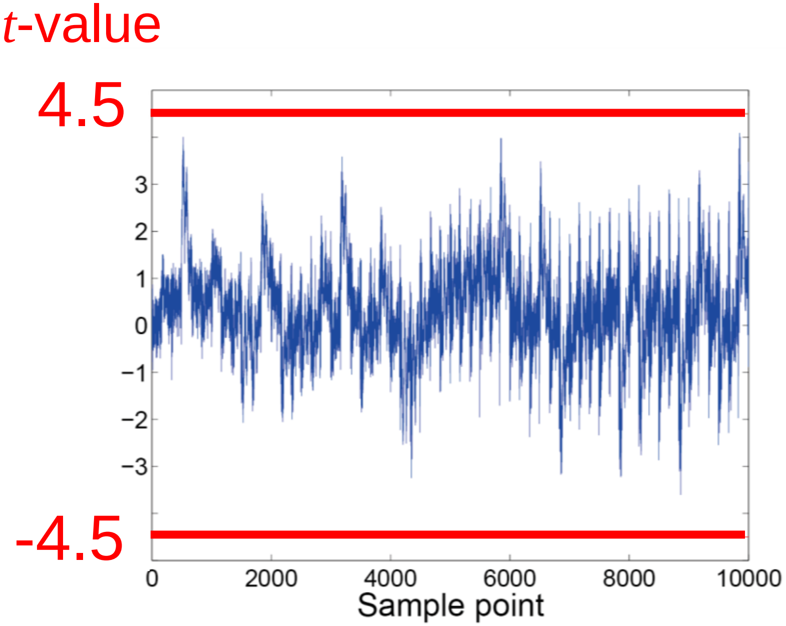
Result

- We can confirm first-order DPA resistance of our architecture under 500,000 traces



PRNG off

(Unprotected)



PRNG on

(Protected)

Concluding remarks

- Efficient DPA-resistant AES HW architecture based on TI
  - New TI-based S-box
    - 25% smaller area
  - Proposed AES HW architecture for TI
    - 11—21% lower-latency without area overhead
    - Secure under first-order DPAs with 500,000 traces
- Future works
  - Design and evaluate proposed HW architecture with higher-order TI-based S-boxes
  - Reduce randomness