KU LEUVEN

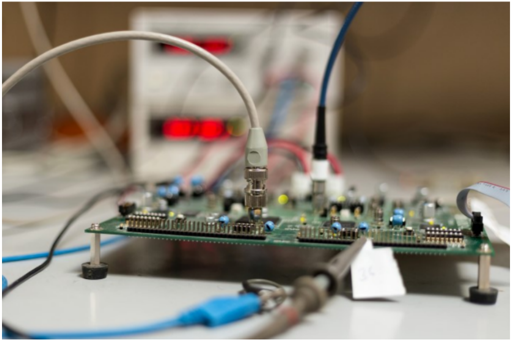
Trade-offs in protecting Keccak against combined side-channel and fault attacks

Antoon Purnal, Victor Arribas
and Lauren De Meyer

KU Leuven, imec-COSIC

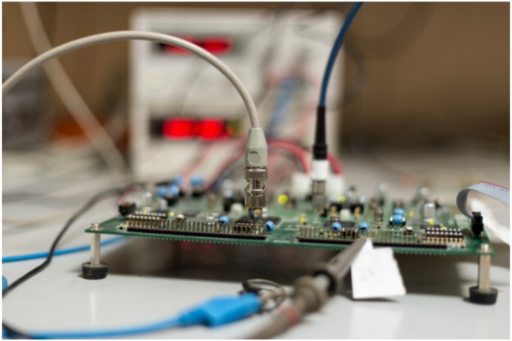
April 5, 2019

Physical attacks



- Side channel analysis
- Fault injection

Physical attacks



- Side channel analysis
- Fault injection
- Combined attacks - combined countermeasures: ParTI [SMG16], M&M [DAN+19], Capa [RDB+18]

Outline

CAPA

Protected implementations of KECCAK

Security evaluation

Conclusion

KU LEUVEN

Outline

CAPA

Protected implementations of Keccak

Security evaluation

Conclusion

KU LEUVEN

Adversarial model: tile-probe-and-fault

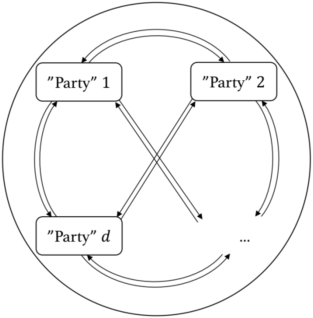


Figure: Tile architecture [RDB+18]

At least one of the d tiles shall remain uncompromised

KU LEUVEN

Representation

- Finite field $\mathbb{F}_q = GF(2)$
  - Addition is denoted $+$, $\sum$
  - Multiplication is denoted $\cdot$, $\prod$

Representation

- Finite field $\mathbb{F}_q = GF(2)$
  - Addition is denoted $+$, $\sum$
  - Multiplication is denoted $\cdot$, $\prod$

- MAC key $\alpha \in \mathbb{F}_q$
  - Every $x \in \mathbb{F}_q$ is authenticated by MAC tag $\tau^x = \alpha \cdot x$
  - MAC key is shared between the d tiles s.t. $\alpha = \sum \alpha_i$

KU LEUVEN

Representation

- Finite field $\mathbb{F}_q = GF(2)$
  - Addition is denoted $+$, $\sum$
  - Multiplication is denoted $\cdot$, $\prod$

- MAC key $\alpha \in \mathbb{F}_q$
  - Every $x \in \mathbb{F}_q$ is authenticated by MAC tag $\tau^x = \alpha \cdot x$
  - MAC key is shared between the d tiles s.t. $\alpha = \sum \alpha_i$

- Representation of a secret value $x \in \mathbb{F}_q$ in the masked domain

$$\langle x \rangle = (\boldsymbol{x}, \boldsymbol{\tau}^x)$$

Data shares $\boldsymbol{x} = (x_1, x_2, \ldots, x_d)$ such that $x = \sum x_i$
Tag shares $\boldsymbol{\tau}^x = (\tau_1^x, \tau_2^x, \ldots, \tau_d^x)$ such that $\tau^x = \sum \tau_i^x$

Computing procedure - addition

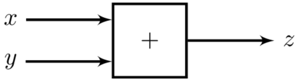


Figure: Original addition

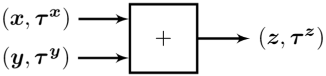


Figure: Masked addition

Computing procedure - addition

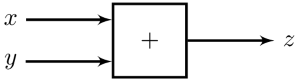


Figure: Original addition

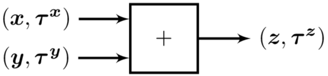


Figure: Masked addition

- Each tile $\mathbb{T}_i$ locally computes its share of the output z
  - Data share $z_i = x_i + y_i$
  - Tag share $\tau_i^z = \tau_i^x + \tau_i^y$

Computing procedure - addition

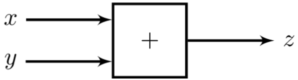


Figure: Original addition

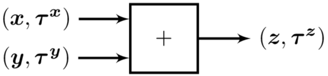


Figure: Masked addition

- Each tile $\mathbb{T}_i$ locally computes its share of the output z
  - Data share $z_i = x_i + y_i$
  - Tag share $\tau_i^z = \tau_i^x + \tau_i^y$
- *Correctness.*

$$\sum z_i = \sum (x_i + y_i) = \sum x_i + \sum y_i = x + y = z$$
$$\sum \tau_i^z = \sum (\tau_i^x + \tau_i^y) = \sum \tau_i^x + \sum \tau_i^y = \tau^x + \tau^y = \tau^z$$

Computing procedure - multiplication

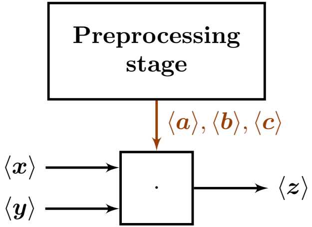


Figure: Auxiliary triple for multiplication

- Using Beaver triple $\langle a \rangle$, $\langle b \rangle$, $\langle c \rangle$ where $c = a \cdot b$
- Two-cycle latency
- MAC tag check

Capa

- Evaluation and preprocessing stage
- Number of tiles $d \Longrightarrow (d-1)$th order SCA resistance
- Security parameter $m \Longrightarrow$ fault detection probability $1-2^{-m}$
  - m independent MAC keys α

Outline

CAPA

Protected implementations of KECCAK
- Preliminaries
- Hardware designs
- Results

Security evaluation

Conclusion

KECCAK-f permutations

- Permutation width $b \in \{25, 50, 100, 200, 400, 800, 1600\}$
- Round function R
- Number of rounds $n_r = 12 + 2\log_2(w)$, where $w = \frac{b}{25}$

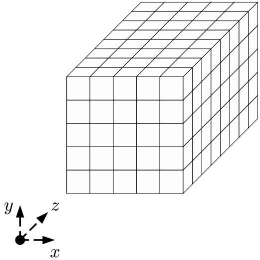
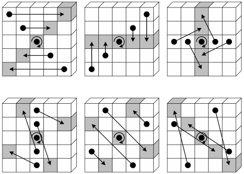


Figure: The KECCAK state [BDPVA09]

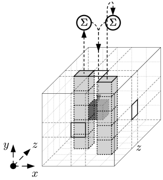
KU LEUVEN

KECCAK-f permutations

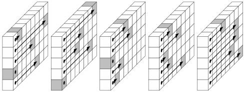
$$R = \iota \circ \chi \circ \pi \circ \rho \circ \theta$$



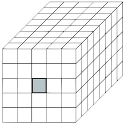
π



θ



ρ



ι

KU LEUVEN

KECCAK-f permutations

$$R = \iota \circ \chi \circ \pi \circ \rho \circ \theta$$

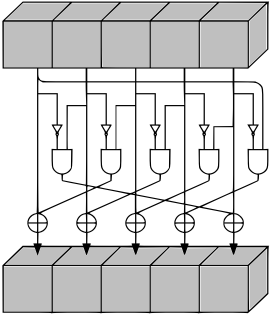


Figure: The χ step mapping [BDPVA09]

- b multiplications each round
- Most expensive operation

Outline

CAPA

Protected implementations of KECCAK

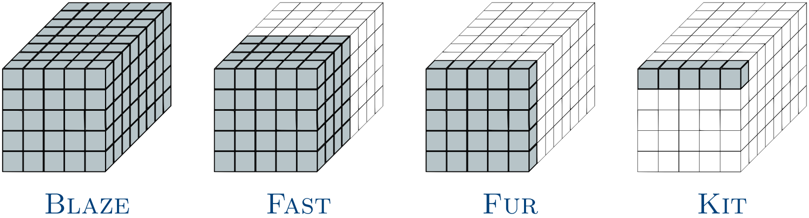
- Preliminaries
- Hardware designs
- Results

Security evaluation

Conclusion

The speed-area tradeoff

$$R = \iota \circ \chi \circ \pi \circ \rho \circ \theta$$



KU LEUVEN

Blaze - high throughput

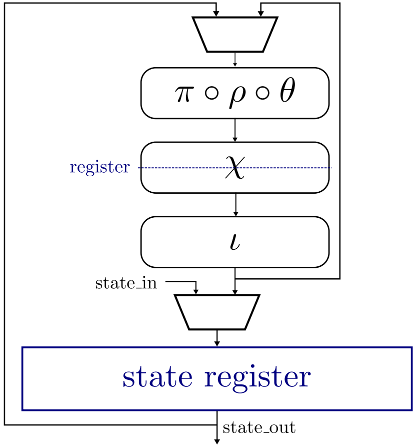
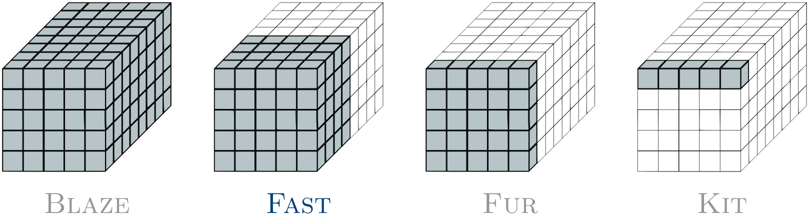


Figure: High-level architecture for Blaze

The speed-area tradeoff

$$R = \iota \circ {\color{red}\chi} \circ \pi \circ \rho \circ \theta$$



FAST - moderate throughput

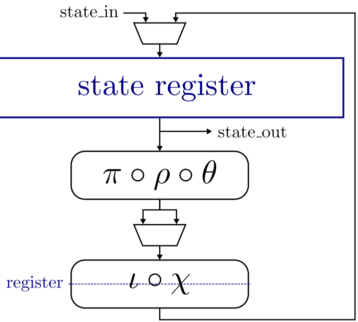
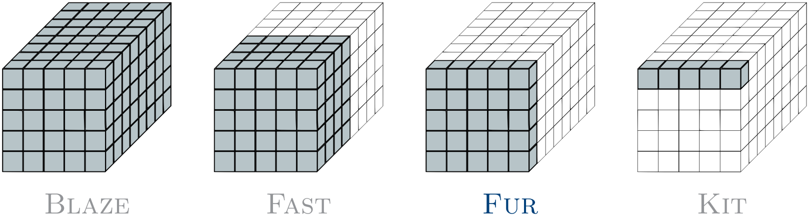


Figure: High-level architecture for FAST

- Half state for $\iota \circ \chi$
- Full state for $\pi \circ \rho \circ \theta$
- ≈ 3 cycles per round

The speed-area tradeoff

$$R = \iota \circ \chi \circ \pi \circ \rho \circ \theta$$



Slice-based processing

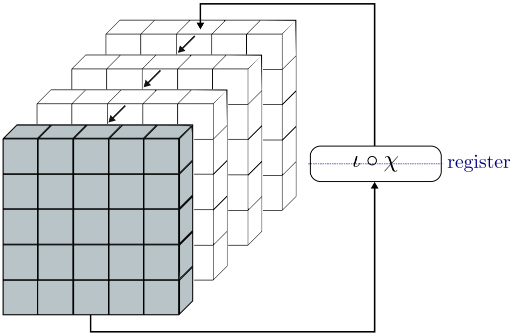


Figure: Slice-based processing

FUR - moderate area

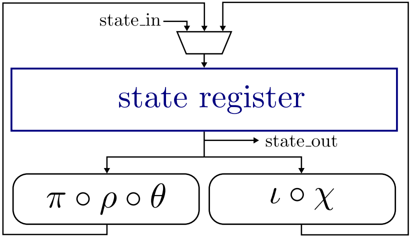
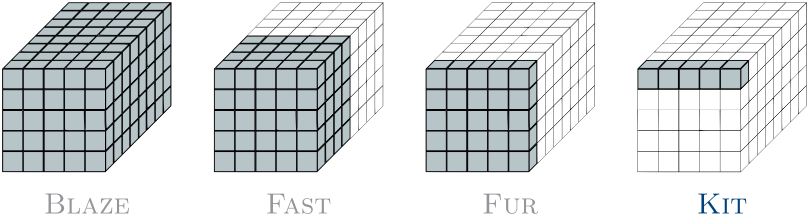


Figure: High-level architecture for FUR

- Full state for $\pi \circ \rho \circ \theta$
- Slice-based for $\iota \circ \chi$
- $\approx w + 2$ cycles per round

The speed-area tradeoff

$$R = \iota \circ \chi \circ \pi \circ \rho \circ \theta$$



Row-based processing

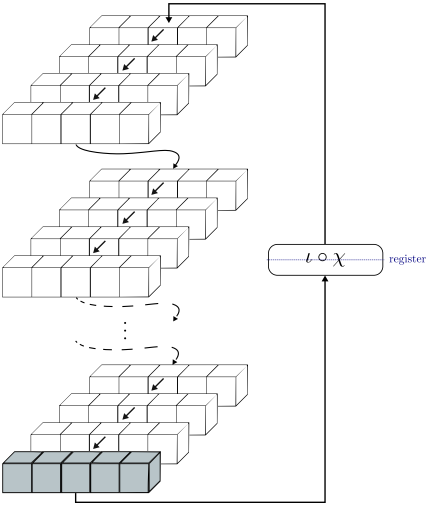


Figure: Row-based processing

KU LEUVEN

KIT - low area

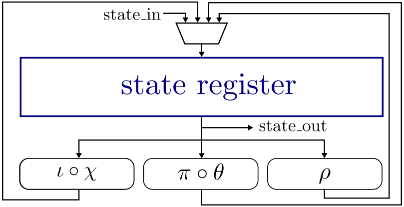
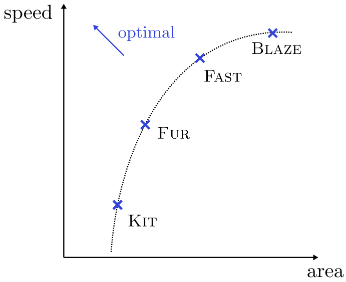


Figure: High-level architecture for KIT

- Slice-based for $\pi \circ \theta$
- Slice-based for ρ
- Row-based for $\iota \circ \chi$
- $\approx 7w + 1$ cycles per round

Summary



Design	S-boxes (χ)	Preprocessing	Cycle count
BLAZE	$b/5$	b	$n_r + 2$
FAST	$b/10$	$b/2$	$3 \cdot n_r + 1$
FUR	5	25	$(w+2) \cdot n_r + 1$
KIT	1	5	$(7w+1) \cdot n_r + 1$

KU LEUVEN

Outline

Literature comparison

Order	Design	AREA [kGE] Evaluation χ	θ	State	Σ	Prep.	Total	Rand. [bpc]	f_{max} [MHz]	Cycles [/]
KECCAK-f[1600] in NANGATE 45nm ($m=0$)										
1	BLAZE	145.1	12.8	33.7	199.7	231.0	430.7	16000	892	25
	Parallel [GSM17]	38.4	15.0	32.2	85.7	-	85.7	480	891	48
	Parallel-3sh [BDN+13]	40.6	19.2	56.8	116.6	-	116.6	4	592	25
2	BLAZE	235.2	19.2	50.5	317.1	449.3	766.4	28800	884	25
	Parallel [GSM17]	114.0	22.5	51.1	188.1	-	188.1	4800	898	48
KECCAK-f[200] in NANGATE 45nm ($m=0$)										
1	BLAZE	18.1	1.6	4.2	25.2	28.9	54.0	2000	892	19
	5-10-5 [ABP+18]	73.4	14.0	11.9	99.3	-	99.3	-	395.25	9
	6-6-6 [ABP+18]	44.6	11.3	14.2	70.1	-	70.1	-	436.7	9

Table: Comparison with previous work for representative designs

KU LEUVEN

Literature comparison

Keccak-f[1600] in NANGATE 45nm ($m=0$)										
		AREA [kGE]						Rand.	f_{max}	Cycles
		Evaluation				Prep.	Total			
Order	**Design**	χ	θ	State	Σ			[bpc]	[MHz]	[/]
1	Blaze	145.1	12.8	33.7	199.7	231.0	430.7	16000	892	25
	Parallel [GSM17]	38.4	15.0	32.2	85.7	-	85.7	480	891	48
	Parallel-3sh [BDN+13]	40.6	19.2	56.8	116.6	-	116.6	4	592	25
	Kit	0.5	0.6	26.1	29.1	0.7	29.8	50	1538	10776
	Serial-Area [GSM17]	0.4	0.4	14.5	15.7	-	15.7	-	850	3160
	Serial-3sh [BDN+13]	0.6	0.3	38.1	39.0	-	39.0	< 1	645	1625
2	Blaze	235.2	19.2	50.5	317.1	449.3	766.4	28800	884	25
	Parallel [GSM17]	114.0	22.5	51.1	188.1	-	188.1	4800	898	48
	Kit	0.7	1.0	39.1	43.7	1.4	45.1	90	1351	10776
	Serial-Area [GSM17]	2.2	0.6	21.4	24.2	-	24.2	75	898	3160
Keccak-f[200] in NANGATE 45nm ($m=0$)										
1	Blaze	18.1	1.6	4.2	25.2	28.9	54.0	2000	892	19
	5-10-5 [ABP+18]	73.4	14.0	11.9	99.3	-	99.3	-	395.25	9
	6-6-6 [ABP+18]	44.6	11.3	14.2	70.1	-	70.1	-	436.7	9

Table: Comparison with previous work for representative designs

Outline

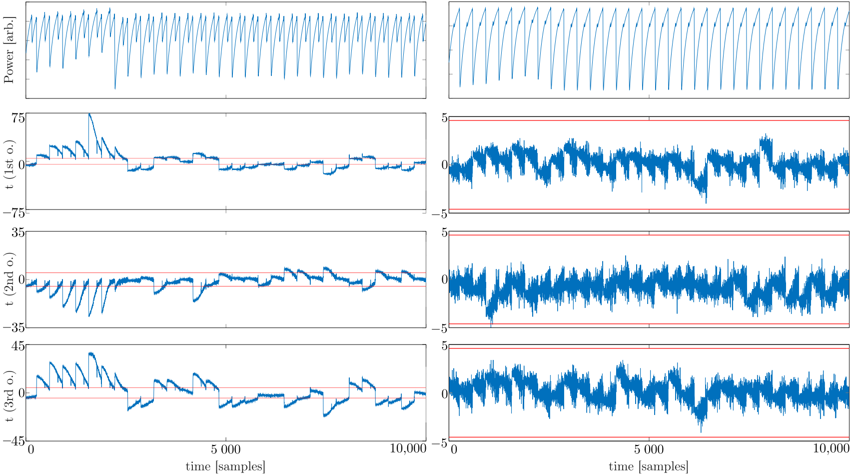
CAPA

Protected implementations of KECCAK

Security evaluation

Conclusion

Leakage detection (Kit, $d = 3$, $m = 2$)



(a) Masks off (b) Masks on

Platform: Sakura-G board (2x Xilinx Spartan 6 FPGA)

KU LEUVEN

Leakage detection - over time

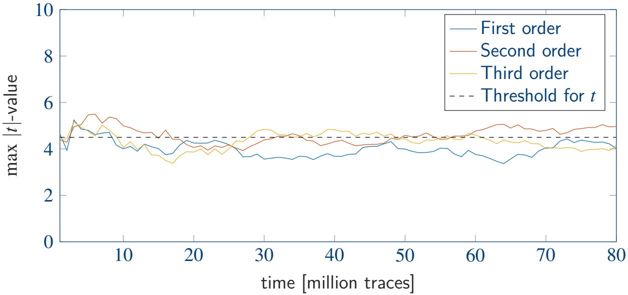


Figure: Maximum t-test value over time

Fault coverage (Kit, $d = 2$, m varies)

	$\boldsymbol{m} = 2$	$\boldsymbol{m} = 4$	$\boldsymbol{m} = 6$	$\boldsymbol{m} = 8$
# valid $\langle \boldsymbol{f} \rangle$	32	512	8192	131072
# detected $\langle \boldsymbol{f} \rangle$	24	480	8064	130560

Table: Experimental fault resistance results

- Simulation-based testing (HDL): fault vectors $\langle \boldsymbol{f} \rangle$
- Fault at different locations but stick to one MAC key guess
- Deterministic experiment: $1 - 2^{-m}$
- Extrapolate results for $m > 8$

Outline

CAPA

Protected implementations of KECCAK

Security evaluation

Conclusion

Conclusion and future work

- First implementations of KECCAK with resistance against combined attacks
  - Design space exploration: BLAZE, KIT and everything in between
  - Combined countermeasures skew the hardware design space
- Performance assessment as a function of the security parameters b, m, d [see paper]
- More efficient preprocessing stage, generally applicable [see paper]
- Currently only the small implementations have realistic requirements
  - Relax attacker model?
  - Define authentication tag in a different way?

References I

Victor Arribas, Begül Bilgin, George Petrides, Svetla Nikova, and Vincent Rijmen.
Rhythmic Keccak: SCA security and low latency in HW.
IACR Trans. Cryptogr. Hardw. Embed. Syst., 2018(1):269–290, 2018.

Begül Bilgin, Joan Daemen, Ventzislav Nikov, Svetla Nikova, Vincent Rijmen, and Gilles Van Assche.
Efficient and first-order DPA resistant implementations of Keccak.
In *Smart Card Research and Advanced Applications - 12th International Conference, CARDIS 2013, Berlin, Germany, November 27-29, 2013. Revised Selected Papers*, pages 187–199, 2013.

Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche.
Keccak sponge function family main document.
Submission to NIST (Round 2), 3(30), 2009.

Lauren De Meyer, Victor Arribas, Svetla Nikova, Ventzislav Nikov, and Vincent Rijmen.
M&M: Masks and Macs against physical attacks.
IACR Trans. Cryptogr. Hardw. Embed. Syst., 2019(1):25–50, 2019.

Hannes Groß, David Schaffenrath, and Stefan Mangard.
Higher-order side-channel protected implementations of Keccak.
In *Euromicro Conference on Digital System Design, DSD 2017, Vienna, Austria, August 30 - Sept. 1, 2017*, pages 205–212, 2017.

References II

Oscar Reparaz, Lauren De Meyer, Begül Bilgin, Victor Arribas, Svetla Nikova, Ventzislav Nikov, and Nigel P. Smart.
CAPA: the spirit of Beaver against physical attacks.
In *Advances in Cryptology - CRYPTO '18, 38th Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2018*, 2018.

Tobias Schneider, Amir Moradi, and Tim Güneysu.
ParTI - towards combined hardware countermeasures against side-channel and fault-injection attacks.
In *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, pages 302–332, 2016.

KU LEUVEN