An Area-Optimized Serial Implementation of ICEPOLE Authenticated Encryption Schemes

Michael Tempelmeier*, Fabrizio De Santis*, Jens-Peter Kaps‡ and Georg Sigl†*

*Technical University of Munich,
‡George Mason University,
†Fraunhofer AISEC

CAESAR

Competition for Authenticated Encryption: Security, Applicability, and Robustness

Important Dates:

- 2013 January: Competition announced
- 2014 March: Deadline for first-round submissions

- **2015 July: Announcement of second-round candidates**
- ~~**2015 December: Deadline for second-round Verilog/VHDL**~~
- **Today: HOST2016**
- **2016 May (tentative): Deadline for second-round Verilog/VHDL**

- **2016 June: Announcement of third-round candidates**
- 2017 December: Announcement of final portfolio

ICEPOLE

Motivation

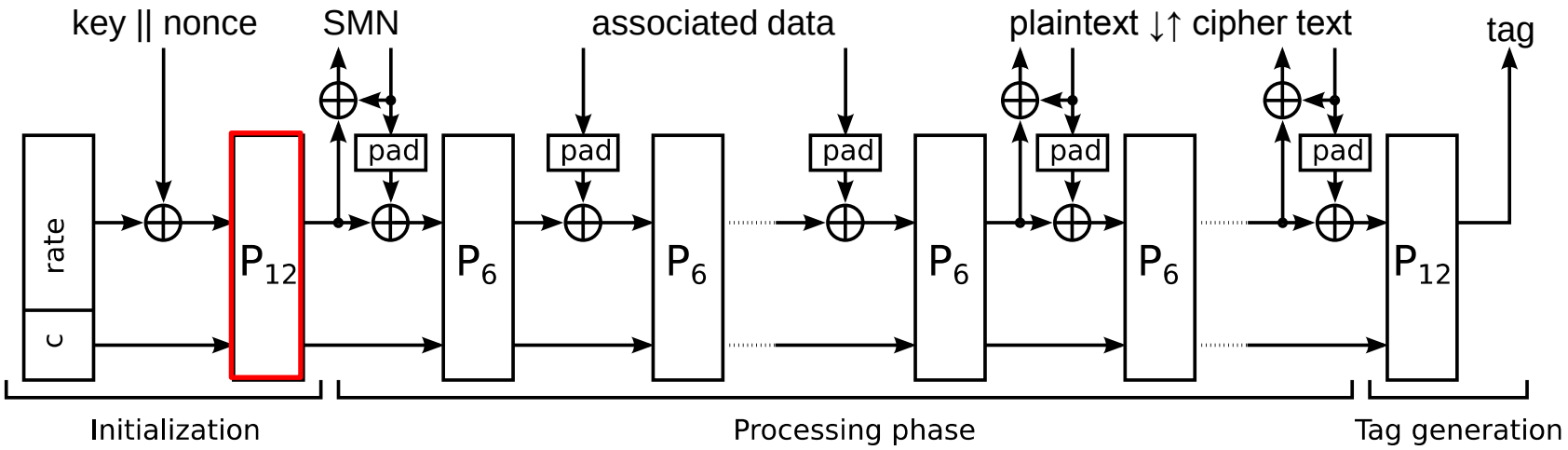
- No (serial) hardware implementations so far
- Low area implementations important for fair comparison
- Deadline for hardware implementations is nigh
  ➔ Straight forward implementation will be published soon

Artwork by Ewa Baran

Agenda

- ICEPOLE-128
  - Overview
  - State organisation
  - Round function
  - Corner case: Kappa step
- Slice-serial Implementation
  - 20 bit architecture
  - I/O-Interface
- Results
- Conclusion & Outlook

ICEPOLE-128



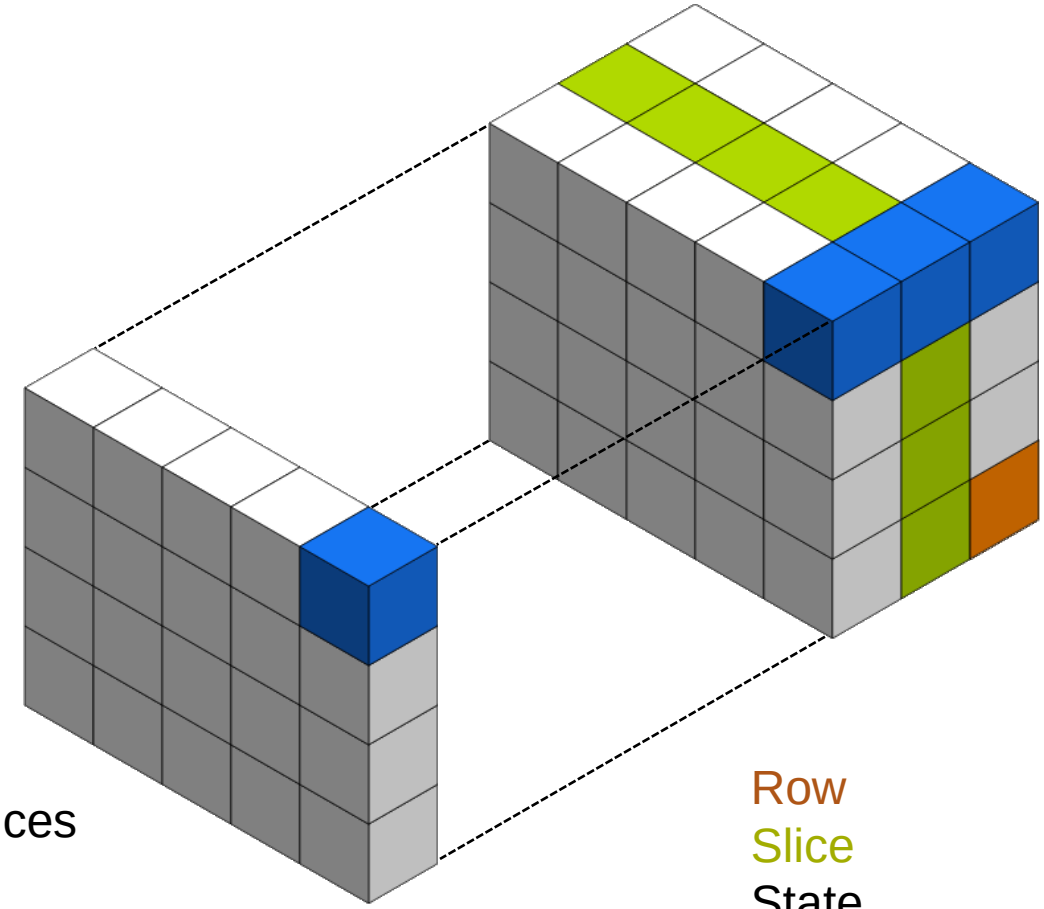
"High-speed, hardware-oriented family of single-pass authenticated encryption schemes"

- Duplex construction (similar to sponge construction like SHA-3)
- 128-bit key, nonce, secret message number (SMN), tag
- Up to 1024-bit block size (associated data, plaintext, cipher text)

Permutation Layer

State Organisation

- 1280-bit (capacity and rate)
- 4 x 5 x 64 cube



Round Function

$R = \kappa \circ \psi \circ \pi \circ \rho \circ \mu$

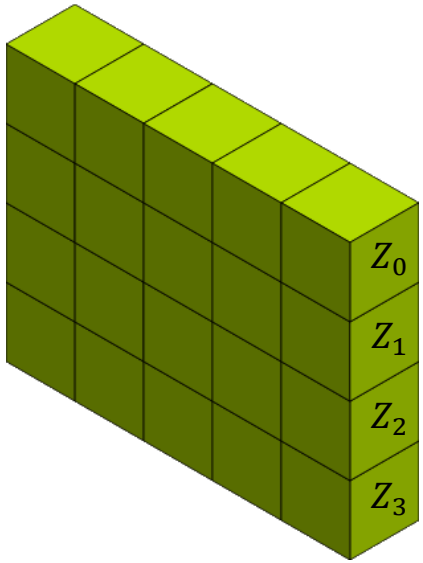
- μ linear transformation of the slices
- ρ linear rotation of each word
- π linear permutation of the words
- ψ nonlinear permutation of one row
- κ addition of round constant

Picture created with Isometric Drawing Tool:
http://illuminations.nctm.org

Round Function

R = κ ∘ ψ ∘ π ∘ ρ ∘ μ

μ Step



- GF(2^5) multiplication modulo $x^5 + x^2 + 1$
- Main source of diffusion
- Linear transformation of the slices

  ➔ only XORs needed

$$\begin{bmatrix} 2 & 1 & 1 & 1 \\ 1 & 1 & 18 & 2 \\ 1 & 2 & 1 & 18 \\ 1 & 18 & 2 & 1 \end{bmatrix} \begin{bmatrix} Z_0 \\ Z_1 \\ Z_2 \\ Z_3 \end{bmatrix} = \begin{bmatrix} 2Z_0 + Z_1 + Z_2 + Z_3 \\ Z_0 + Z_1 + 18Z_2 + 2Z_3 \\ Z_0 + 2Z_1 + Z_2 + 18Z_3 \\ Z_0 + 18Z_1 + 2Z_2 + Z_3 \end{bmatrix}$$

Picture created with Isometric Drawing Tool:
http://illuminations.nctm.org

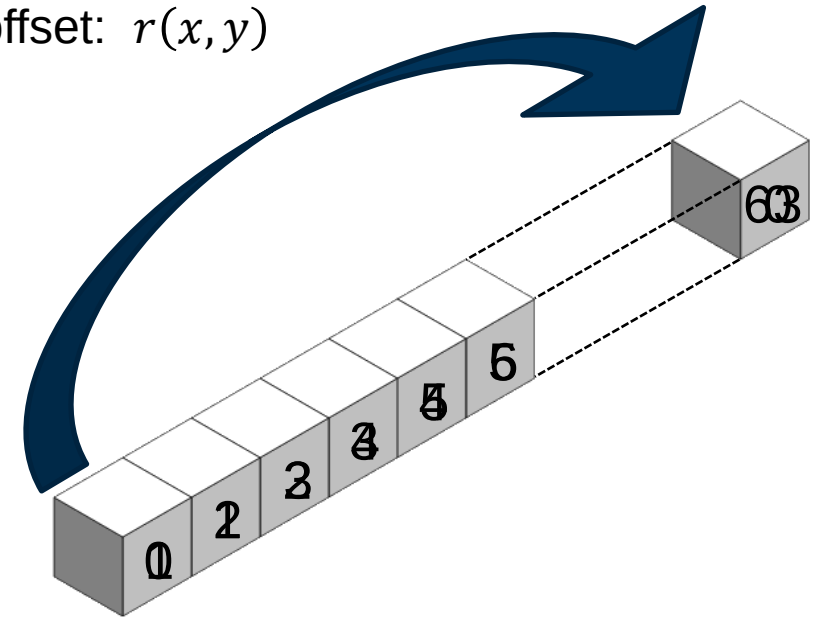
Round Function

R = κ ◦ ψ ◦ π ◦ ρ ◦ μ

ρ Step

- Rotation of each word with a different offset: $r(x, y)$
- Mixes information between slices

$$S[z'] = S[z + r(x, y) \; mod \; 64]$$



Picture created with Isometric Drawing Tool:
http://illuminations.nctm.org

Round Function

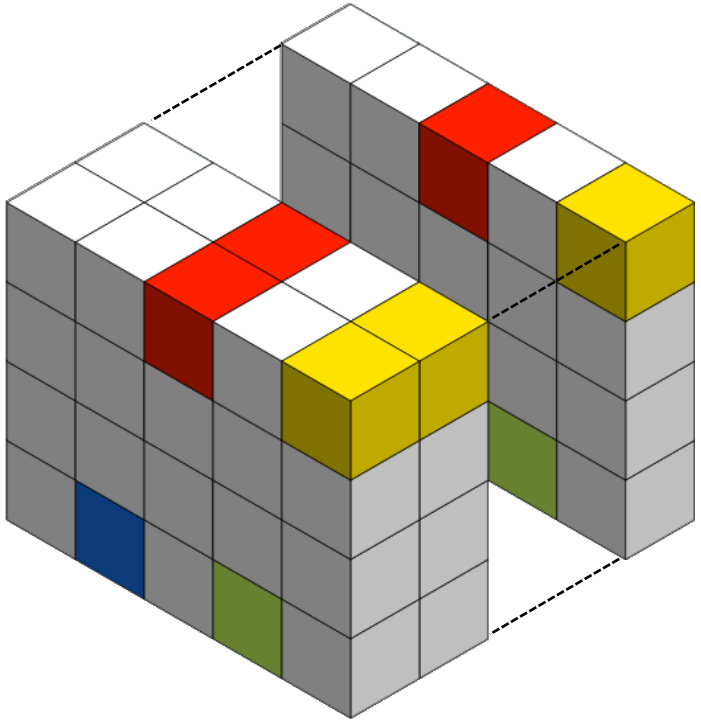
R = κ ∘ ψ ∘ π ∘ ρ ∘ μ

π Step

- Linear permutation of the words in the state
- No additional logic required

$$S[x'][y'] \leftarrow \pi(S[x][y])$$

$$x' := (x + y) \bmod 4$$
$$y' := \big(((x + y) \bmod 4) + y + 1\big) \bmod 5$$



Picture from:
Morawiecki, Paweł, et al. [presentation slides] “ICEPOLE: high-speed, hardware-oriented authenticated encryption.” – CHES 2014.
created with Isometric Drawing Tool: http://illuminations.nctm.org

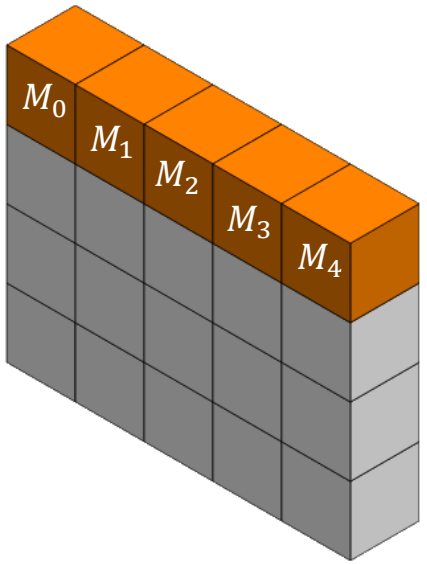
Round Function

R = κ ∘ ψ ∘ π ∘ ρ ∘ μ

ψ Step

- Nonlinear transformation of the state
- 5 bit s-box maps input M_k to output Z_k

$$Z_k = M_k \oplus (\bar{M}_{k+1} \wedge M_{k+2}) \oplus (\bar{M}_0 \wedge \bar{M}_1 \wedge \bar{M}_2 \wedge \bar{M}_3 \wedge \bar{M}_4) \oplus (M_0 \wedge M_1 \wedge M_2 \wedge M_3 \wedge M_4)$$



Picture created with Isometric Drawing Tool:
http://illuminations.nctm.org

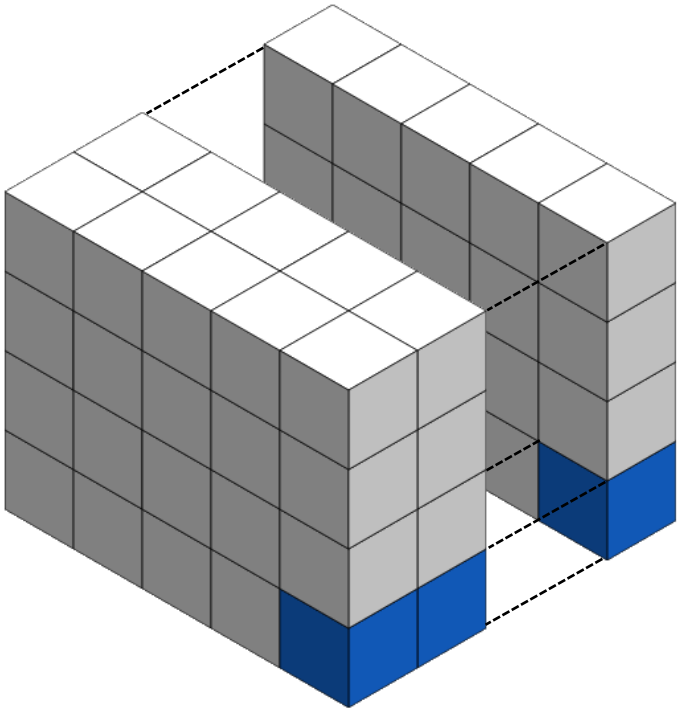
Round Function

R = κ ∘ ψ ∘ π ∘ ρ ∘ μ

κ Step

- Adds a round constant to each round:
  $S[0][0] \coloneqq S[0][0] \oplus const[roundNumber]$

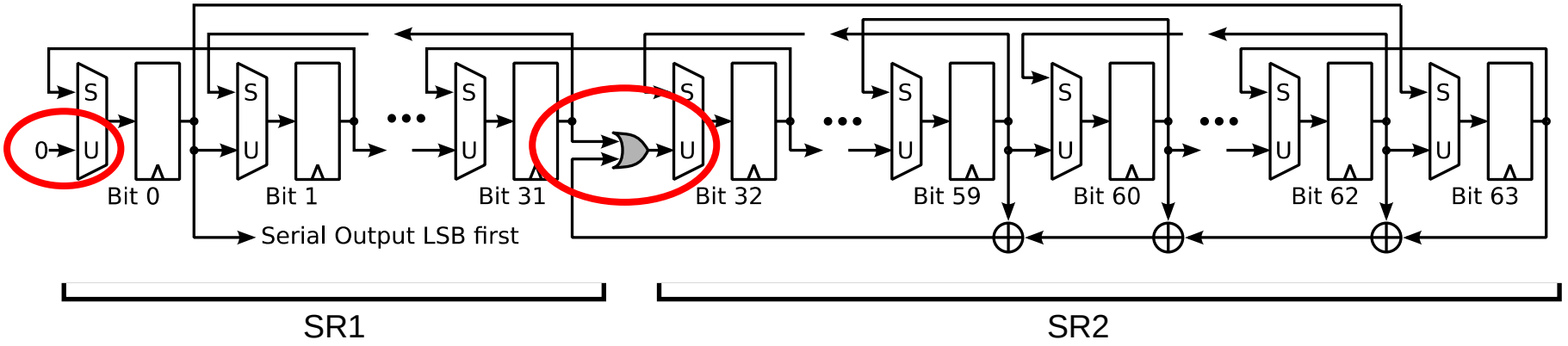
- Constants should be generated by LFSR with feedback polynomial:
  $f(x) = 1 + x^{60} + x^{61} + x^{63} + x^{64}$



Picture created with Isometric Drawing Tool:
http://illuminations.nctm.org

Corner Case: Kappa

- Constants **cannot** be generated with the proposed 64-bit LFSR!
- Instead: two 32-bit shift registers and nonlinear feedback path



Two modes:

- Update: Zero input, nonlinear feedback (right shift)
- Shift: Left shift out for endianness in serial implementation (same state after 64 cycles)

Round Function

Summary

Round Function

R = κ ◦ ψ ◦ π ◦ ρ ◦ μ

- μ: slice → slice
- ρ: word → word
- π: slice → slice
- ψ: row → row ➔ four parallel ψ: slice → slice
- κ: word → word ➔ serialised: bit → bit

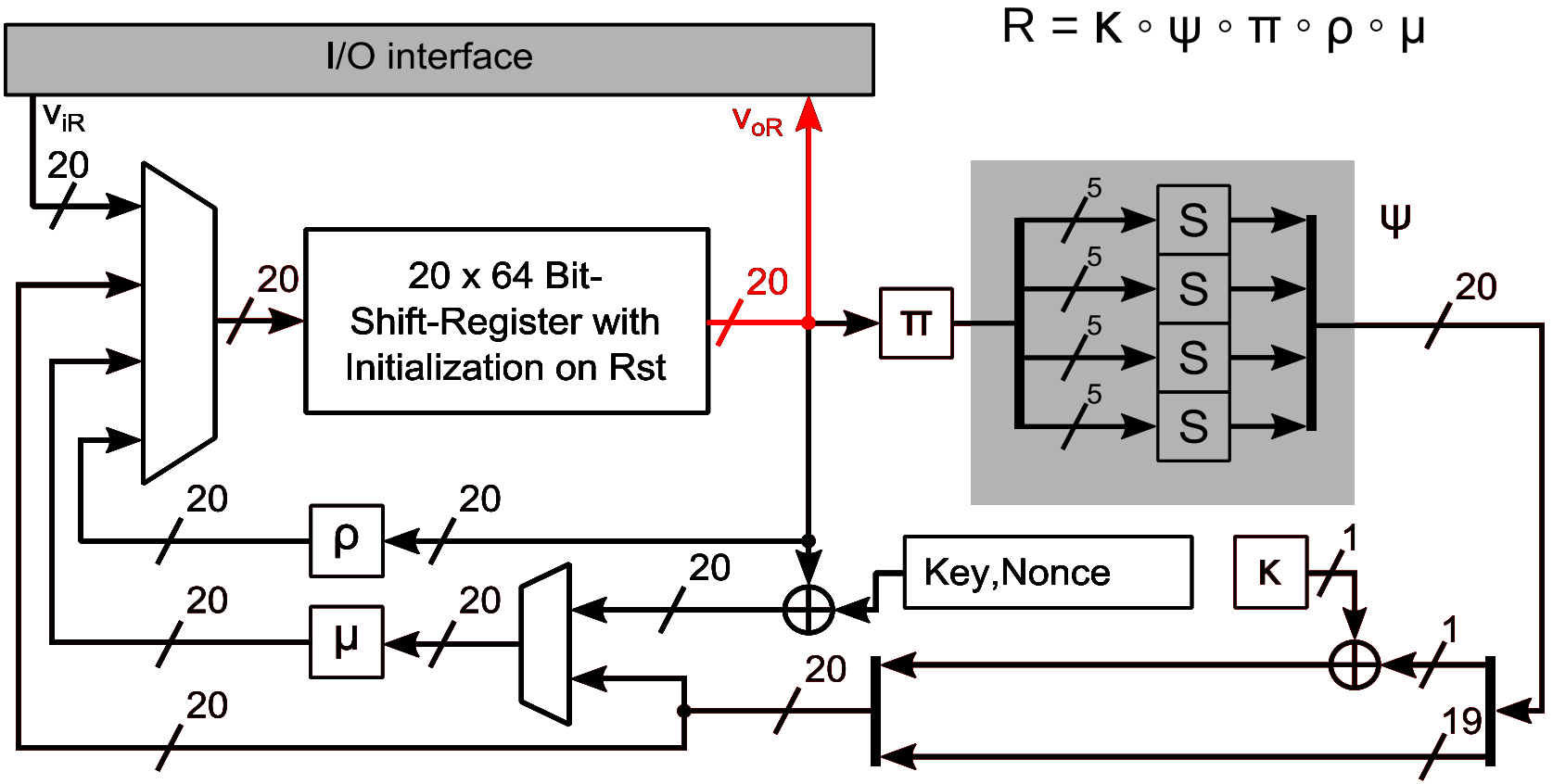
Slice Serial Architecture

State Organisation

- 20 x 64 bit shift enable registers (ρ)
- Access to the same bits of all words at the same time (κ,ψ,π,μ)

Hardware Implementation

Slice Serial Architecture



I/O-Interface

How to load the state?

Slices vs. words

- Determines time to load/process one input block
- High influence on throughput

Slice serial:

- ➖ Input data must be reordered
- ➕ Fast

Bit serial:

- ➕ Only parallel serial converter needed
- ➖ Slow

Implementation Results

Low-Area (ICEPOLE-128)

Device	Area			Freq. max. (MHz)	Throughput (Mbps)
	Slices	LUTs	FFs		
Spartan-3E	951	1,846	1,620	157	197
Virtex-6	274	946	1,618	374	468
Artix-7	359	957	1,618	296	370

Area (slices) can be shrunk to 50% (realised in subsequent work):

- Use SRLC16E instead of flip flops for state
- Change endianness ➔ change ρ step ➔ $S[z] = S[64 - (z + r(x, y)\ mod\ 64)]$
  ➔ change κ step ➔ no multiplexers are needed

Implementation Results

Low-Area (XILINX VIRTEX-6)

Design	Area (Slices)	Throughput (Mbps)	Throughput/Area (Mbps/Slices)
ICEPOLE-128	274	468	1.710
Keyak [1]	218	688	3.154
AES-CCM [2]	190	474	2.474
AES-GCM [1]	350	127	0.363

[1] P. Yalla, E. Homsirikamol, and J.-P. Kaps, “Comparison of multi-purpose cores of Keccak and AES,” DATE 2015

[2] AES-CCM Core family for Xilinx FPGA, Helion Technology Limited, Fulbourn, Cambridge CB21 5DQ, England, 2011

Conclusion & Outlook

- ICEPOLE is a promising candidate

- However: some flaws in documentation - can be remedied:
  - κ Step
  - State description

- Interface is important in serial implementations

  ➔ Standardized I/O interface needed!

  ➔ Maybe the “GMU Hardware API for Authenticated Ciphers”?

Questions?