2 Gbit/s Hardware Realizations of RIJNDAEL and SERPENT: A Comparative Analysis

Adrian K. Lutz[1], Jürg Treichler[2], Frank K. Gürkaynak[3],
Hubert Kaeslin[4], Gérard Basler[2], Andres Erni[1],
Stefan Reichmuth[1], Pieter Rommens[2],
Stephan Oetiker[3], Wolfgang Fichtner[3]

Integrated Systems Laboratory, ETH Zurich

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

[1] Rijndael Design Team [2] Serpent Design Team [3] Integrated Systems Laboratory [4] Microelectronics Design Center

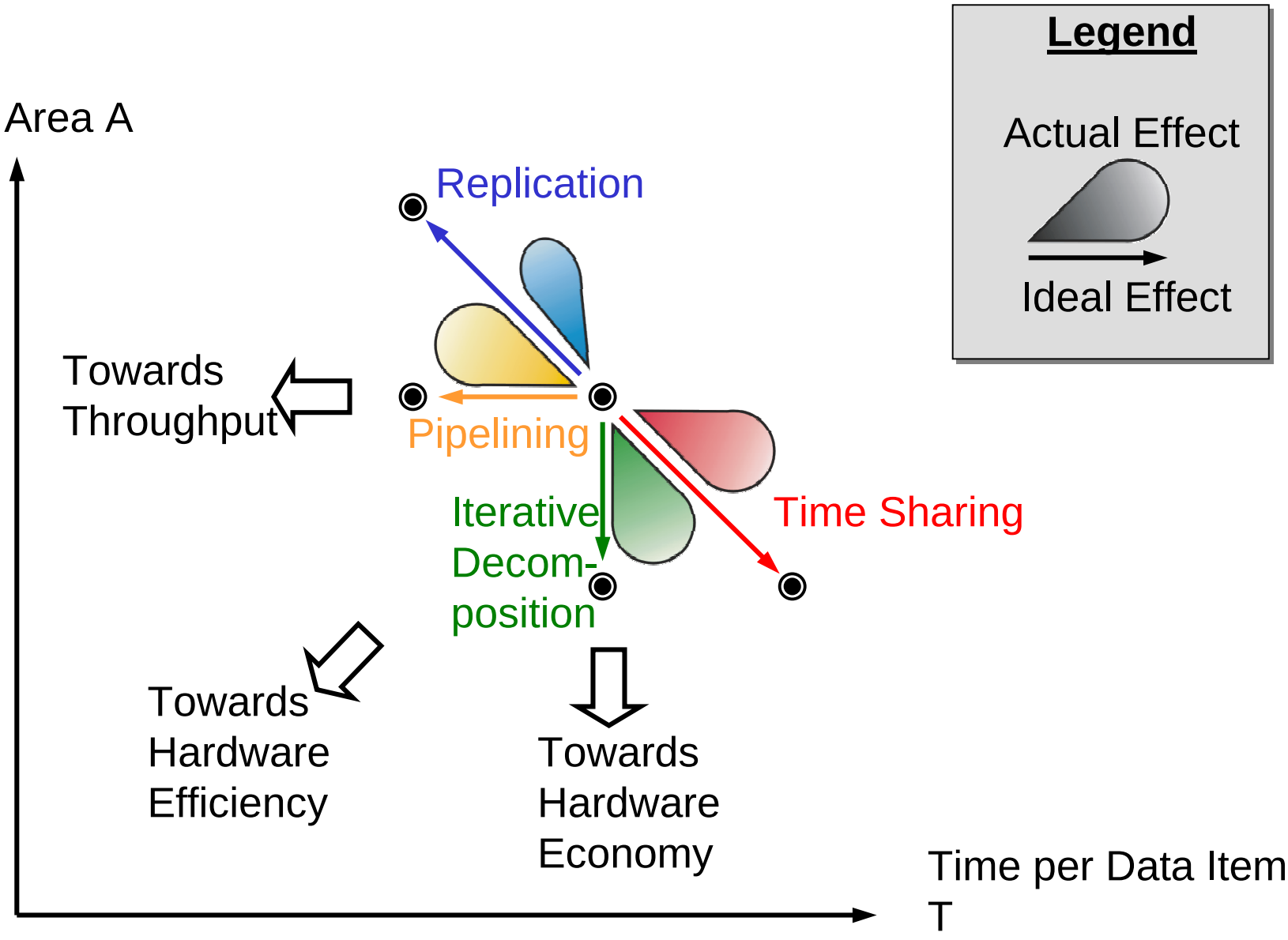
Contents

- Context
- Architectural Transforms
- Variations of Architectural Parameters
- Rijndael Architecture
- Serpent Architecture
- Comparison
- Conclusions

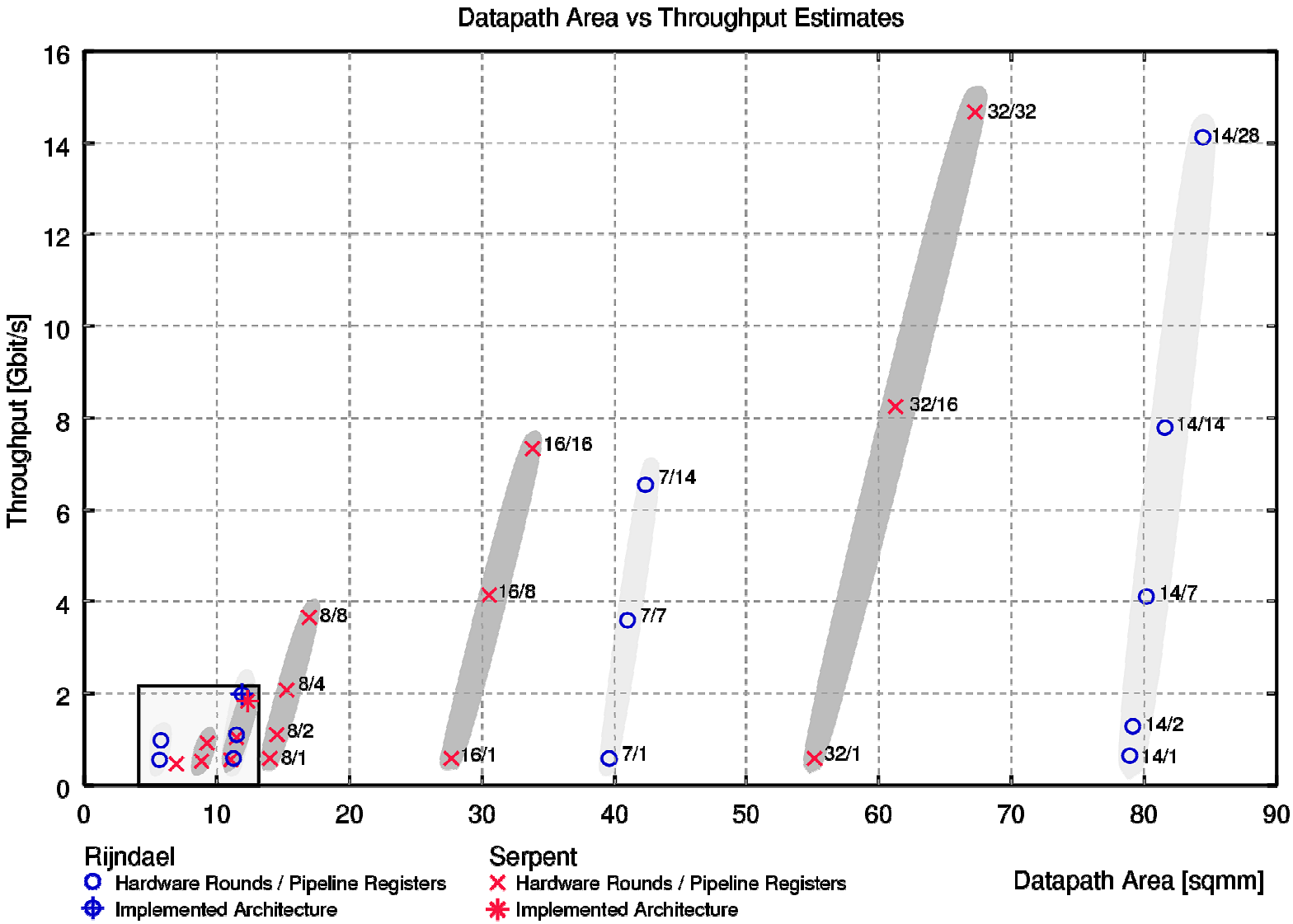
Context

- Term project at the Integrated Systems Laboratory of ETH Zurich
- 2 teams of 3 EE 4th year students
- 14 weeks to tape-out
- 0.6 µm 3 LM CMOS technology
- Maximum die size 50 mm^2
- Pin compatibility
- Focus on ASIC hardware implementation

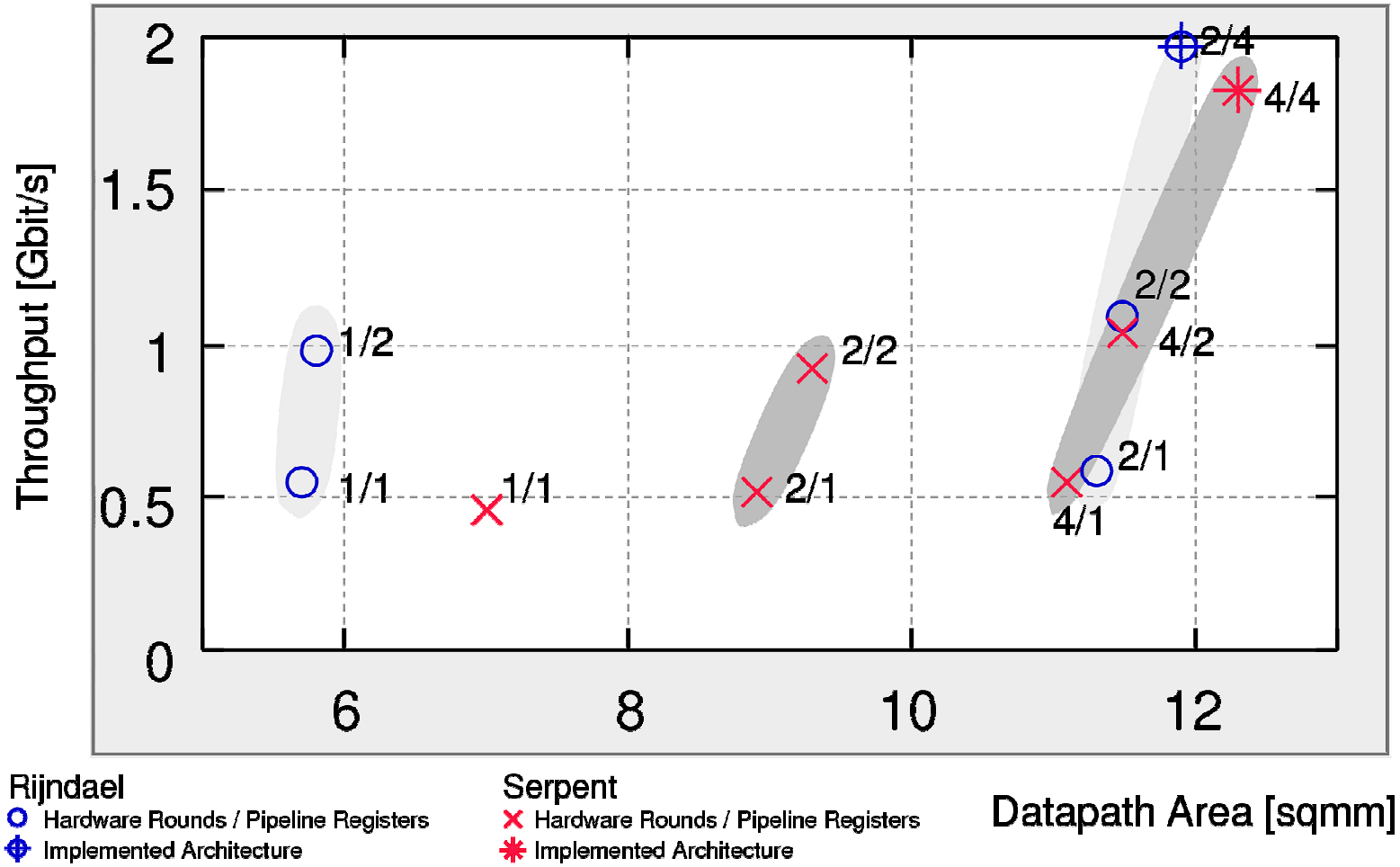
Impact of Architectural Transforms



Variations of Architectural Parameters



Variations of Architectural Parameters



Department of
Information Technology and
Electrical Engineering

Integrated Systems Laboratory

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

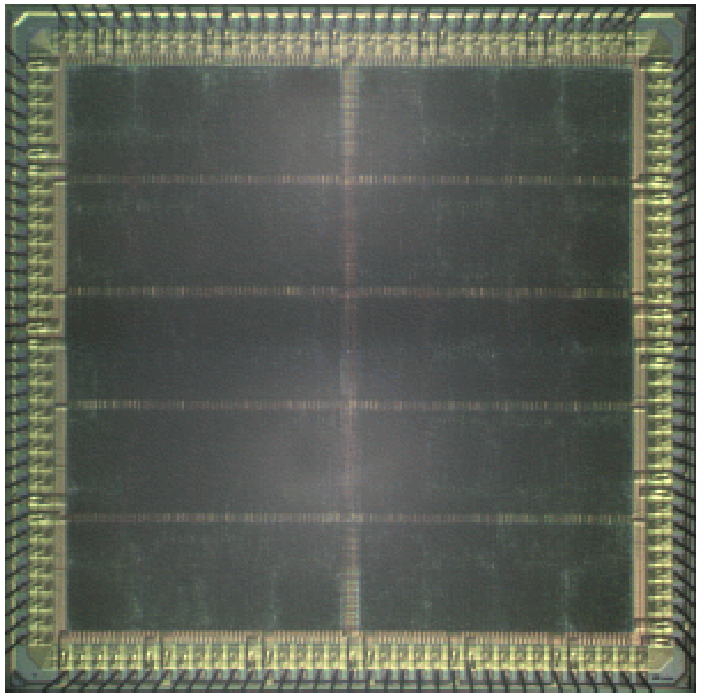
The Rijndael Chip

AES 128bit implementation

Andres Erni

Adrian K. Lutz

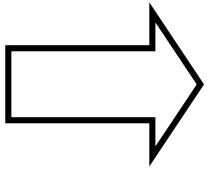
Stefan Reichmuth



Algorithm Summary

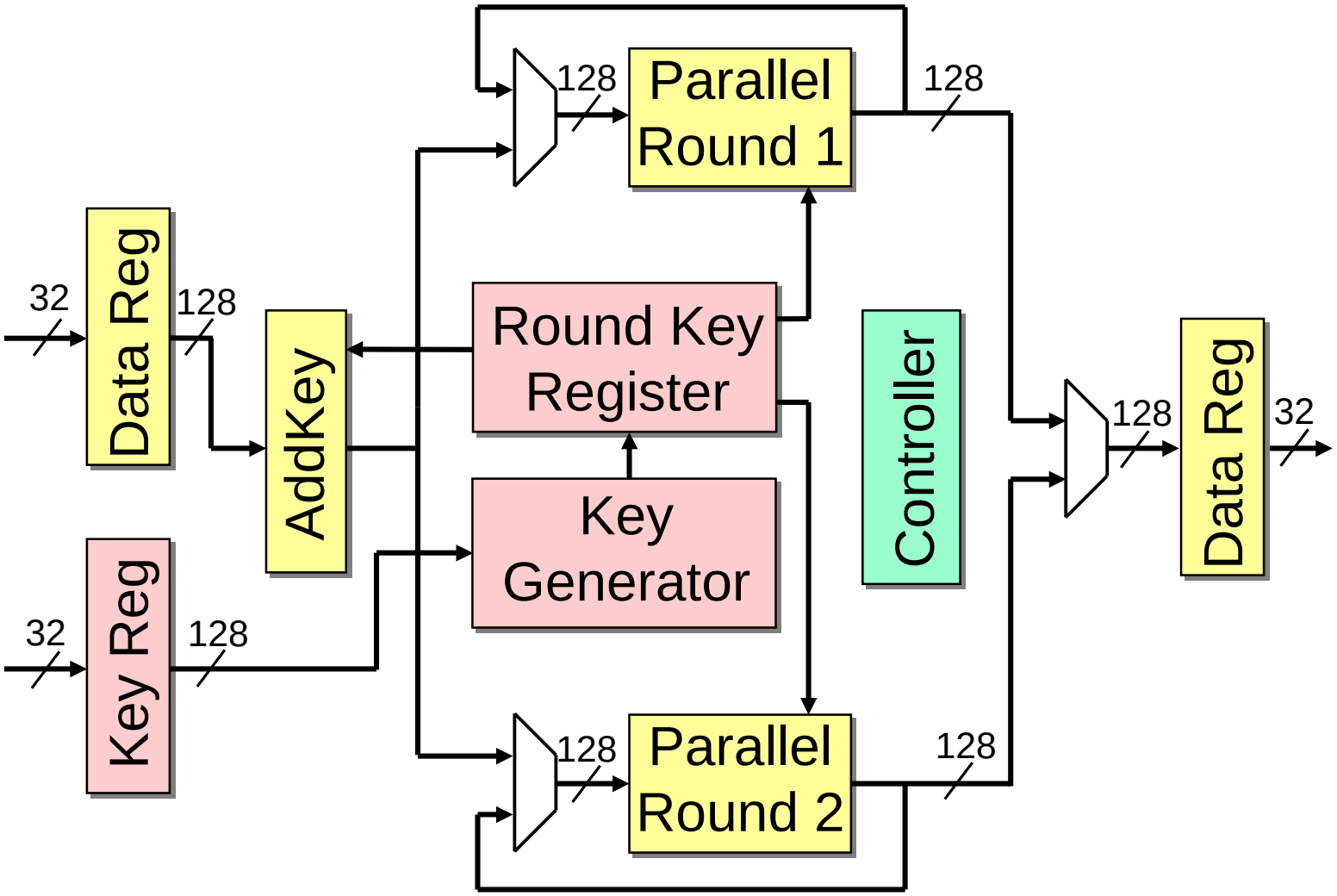
- 128bit user key
- 11 128bit round keys

- 10 rounds
- 8x8-S-box

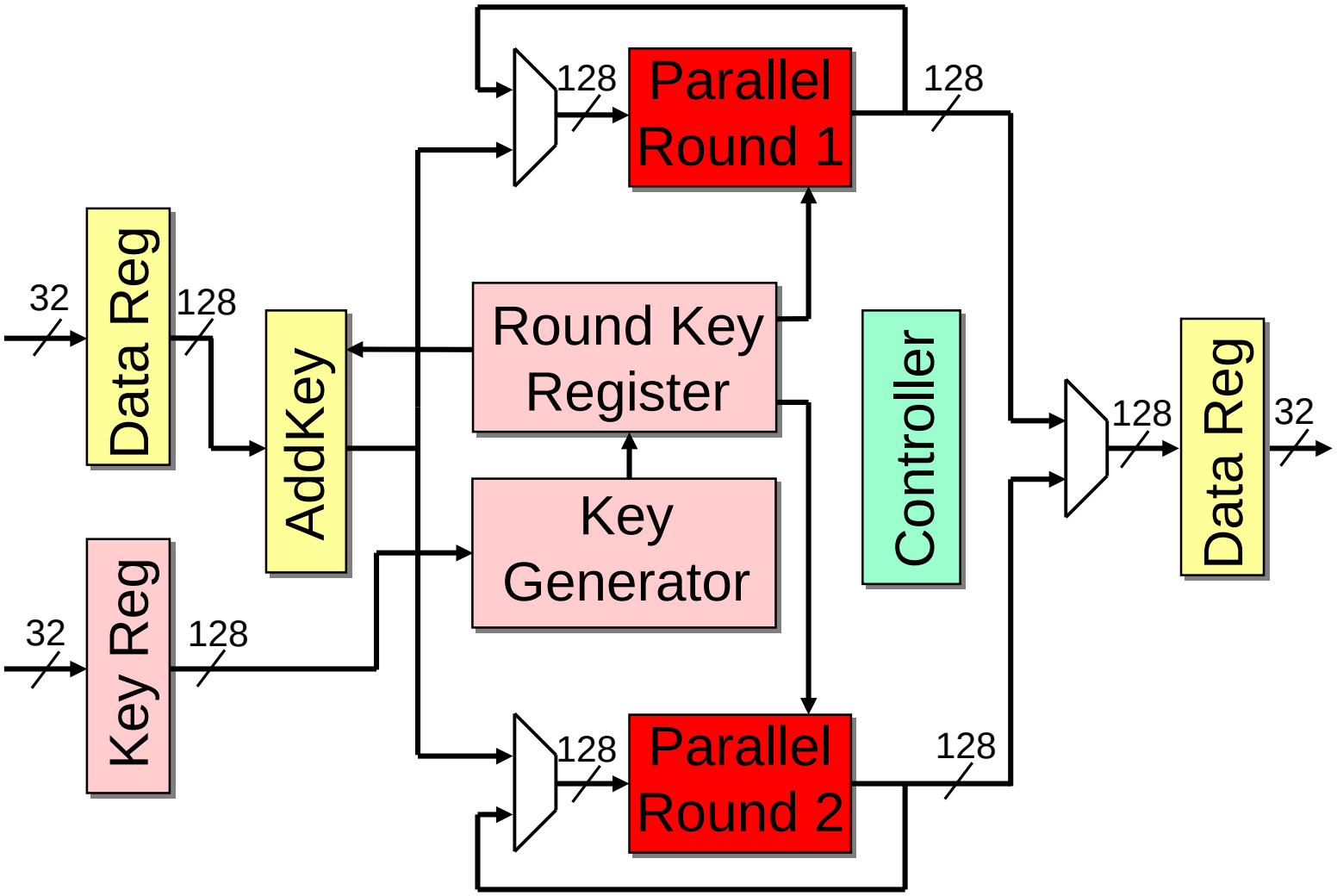


instantiate 1, 2, 5, or 10 rounds

Rijndael Architecture - Overview



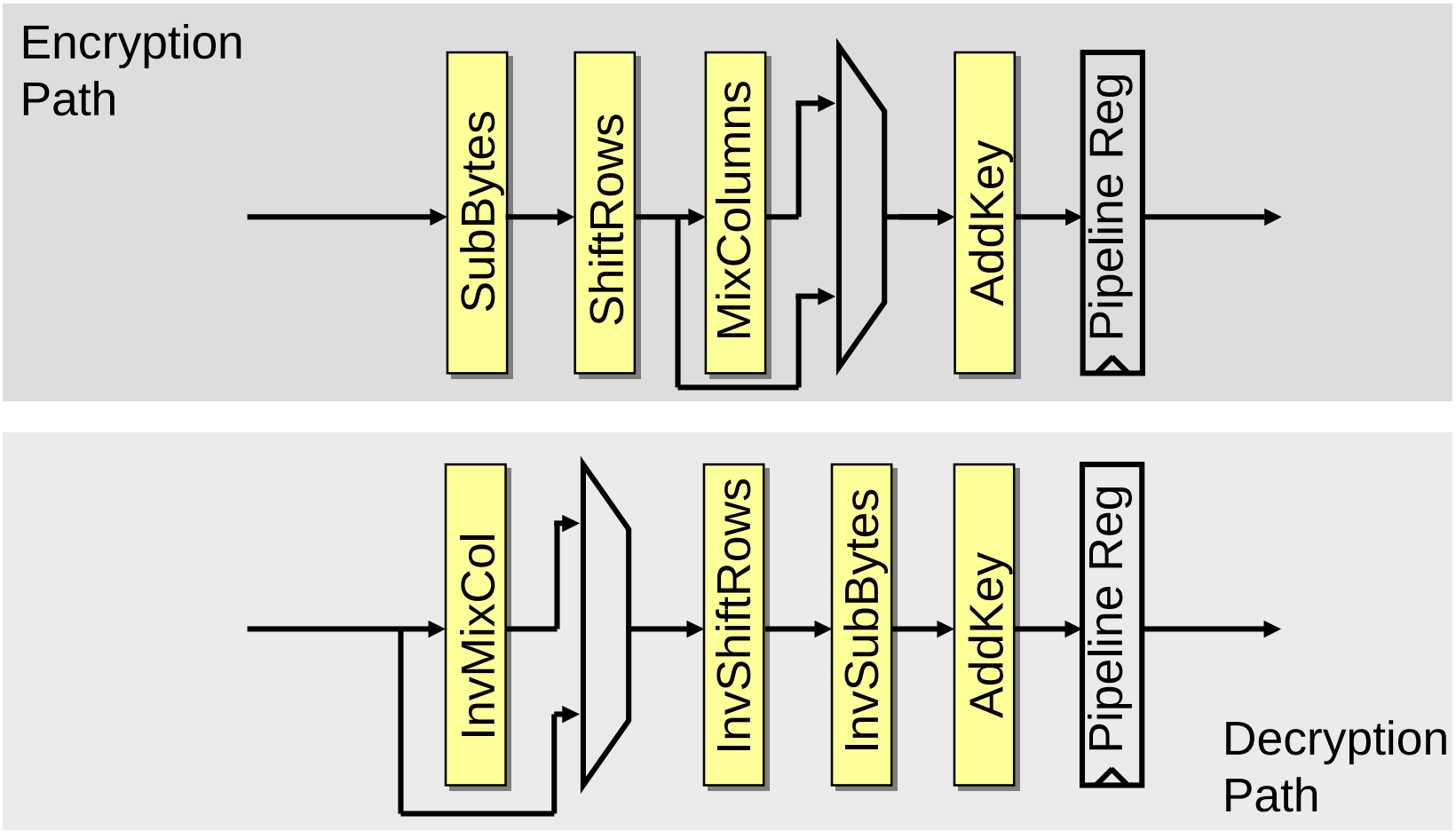
Rijndael Architecture - Overview



Largest potential for optimizations in rounds

Rijndael Architecture - Round

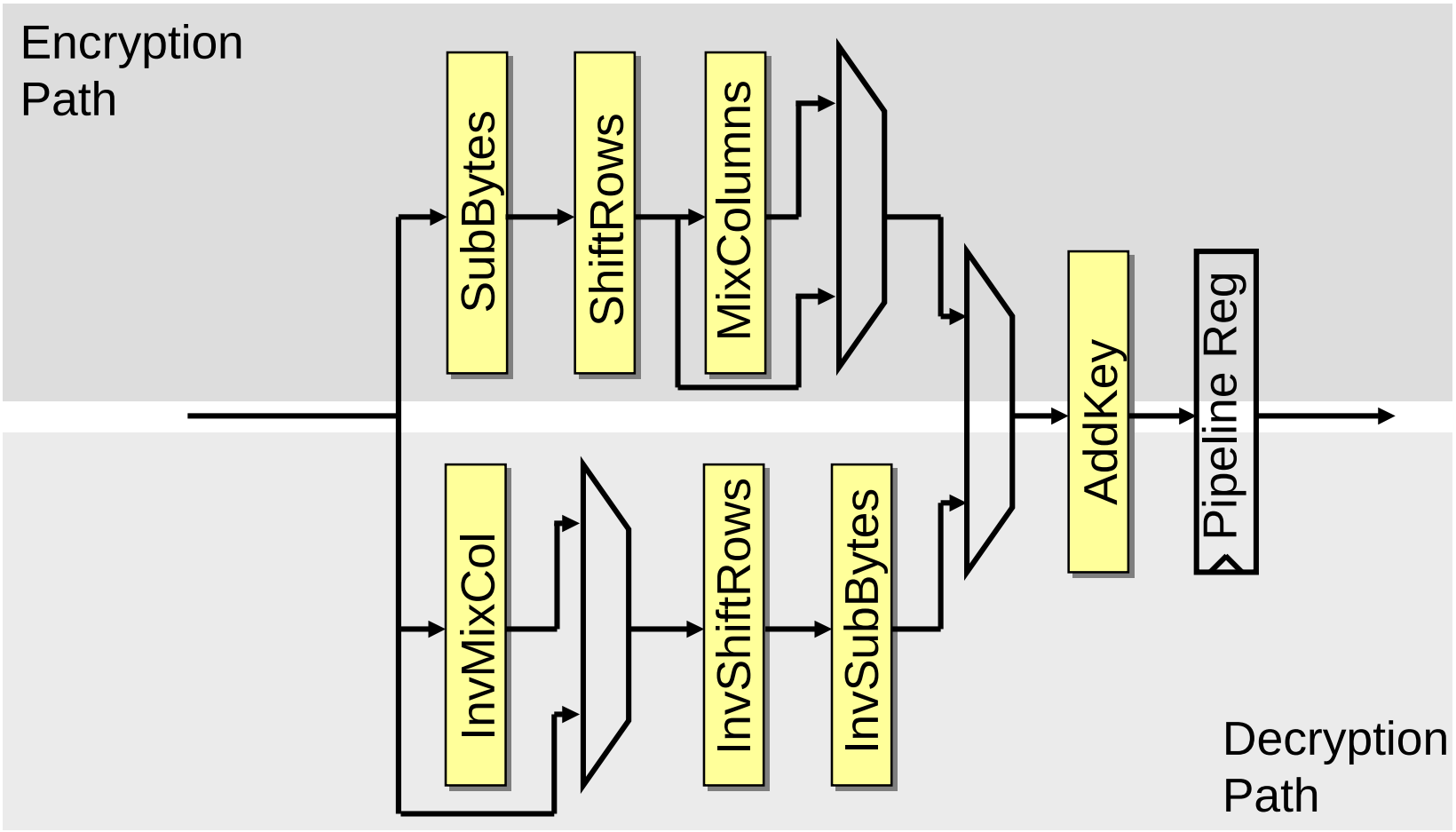
Basic architecture: separate encryption and decryption paths



Problem: Large area requirement

Rijndael Architecture - Round

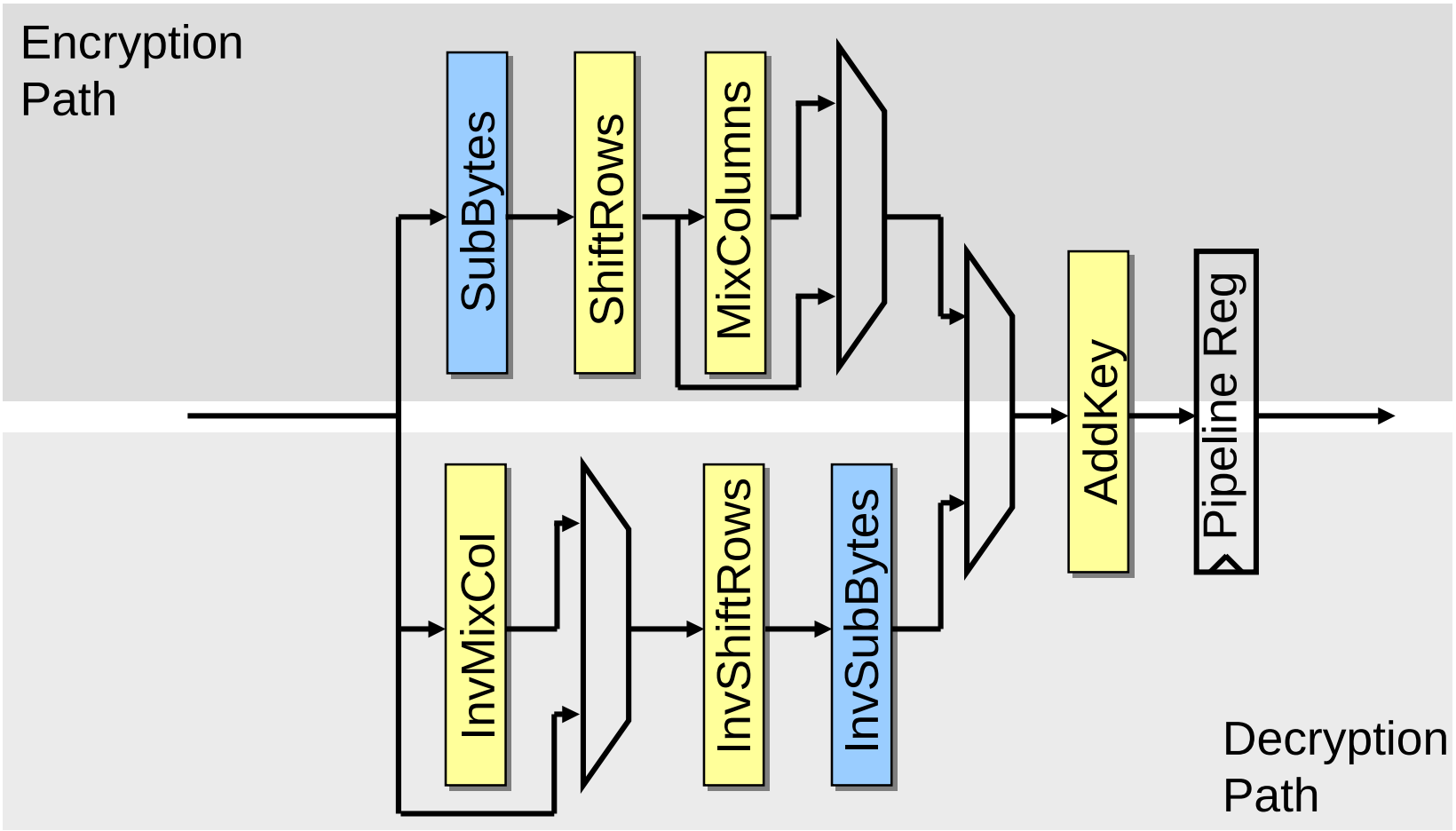
Shared AddKey Function



Problem: Large area requirement

Rijndael Architecture - Round

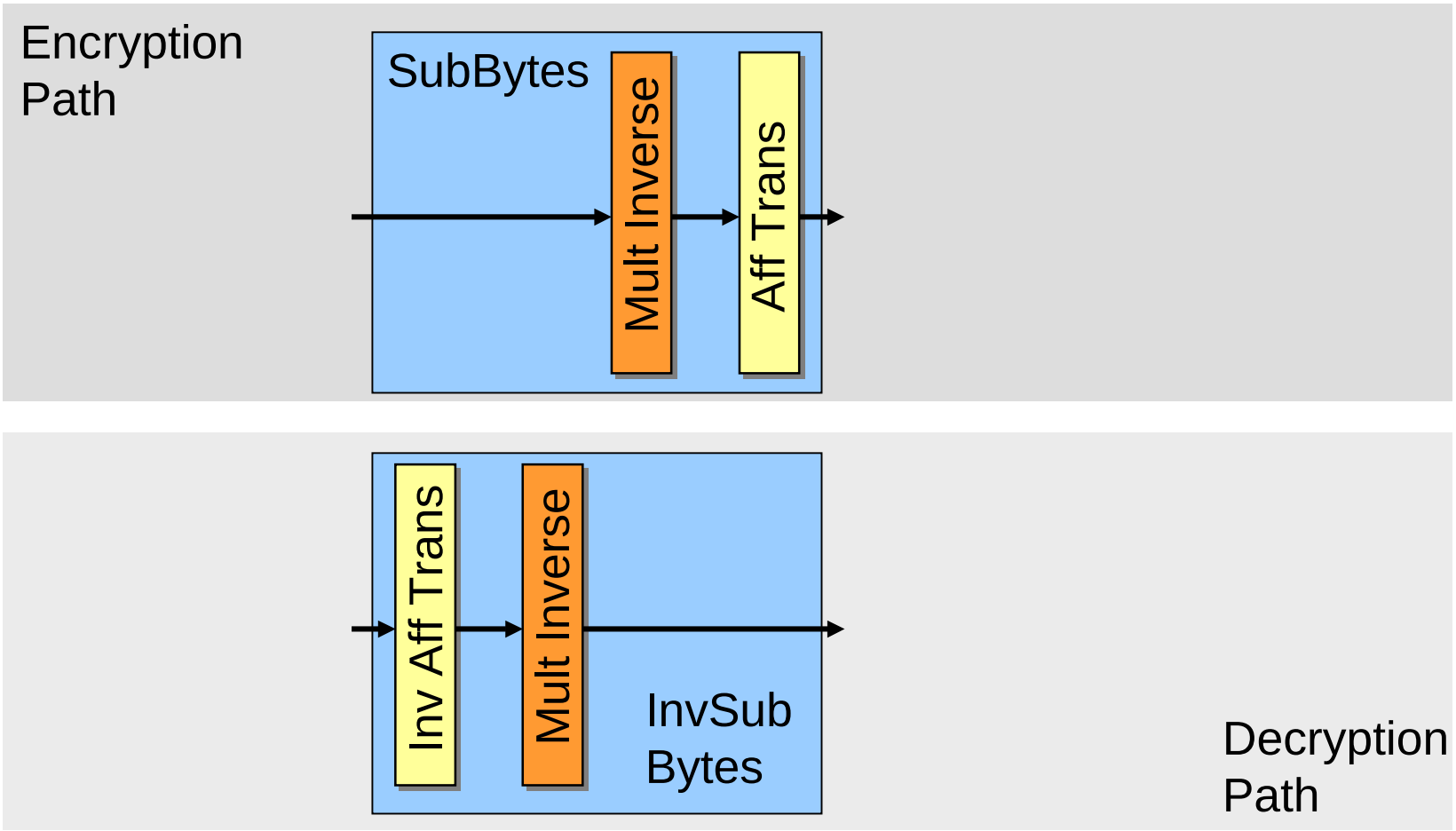
SubBytes and InvSubBytes occupy 85 % of the area



Problem: Large area requirement

Rijndael Architecture - Round

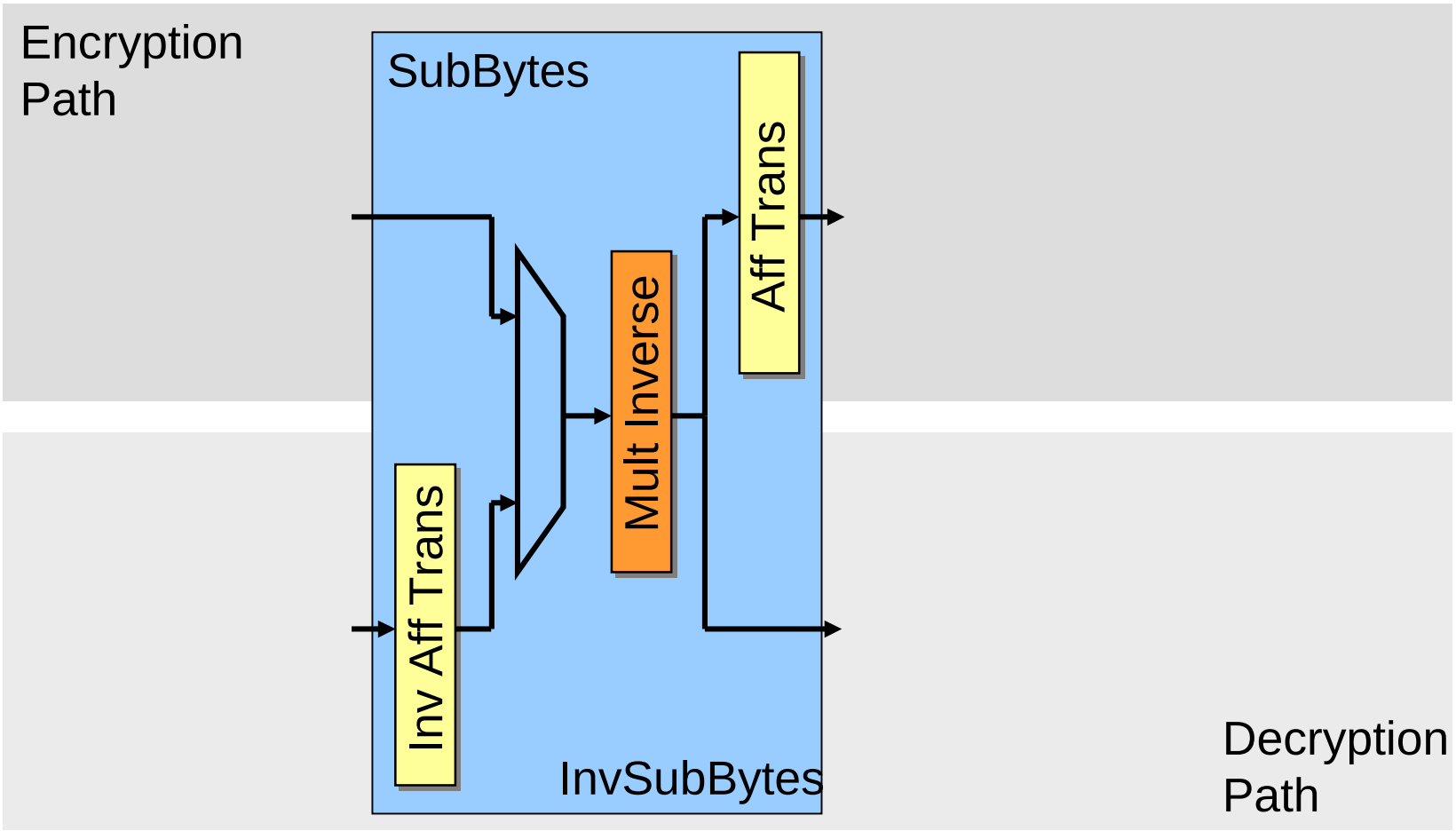
Rijndael S-box consists of two operations



Multiplicative inverse can be shared

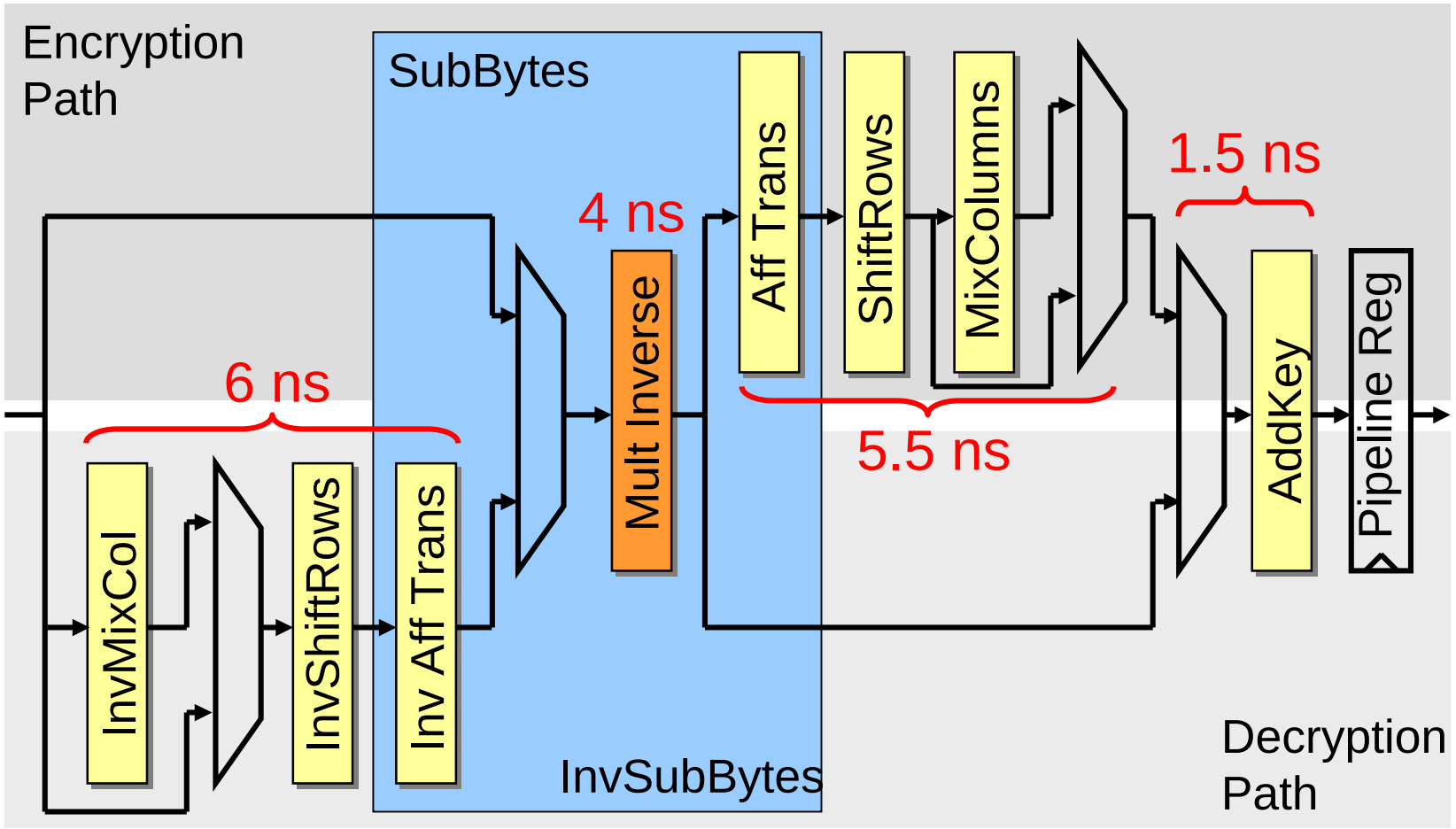
Rijndael Architecture - Round

Sharing Mult Inverse saves 30 % to 50 % of area



Rijndael Architecture - Round

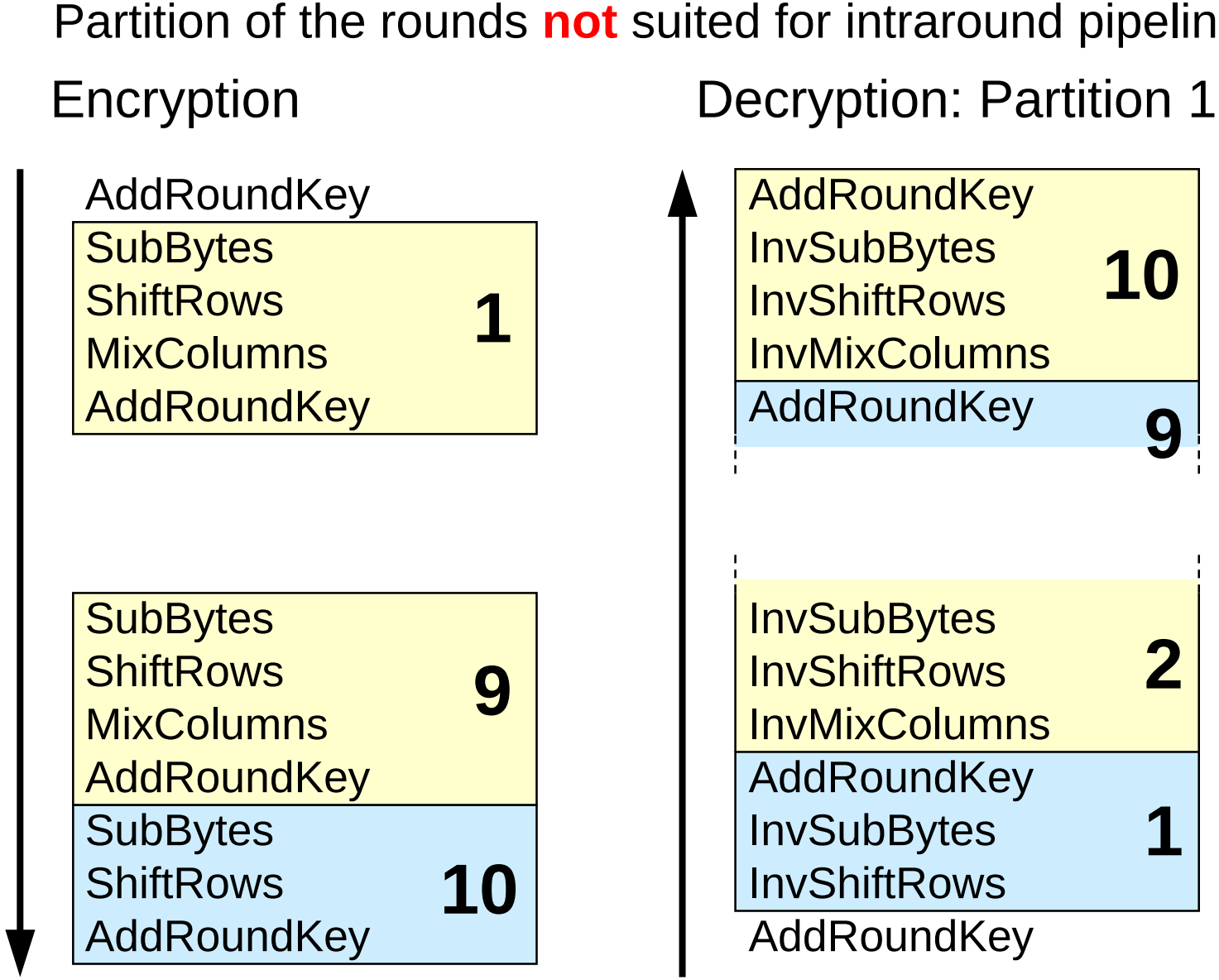
Round architecture with shared Mult Inverse



Problem: Location of intraround pipeline register

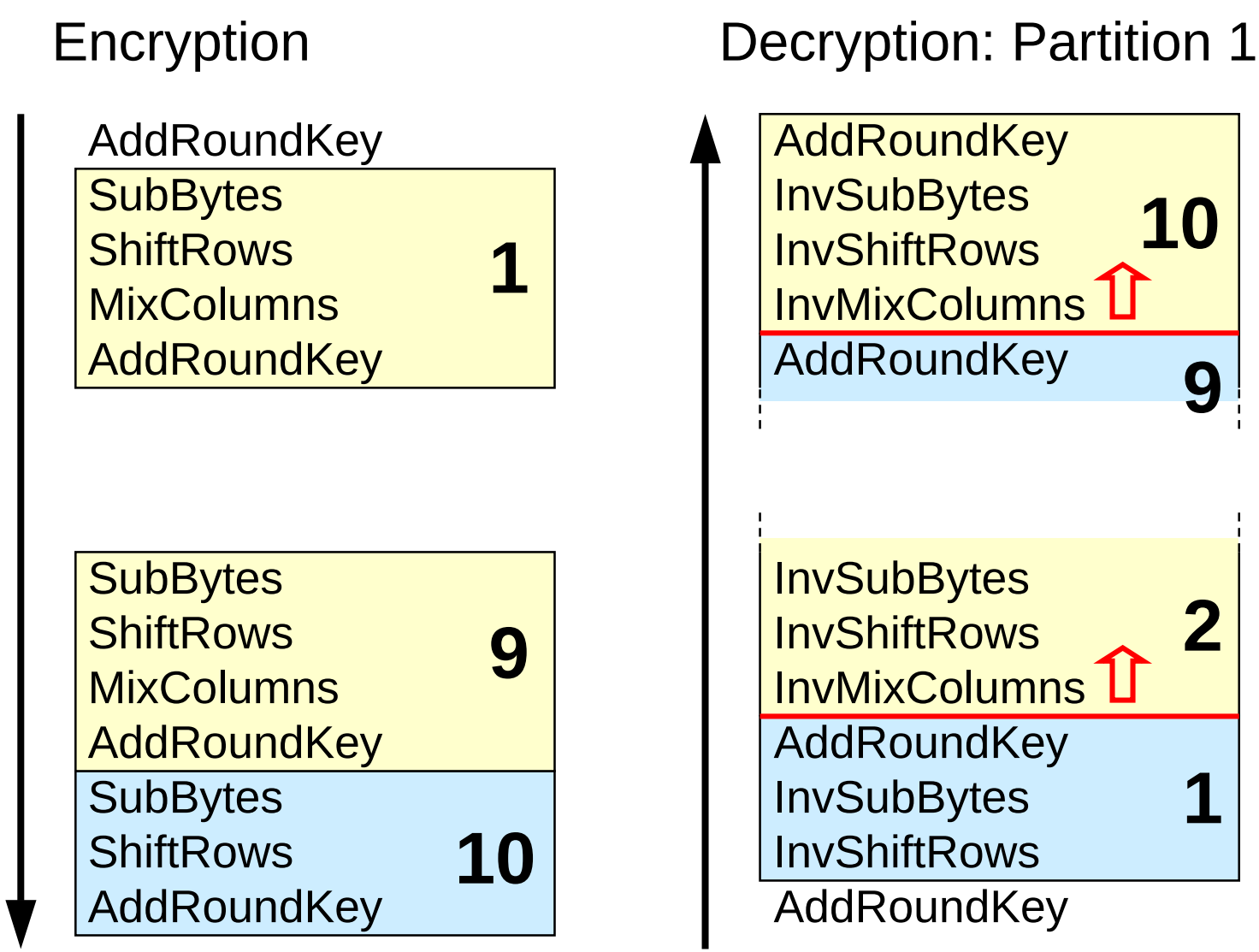
Rijndael Architecture - Round

Partition of the rounds **not** suited for intraround pipelining



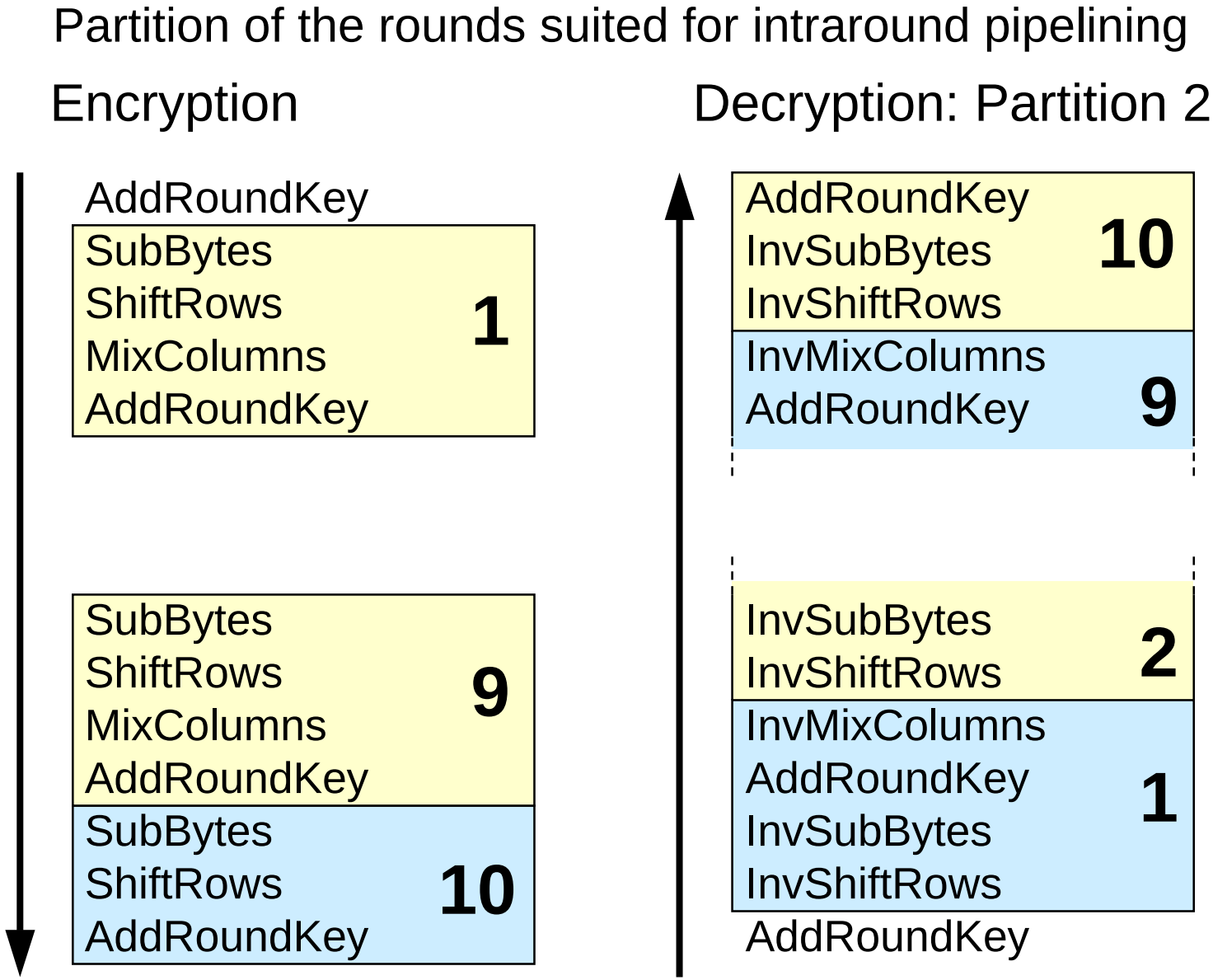
Rijndael Architecture - Round

Shifting round limits makes intraround pipelining possible.



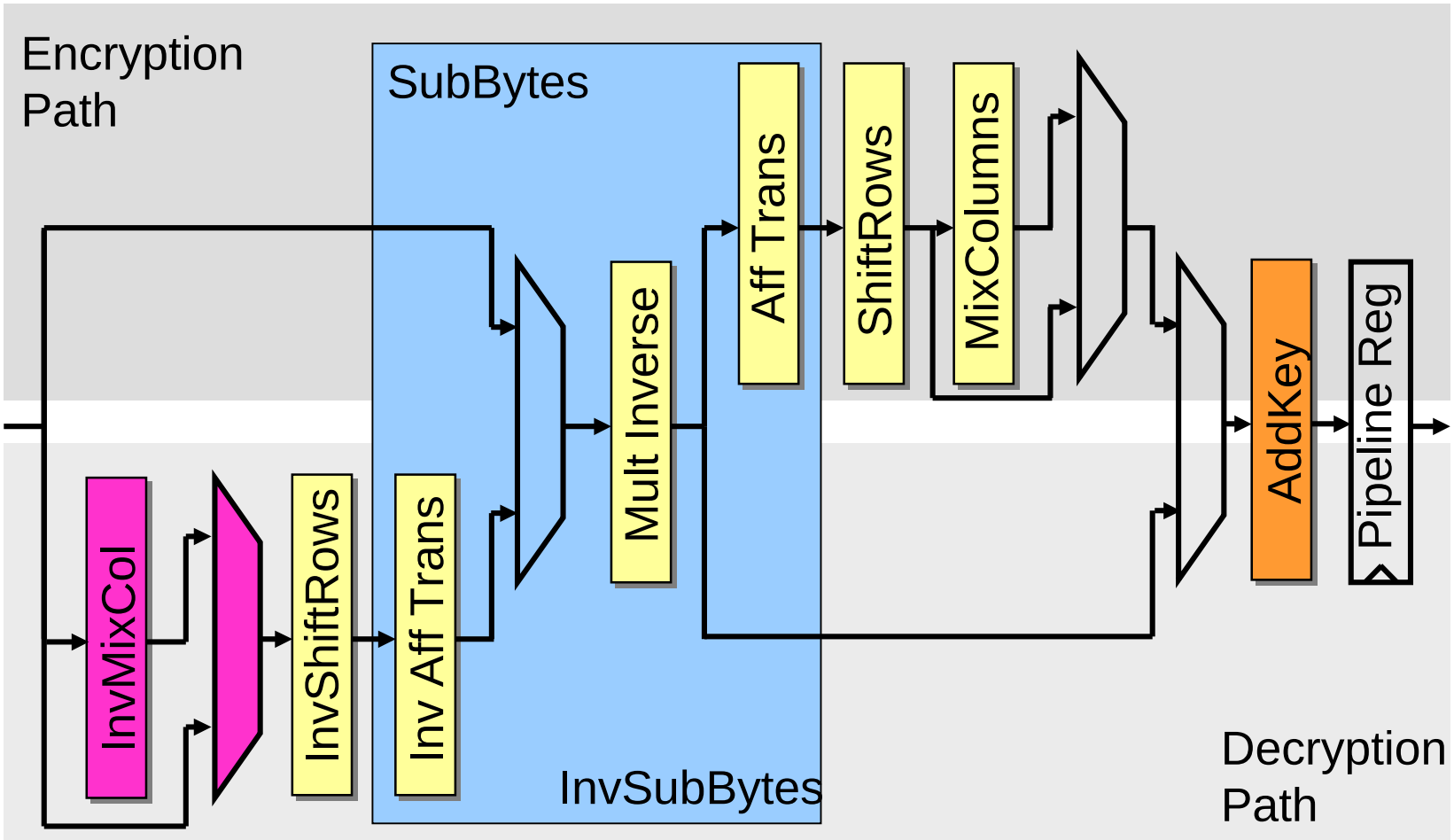
Rijndael Architecture - Round

Partition of the rounds suited for intraround pipelining



Rijndael Architecture - Round

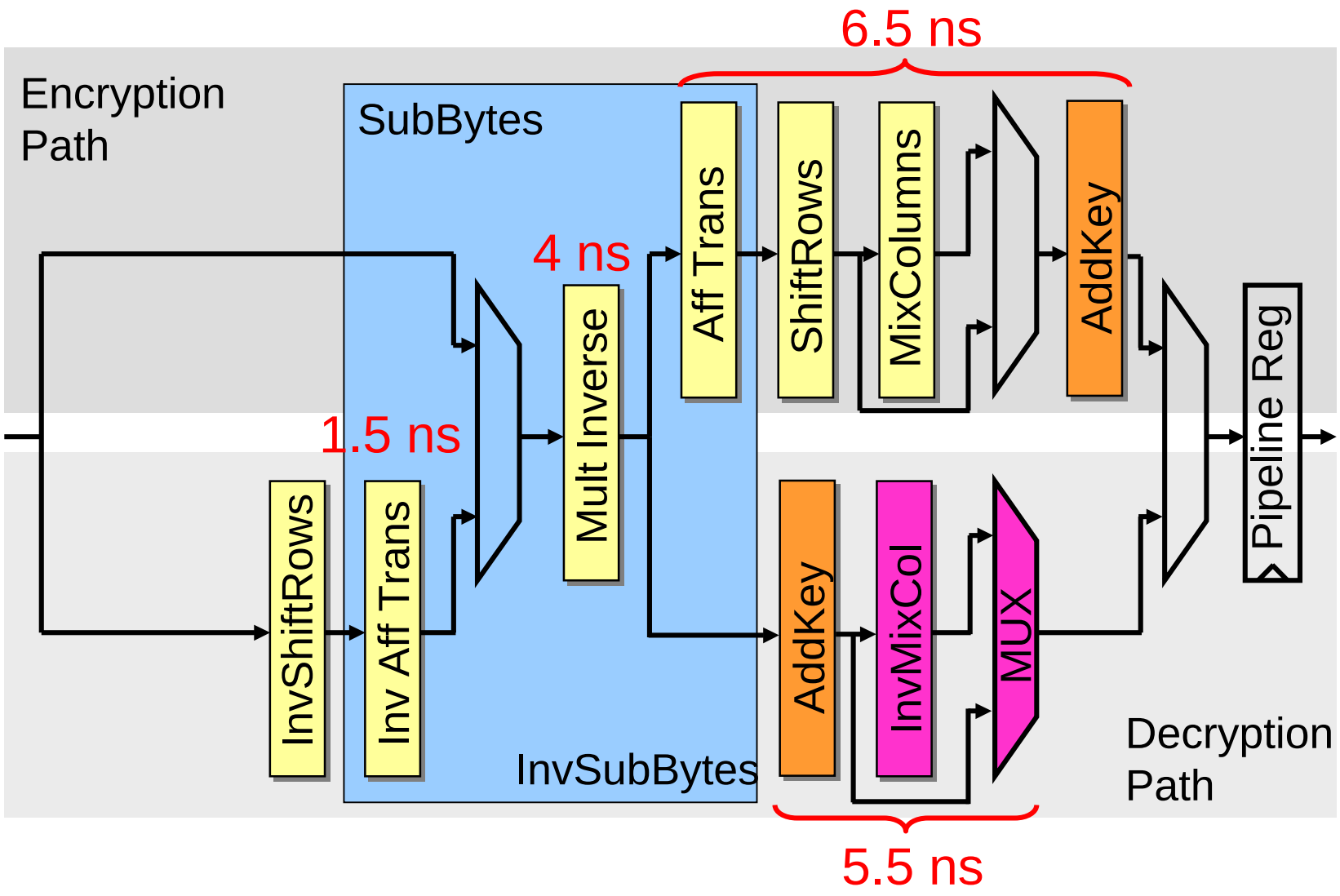
InvMixColumns can be shifted to a better location



AddKey has to be duplicated

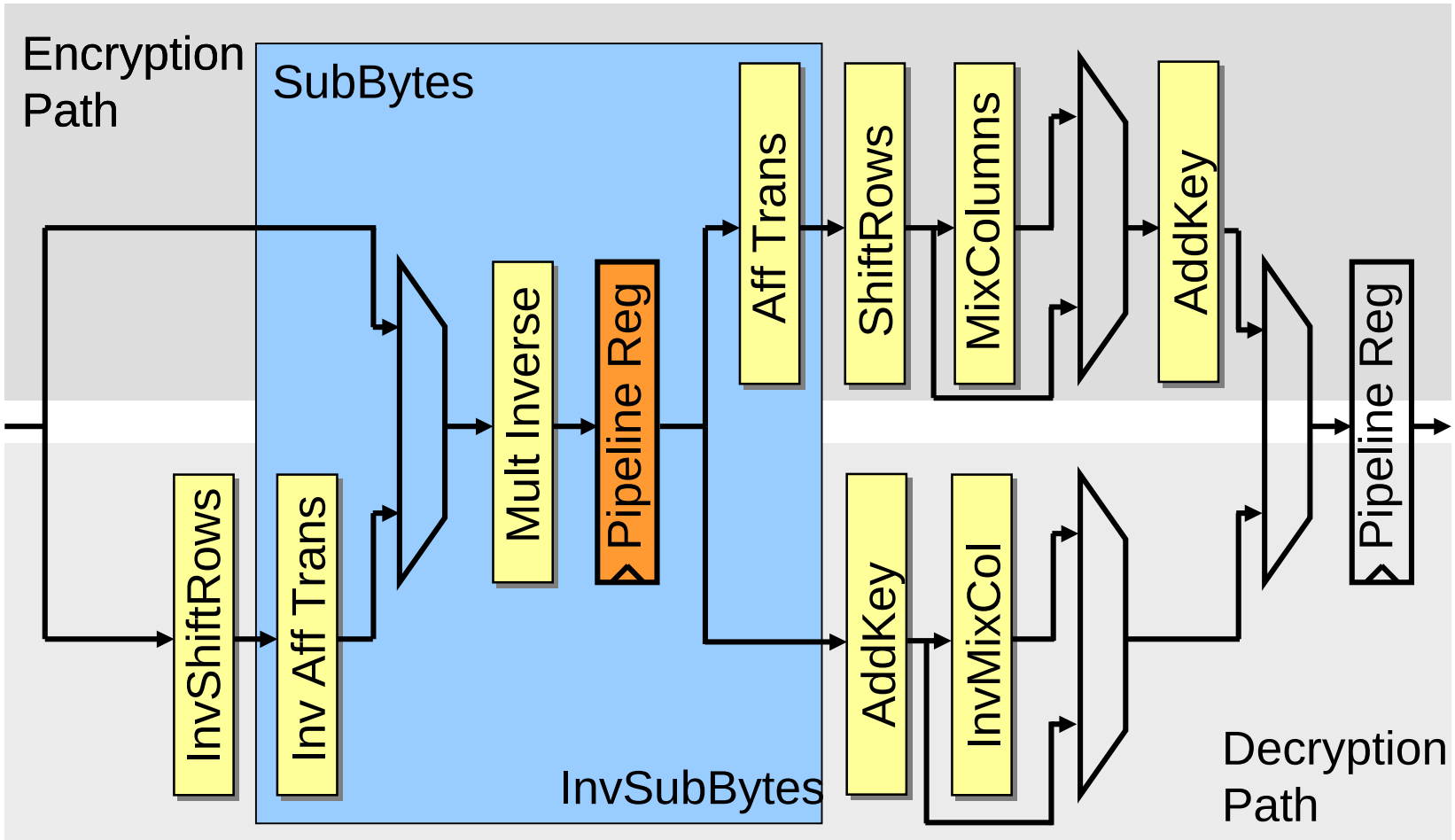
Rijndael Architecture - Round

Architecture suited for intraround pipelining



Rijndael Architecture - Round

Introduction of intraround pipelining



Final architecture implemented in silicon

Department of
Information Technology and
Electrical Engineering

Integrated Systems Laboratory

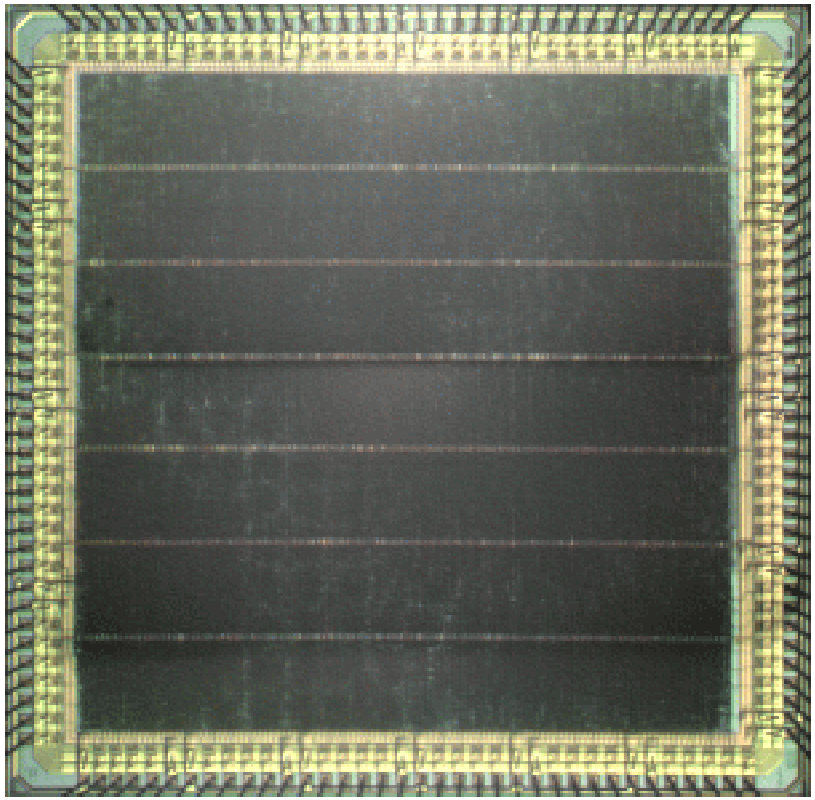
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

The Serpent Chip

Gérard Basler

Pieter Rommens

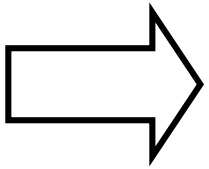
Jürg Treichler



Algorithm Summary

- 256bit user key
- 33 128bit round keys

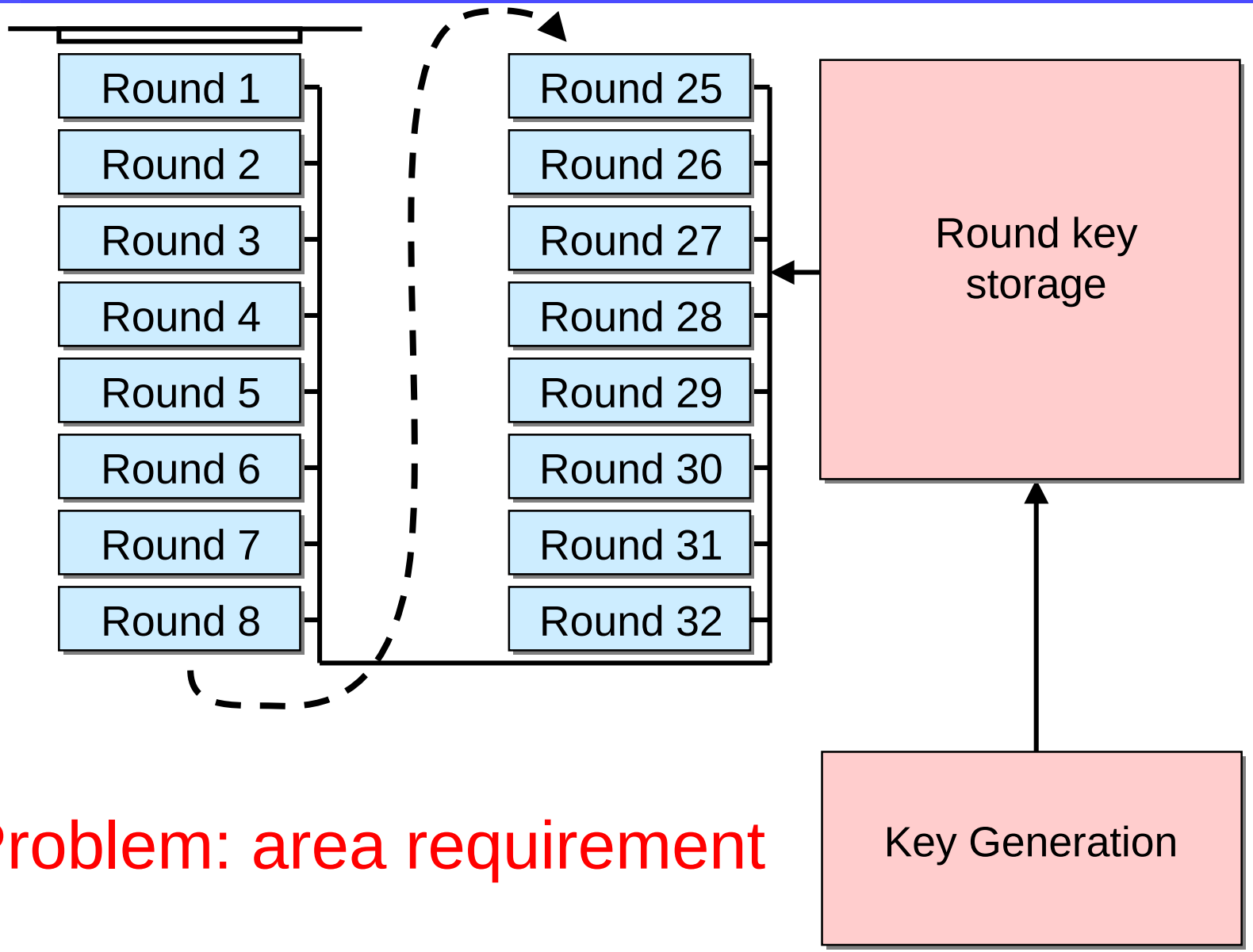
- 32 rounds
- 8 types of 4x4-S-boxes



instantiate preferably 8, 16, or 32 rounds

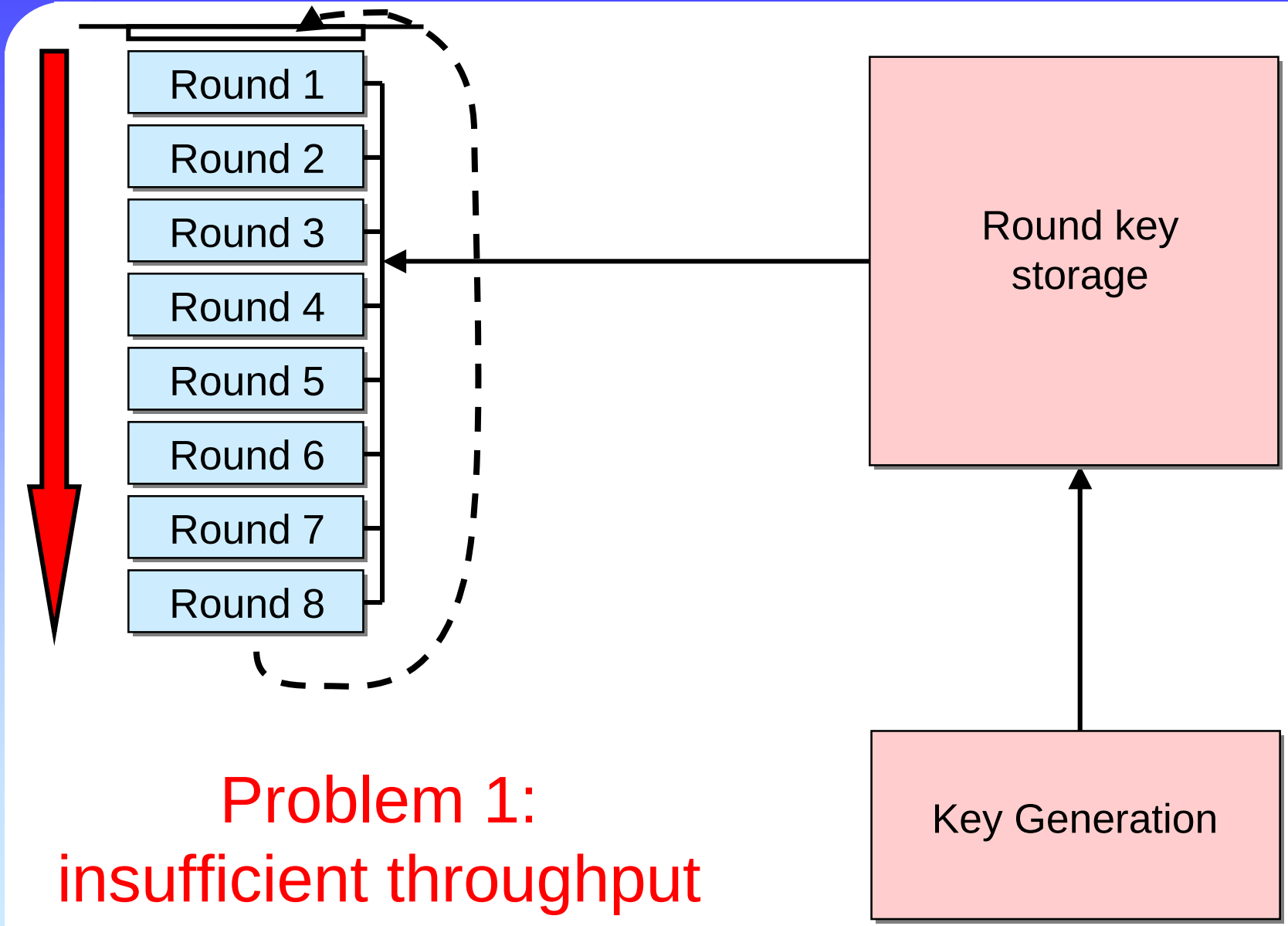
also possible: 1, 2, or 4 rounds

Architecture Development – Step 1



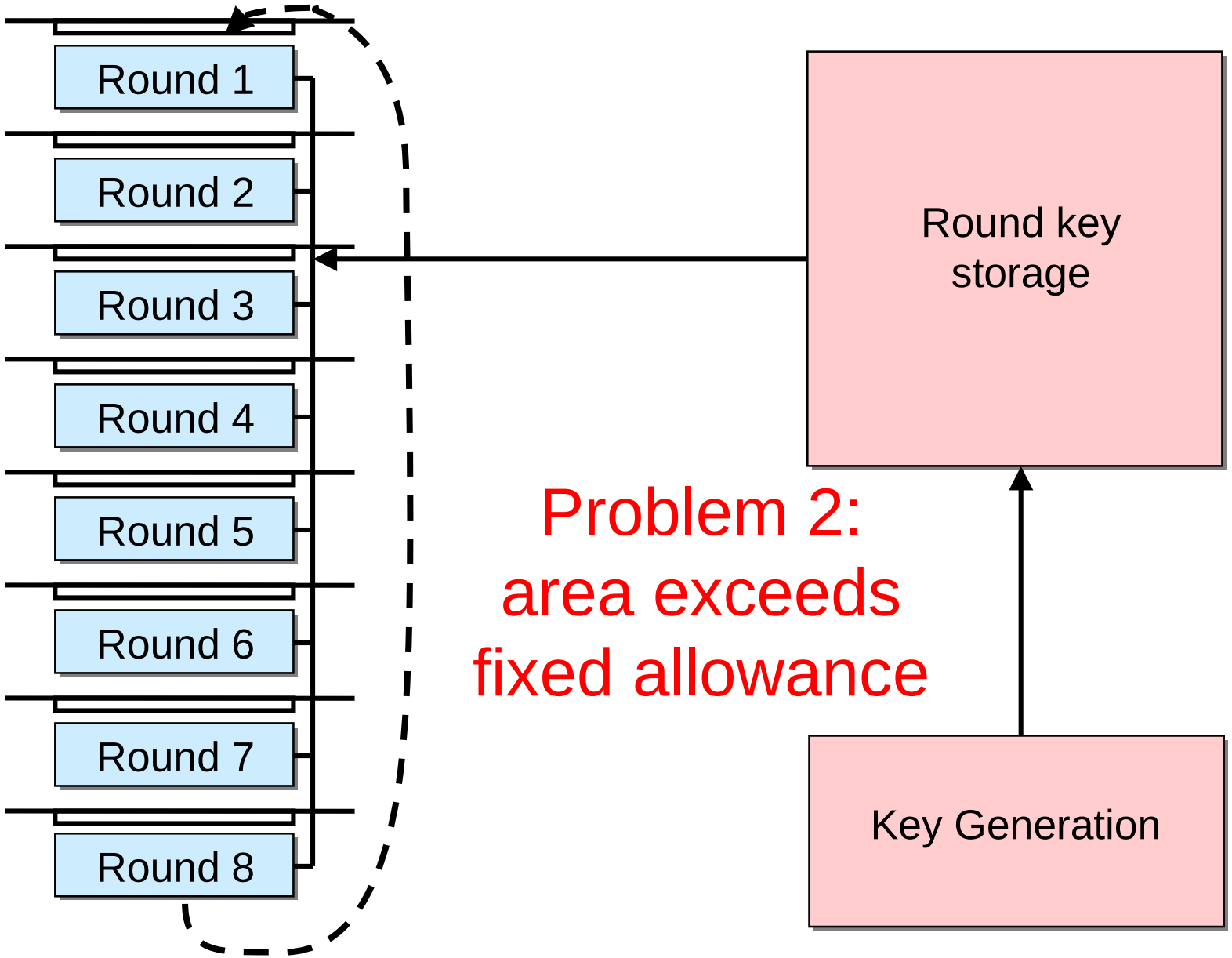
Problem: area requirement

Architecture Development – Step 2

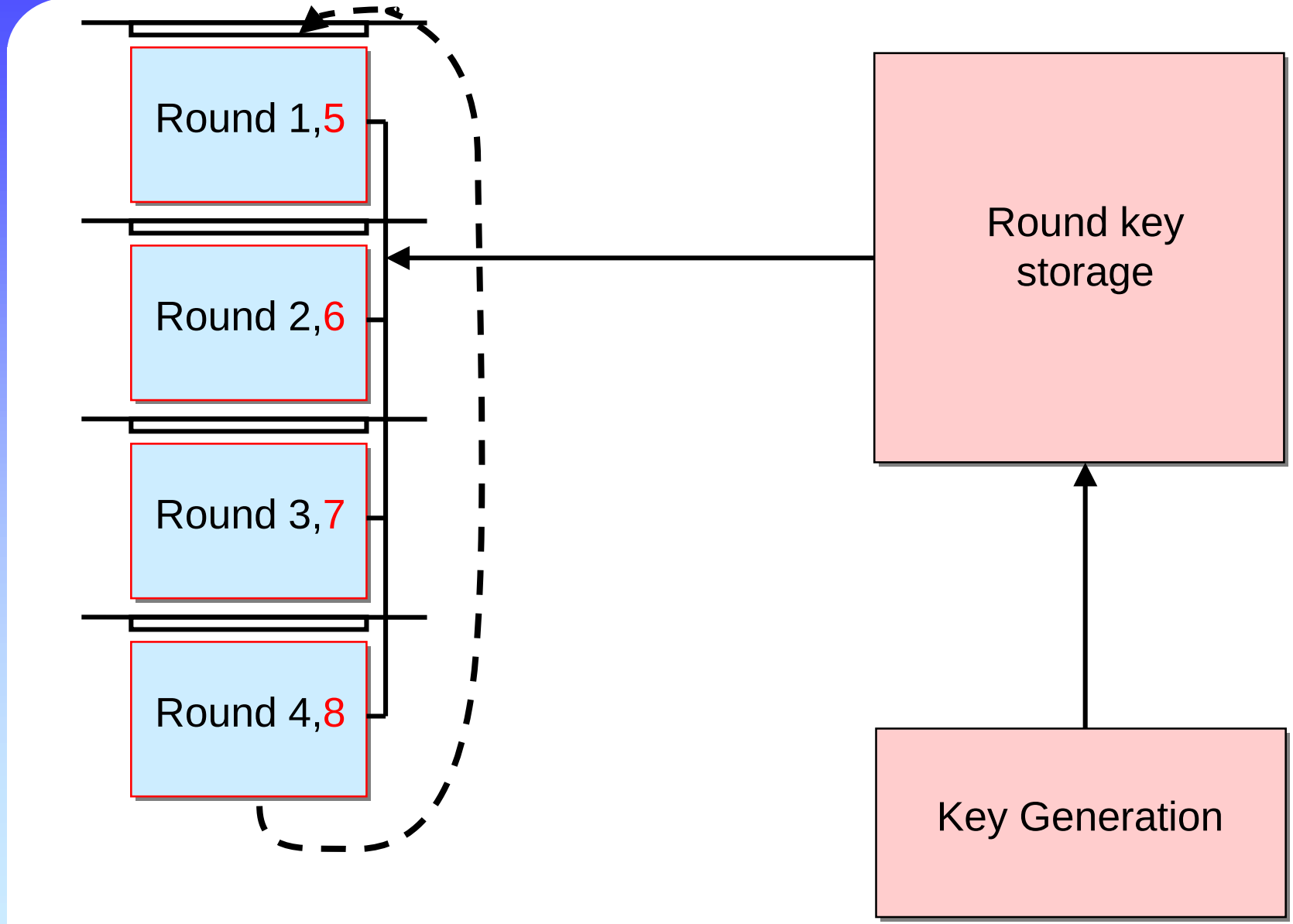


Problem 1:
insufficient throughput

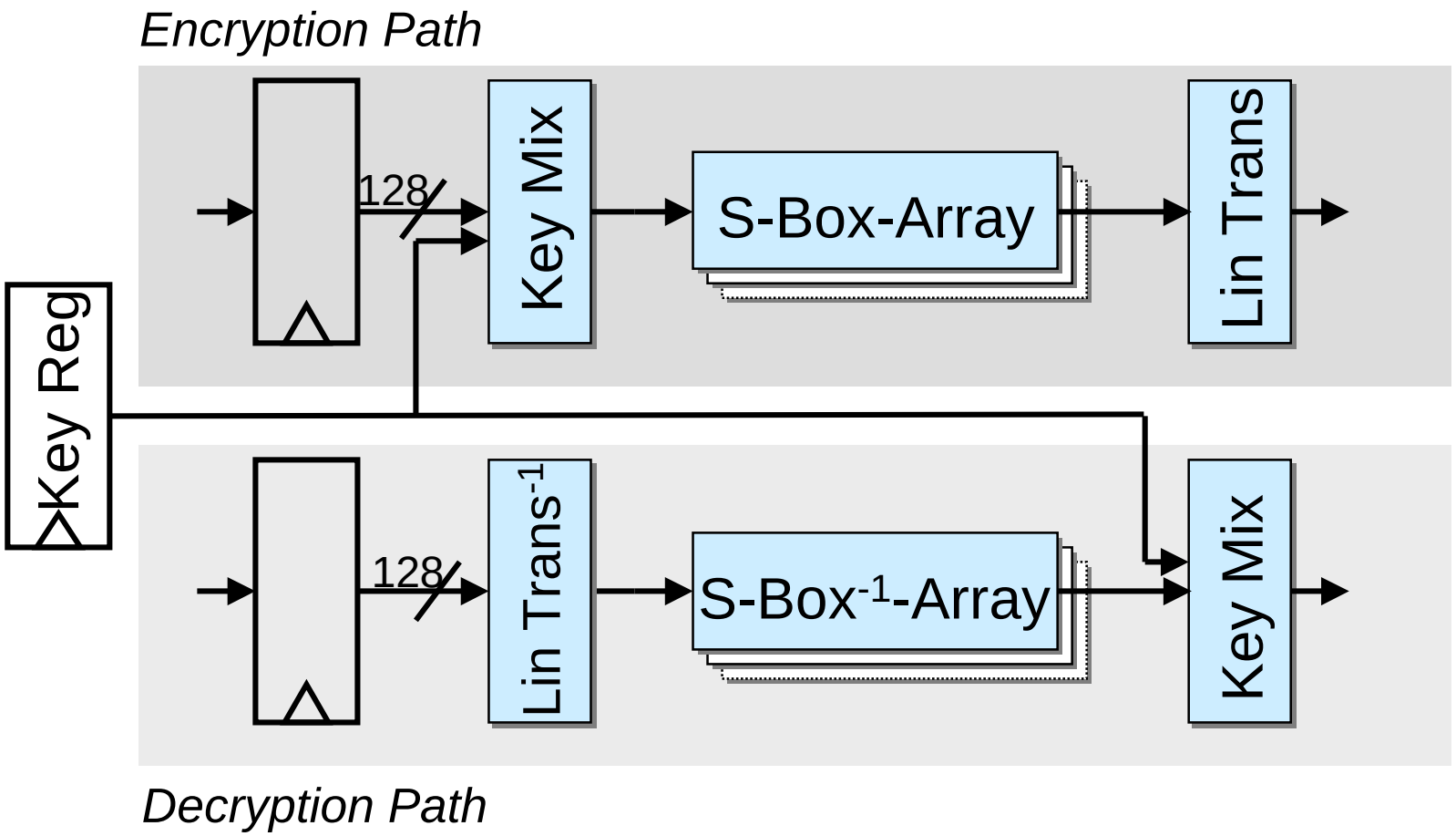
Architecture Development – Step 3



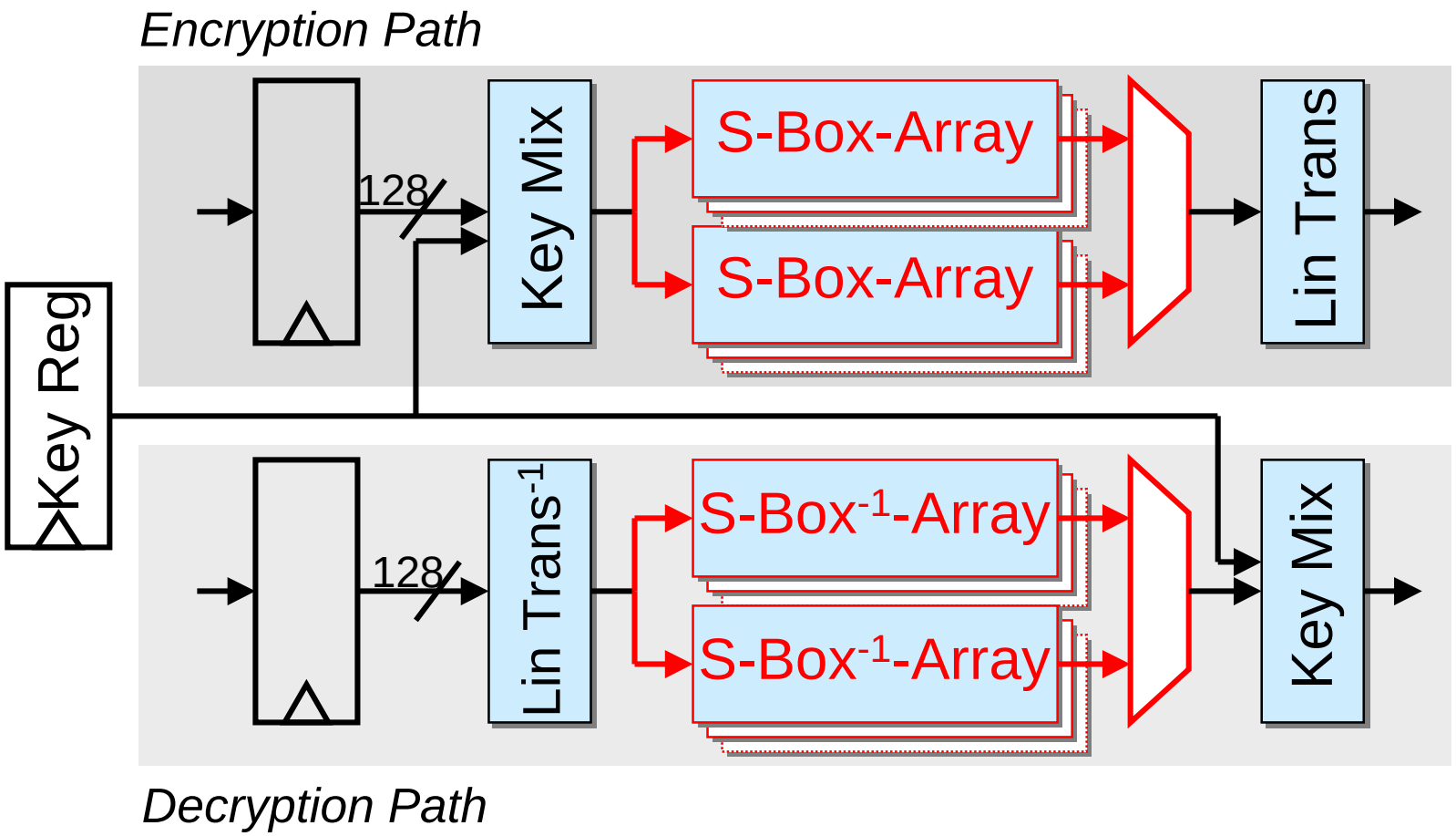
Architecture Development – Step 4



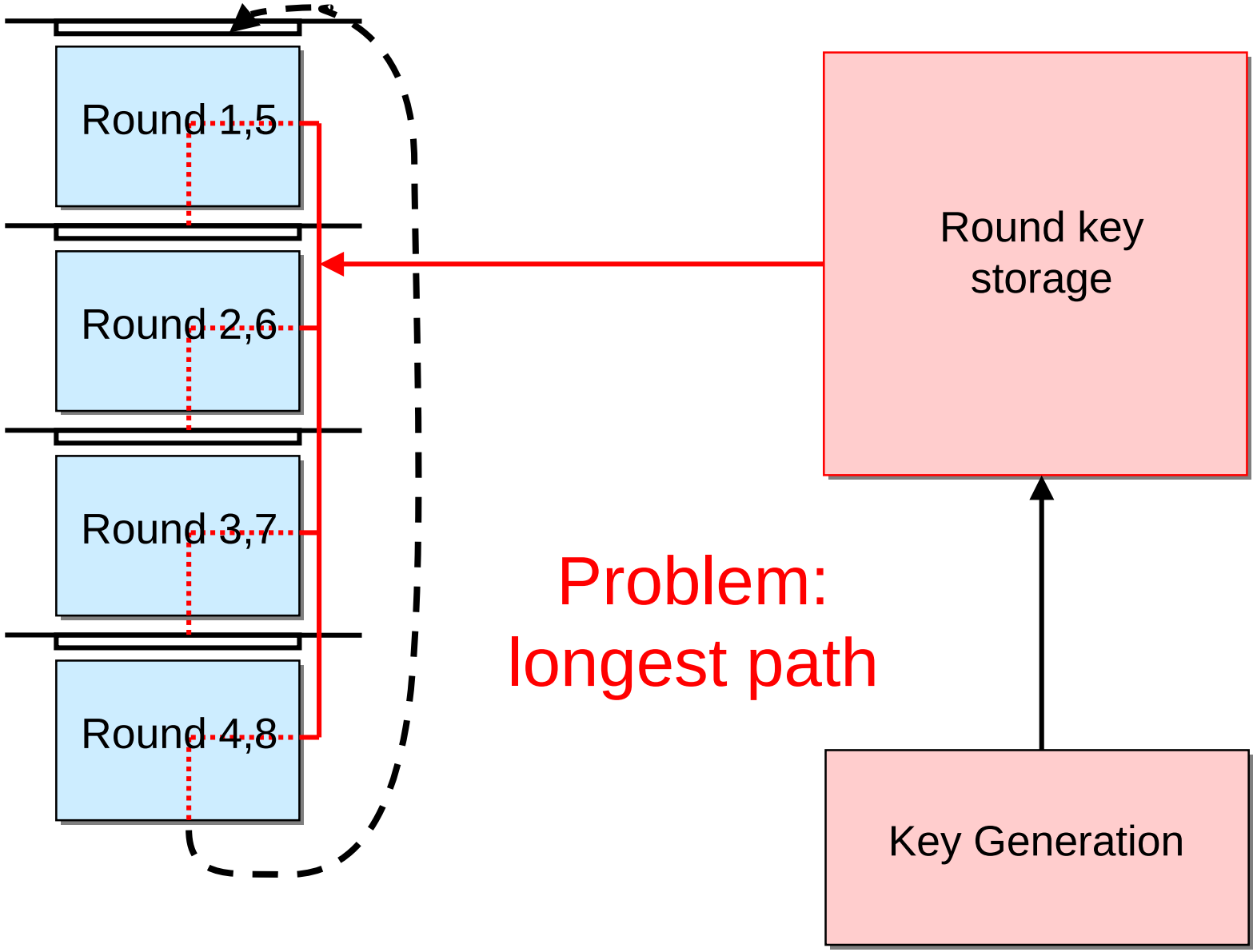
Serpent Architecture - Round



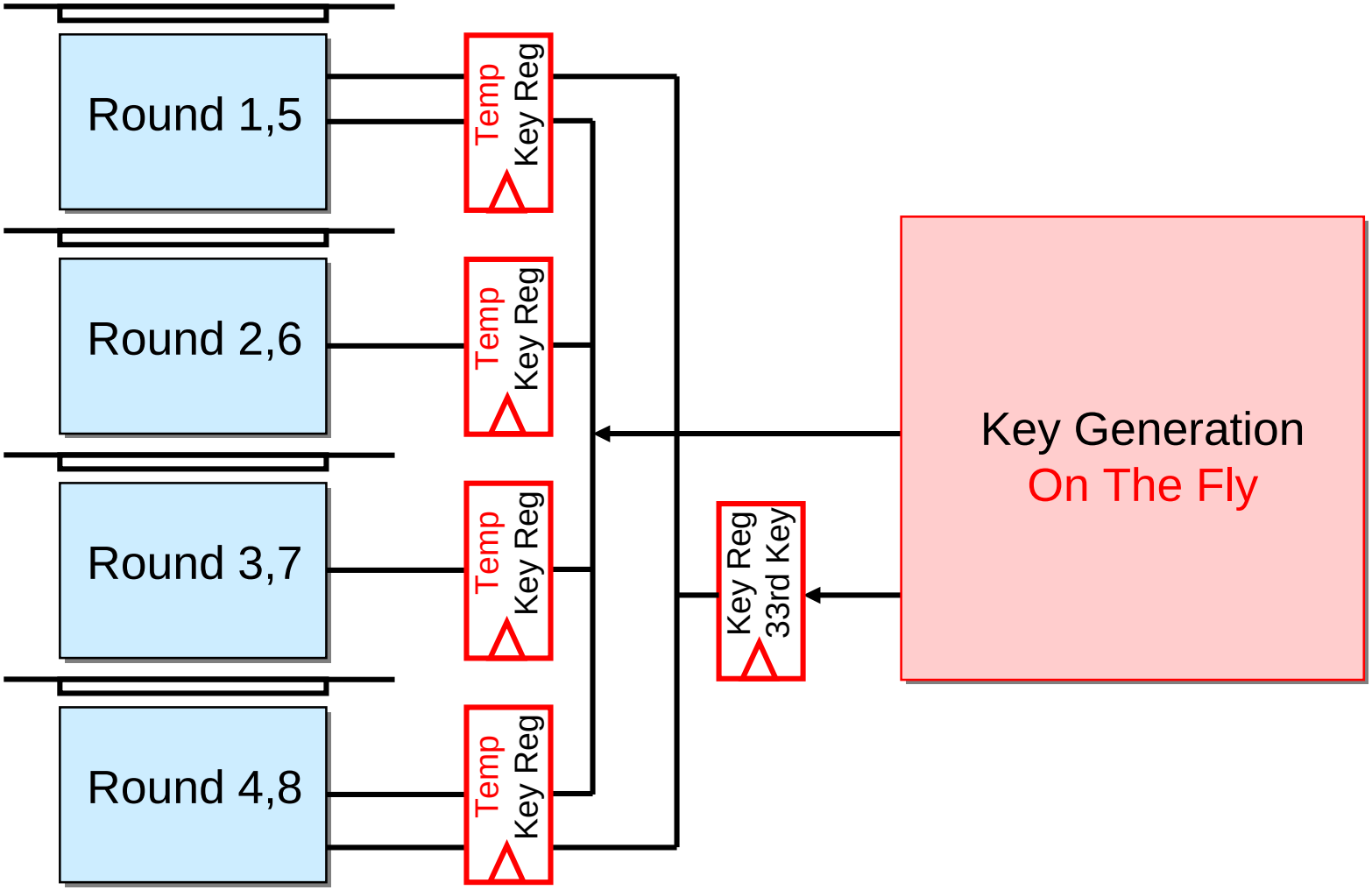
Serpent Architecture - Round



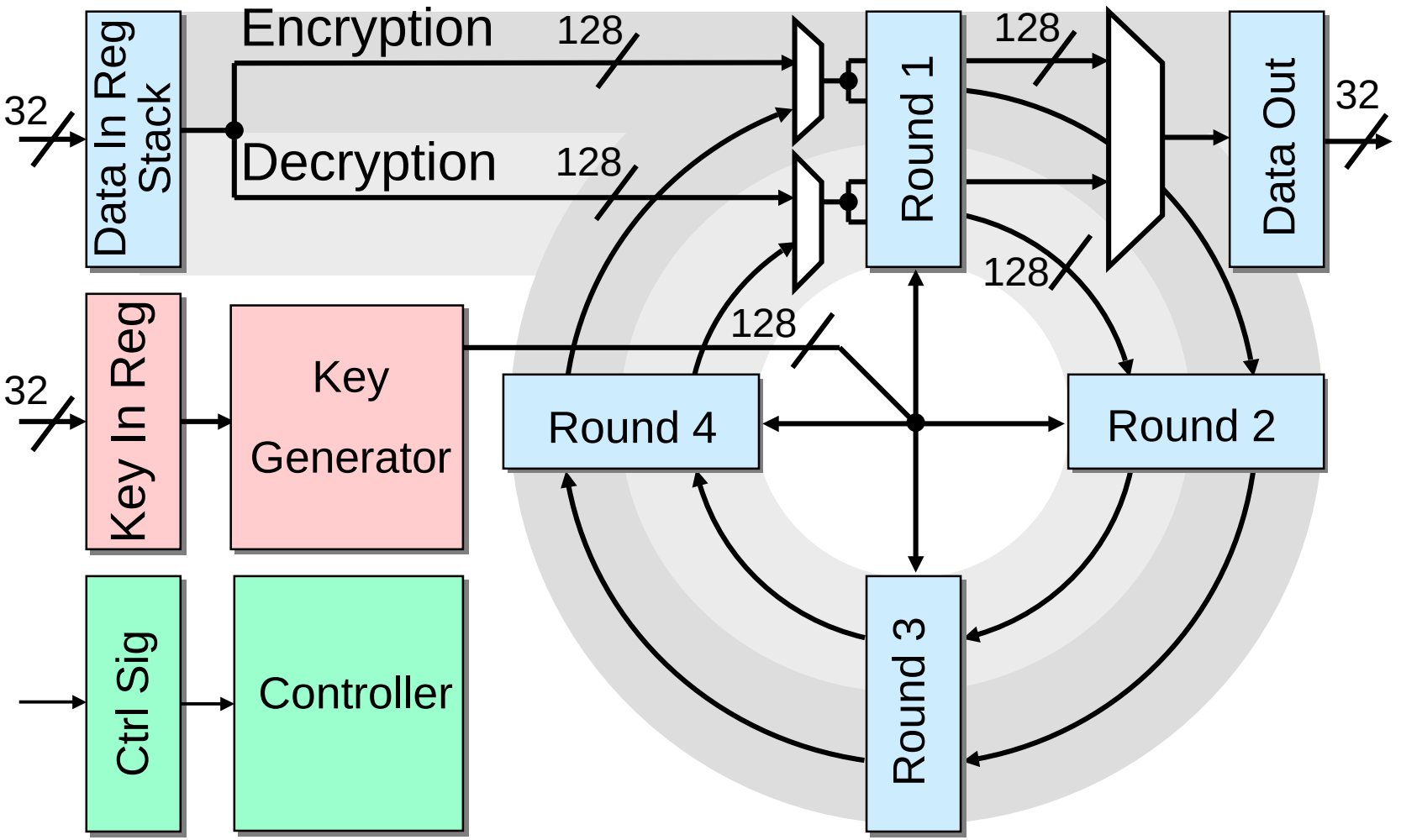
Architecture Development – Step 4



Architecture Development – Step 5



Serpent Architecture – Final



Comparison: Functionality

	Rijndael	Serpent
Rounds in the algorithm	10	32
Data length	128bit	128bit
User key length	128bit	128bit 192bit 256bit
Rounds in hardware	2	4
Round key computation	stored	on the fly

Comparison: Figures of Merit

Comparison based on 128bit keys

	Rijndael	Serpent
Technology, process name	0.6 µm 3LM, AMS CUA	
Area per hardware round (after synthesis)	6.3 mm^2	3.1 mm^2
Area for round key generation	4.5 mm^2	3.8 mm^2
Chip area (estimated, after synthesis)	22.5 mm^2	21.6 mm^2
Chip area (effective)	49 mm^2	49 mm^2
Data throughput (estimated, ECB)	2.16 Gbit/s	1.86 Gbit/s
Data throughput (measured, ECB)	2.26 Gbit/s	1.96 Gbit/s

Conclusions (I)

Both chips were

- designed from scratch
- synthesized
- fabricated
- measured[1]

Throughputs and areas almost identical:

- in ECB and CTR modes: ≈ 2 Gbit/s
- in feedback modes: ≈ 500 Mbit/s

Contributions towards optimum datapath design

[1] Serpent chip: fully functional; Rijndael chip: limited functionality due to error in mask fabrication

Conclusions (II)

Considering that the two algorithms are rather different in nature, their respective performances in hardware come remarkably close.

Rijndael

+ Reuse of the same S-box for all rounds

Serpent

+ Accommodates multiple key lengths with no impact on hardware architecture
+ Number of rounds (32) is hardware-friendly

Thank You for Your Attention

Questions?