

# Low-Latency Encryption in Cyber-Physical Systems: Balancing Security, Efficiency, and Performance



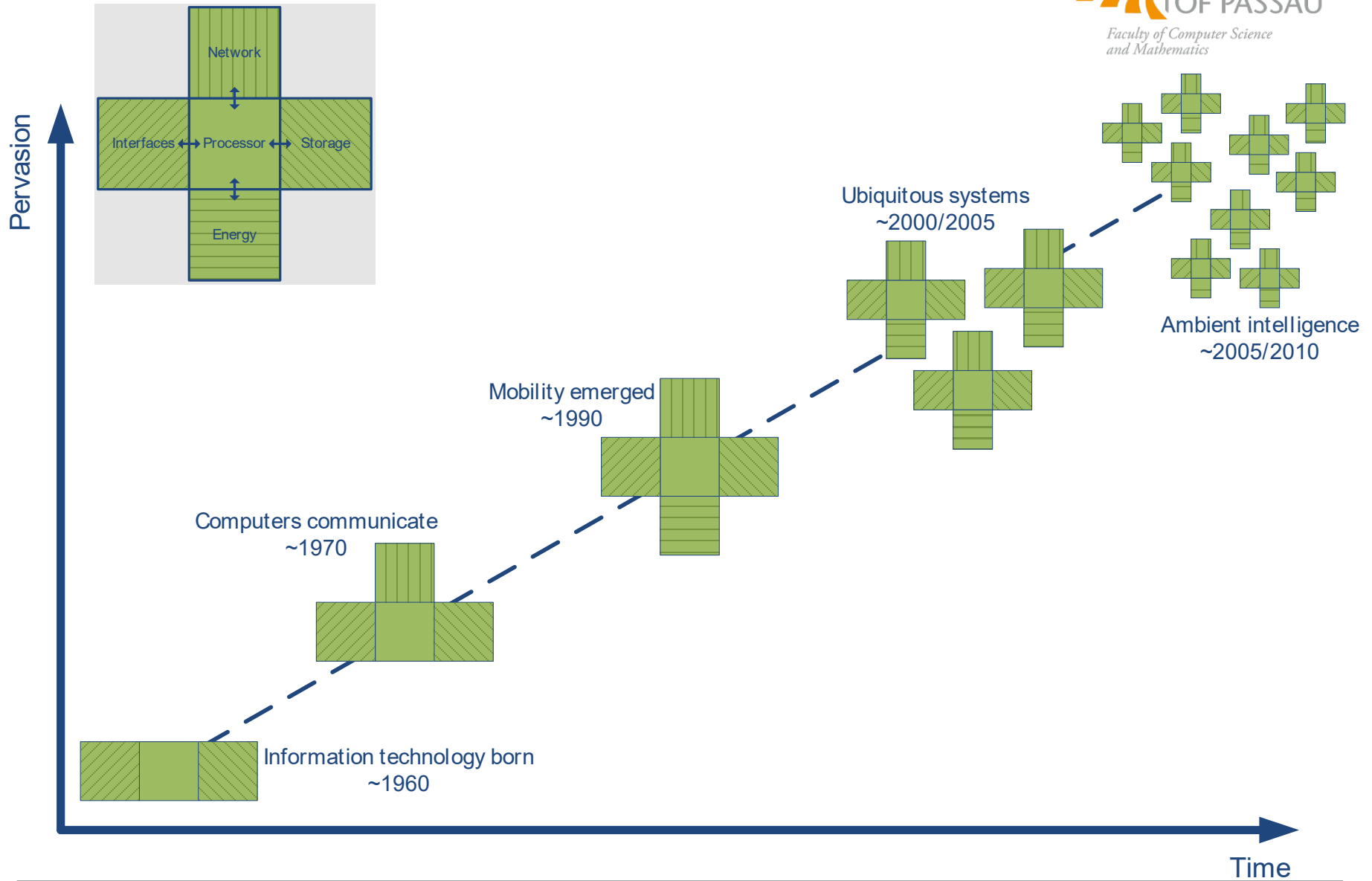
\*internetofbusiness.com

**Elif Bilge Kavun**

September 19, 2024

- Fundamental concepts
  - Need for lightweight security
  - When is low latency a requirement for (crypto) hardware?
  - How to achieve?
- Challenges of low latency crypto on hardware implementations
- Existing low-latency symmetric encryption ciphers
  - PRINCE and PRINCEv2 (2012 and 2020)
  - Midori (2015)
  - MANTIS (2016)
  - QARMA (2017\*)
  - Kcipher (2020)
  - SPEEDY (2021)
  - Orthros (2021)
  - SCARF (2022\*)
  - LLLWBC (2022)
  - Sonic and SuperSonic (2023)
  - BipBip (2023)
  - loVCipher (2024)
- Physical attack resistant variants

# Ubiquitous Computing Era



# Ubiquitous Computing Era

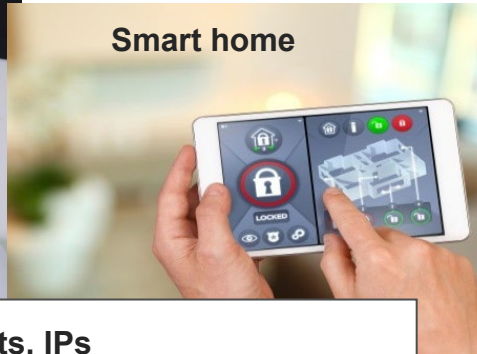
Electronic passports



Smartphones  
Mobile applications



Smart home



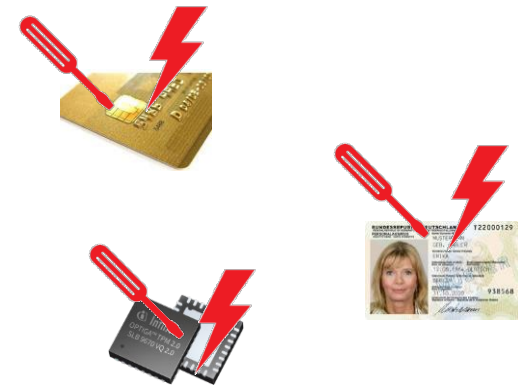
Medical & sensor systems



Payment & toll-collection cards



Products, IPs



Automation components  
Asset tracking systems



Car key systems

**Security Concerns!!!**



## ePayments & Digital Currencies

Home About Calendar Archive Links Advertise Blog

September/October 2011

### Another one bites the dust (Mifare DESFire)!



I'm not sure it's a big surprise but this month David Oswald and Christof Paar from the Horst Gortz Institute for IT Security in Bochum Germany have given a paper at the CHES (Cryptographic Hardware and Embedded Systems) conference in Japan on how to break the Mifare DESFire MF3ICD40 contactless memory chip.

So back to basics, do we need to panic, is it important and will there be further repercussions? Really it's no to all these questions but don't go away yet because that would belittle the quality of their work.

They used Side Channel Analysis (SCA) by using an electromagnetic probe to contactlessly measure the power signal taken by the chip. Using these techniques they were able to recover the 2 DES keys (56 bits each) from

ANDY GREENBERG SECURITY 09.10.2013 01:00 PM

### Hackers Can Steal a Tesla Model S in Seconds by Cloning Its Key Fob

Weak encryption in Tesla Model S key fobs allowed all-too-easy theft, but you can set a PIN code on your Tesla to protect it.



The researchers also found their attack might work against cars sold by McLaren and Karma, and motorcycles sold by Triumph, which use keyless entry systems made by the same manufacturer. STEVE MULLER/GETTY IMAGES

### A New Wireless Hack Can Unlock 100 Million Volkswagens

IDEAS SCIENCE SECURITY

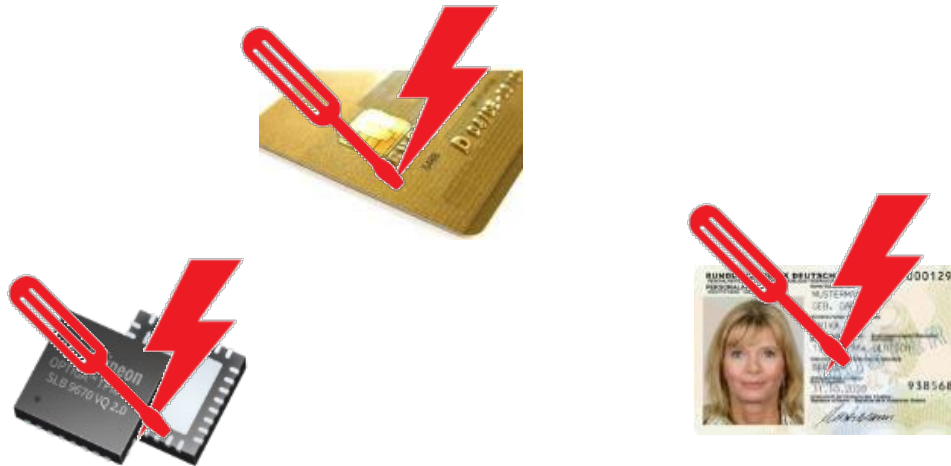
ANDY GREENBERG SECURITY 08.10.13 4:29 PM

### A New Wireless Hack Can Unlock 100 Million Volkswagens



KAZUHIRO NAGAI/AFP/GETTY IMAGES

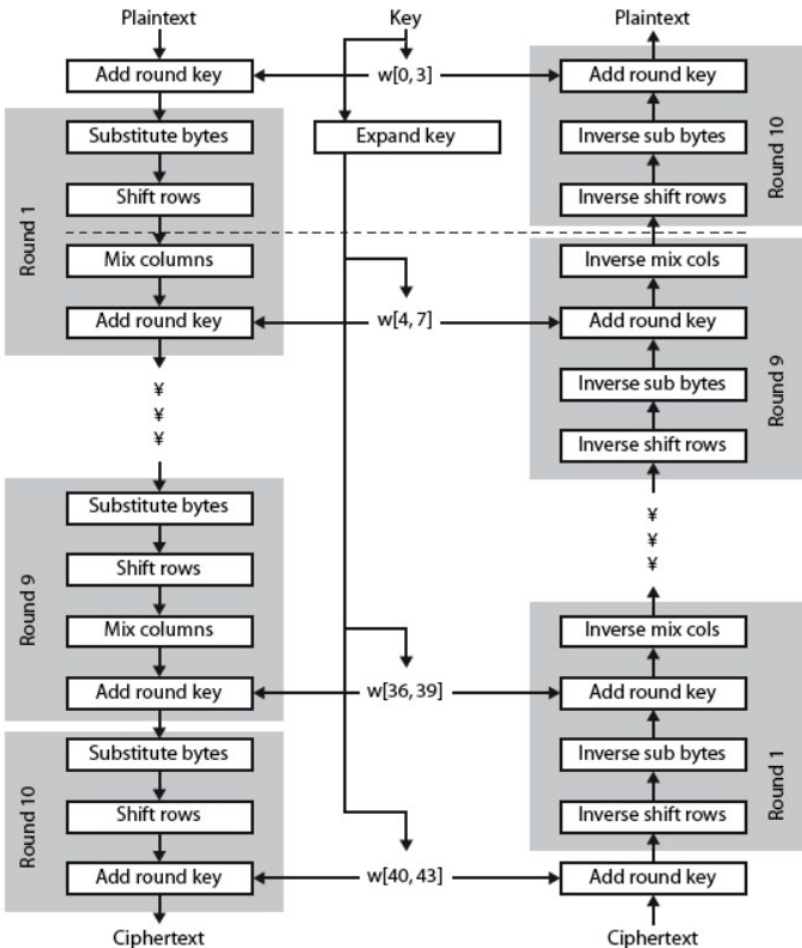
In 2013, when University of Birmingham computer scientist Flavio Garcia and a team of researchers were preparing to reveal a vulnerability that allowed them to start the ignition of millions of Volkswagen cars and drive them off without a key, they were hit with a lawsuit that delayed the publication of their research for two years. But that experience doesn't seem to have deterred Garcia and his



**Good security designs and  
architecture needed to  
resist attacks!**

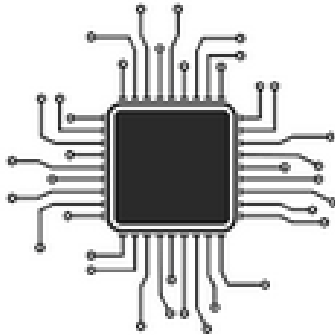
- Access control
- Data confidentiality
- Security
- Counterfeiting mitigations

# Need for Security: Same Level?



- **Conventional cryptography**
  - RSA
  - Standard block cipher solutions (AES, etc.)
- **Applications in**
  - Servers
  - PCs
  - “Strong” tablets, smartphones

## Embedded/IoT devices → Resource-constrained!



**Chip Area: Limited!**

**Power and Energy Consumption: Lowest Possible!**



→ to match these constraints:

**Need for Tailored Cryptography: Lightweight Cryptography**



- Reduces computational efforts to provide security
  - Cheaper than traditional crypto
  - Not weak, but “sufficient,, security
- Many different proposals/implementations especially in the last decade
  - Public-key cryptography: ECC
  - Stream ciphers: Grain, Trivium, etc.
  - Hash functions: Photon, Quark, etc.
  - **Block ciphers**
    - Core for symmetric cryptography, stream ciphers, MACs, etc.

<https://internetofbusiness.com/nist-demands-lightweight-cryptography-to-protect-iot-devices/>



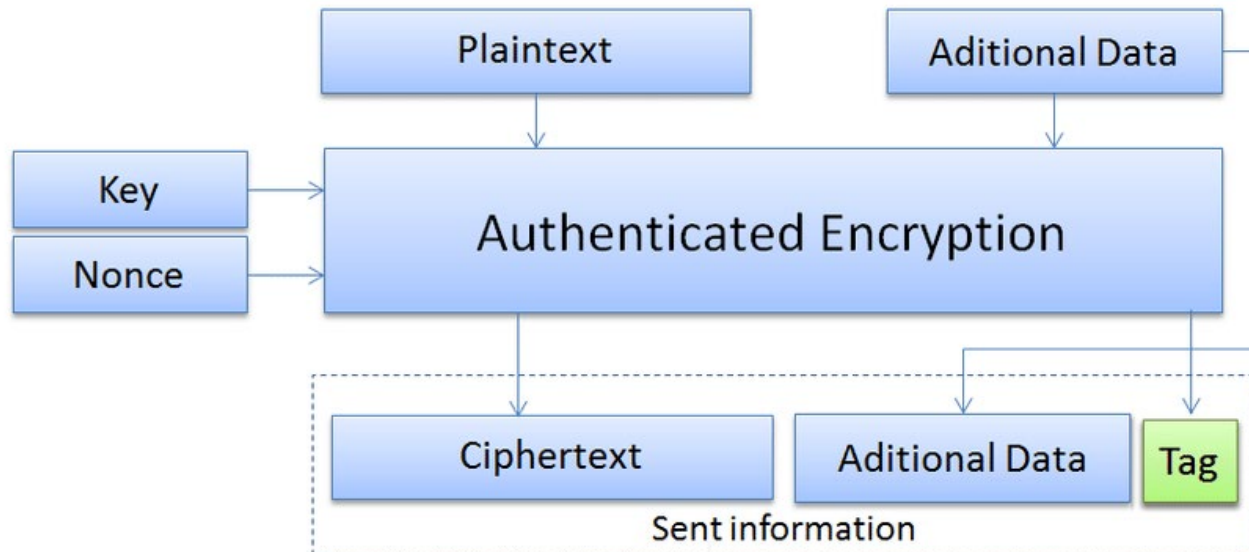
Home > News > NIST demands lightweight cryptography to protect IoT devices



NEWS

## NIST demands lightweight cryptography to protect IoT devices

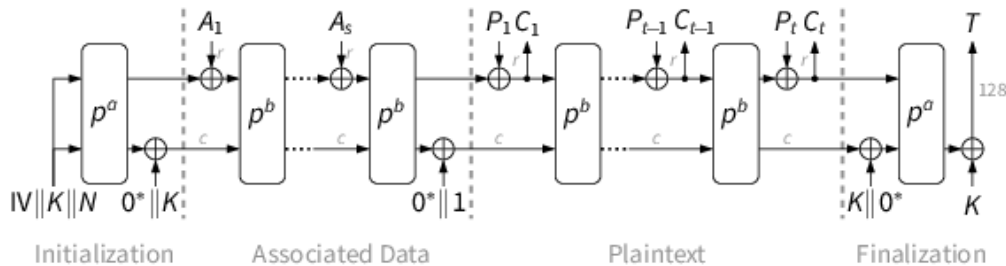
- Lightweight Cryptography (LWC) Standardization Competition by NIST
  - Specifically:  
“Authenticated Encryption with Associated Data” (AEAD)



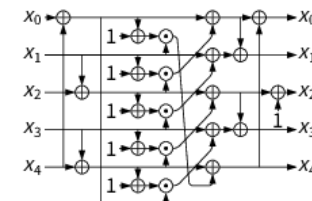
\*Figure from “L. Cardoso Santos and J. López, Pipeline Oriented Implementation of NORX for ARM Processors, SBSEG’17”

- In round format
- ASCON selected out of 10 finalists

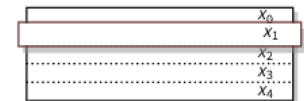
1. **Initialization:** initializes the state with the key  $K$  and nonce  $N$ .
2. **Associated Data Processing:** updates the state with associated data blocks  $A_j$ .
3. **Plaintext Processing:** injects plaintext blocks  $P_j$  into the state and extracts ciphertext blocks  $C_j$ .
4. **Finalization:** injects the key  $K$  again and extracts the tag  $T$  for authentication.



The duplex sponge mode for ASCON authenticated encryption [tex]



ASCON's S-box [tex] [C instructions]



$$\begin{aligned}
 x_0 &:= x_0 \oplus (x_0 \ggg 19) \oplus (x_0 \ggg 28) \\
 x_1 &:= x_1 \oplus (x_1 \ggg 61) \oplus (x_1 \ggg 39) \\
 x_2 &:= x_2 \oplus (x_2 \ggg 1) \oplus (x_2 \ggg 6) \\
 x_3 &:= x_3 \oplus (x_3 \ggg 10) \oplus (x_3 \ggg 17) \\
 x_4 &:= x_4 \oplus (x_4 \ggg 7) \oplus (x_4 \ggg 41)
 \end{aligned}$$

ASCON's linear layer

ASCON's permutation:  $\oplus$  denotes XOR,  $\odot$  denotes AND,  $\ggg$  is rotation to the right.

ISO/IEC 29192-2:2019

Information security — Lightweight cryptography — Part 2: Block ciphers

## PRESENT: An Ultra-Lightweight Block Cipher

A. Bogdanov<sup>1</sup>, L.R. Knudsen<sup>2</sup>, G. Leander<sup>1</sup>, C. Paar<sup>1</sup>, A. Poschmann<sup>1</sup>,  
M.J.B. Robshaw<sup>3</sup>, Y. Seurin<sup>3</sup>, and C. Vikkelsøe<sup>2</sup>

<sup>1</sup> Horst-Görtz-Institute for IT-Security, Ruhr-University Bochum, Germany

<sup>2</sup> Technical University Denmark, DK-2800 Kgs. Lyngby, Denmark

<sup>3</sup> France Telecom R&D, Issy les Moulineaux, France

leander@rub.de, {abogdanov,cpaar,poschmann}@crypto.rub.de

lars@ramkilde.com, chv@mat.dtu.dk

{matt.robshaw,yannick.seurin}@orange-ftgroup.com

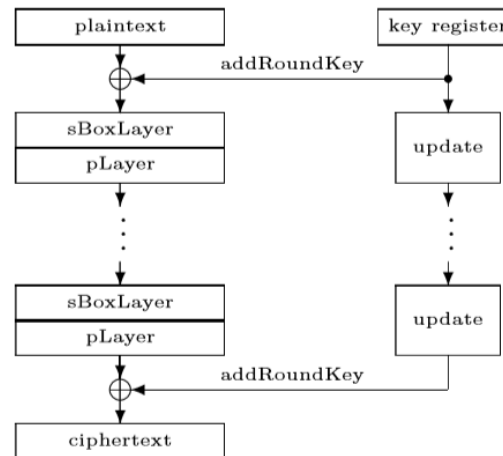
**Abstract.** With the establishment of the AES the need for new block ciphers has been greatly diminished; for almost all block cipher applications the AES is an excellent and preferred choice. However, despite recent implementation advances, the AES is not suitable for extremely constrained environments such as RFID tags and sensor networks. In this paper we describe an ultra-lightweight block cipher, PRESENT. Both security and hardware efficiency have been equally important during the design of the cipher and at 1570 GE, the hardware requirements for PRESENT are competitive with today's leading compact stream ciphers.

- Simple but strong design
  - Well-studied substitution-permutation network (SPN)
- Targeting hardware
- Low-area
  - Permutation is just wiring in hardware!

ISO/IEC 29192-2:2019

Information security — Lightweight cryptography — Part 2: Block ciphers

```
generateRoundKeys()  
for  $i = 1$  to 31 do  
  addRoundKey( $STATE, K_i$ )  
  sBoxLayer( $STATE$ )  
  pLayer( $STATE$ )  
end for  
addRoundKey( $STATE, K_{32}$ )
```



- Simple but strong design
  - Well-studied substitution-permutation network (SPN)
- Targeting hardware
- Low-area
  - Permutation is just wiring in hardware!

## KLEIN: A New Family of Lightweight Block Ciphers

Zheng Gong<sup>1</sup>, Svetla Nikova<sup>2,3</sup> and Yee Wei Law<sup>4</sup>

<sup>1</sup> School of Computer Science, South China Normal University, China  
cis.gong@gmail.com

<sup>2</sup> Faculty of EWI, University of Twente, The Netherlands

<sup>3</sup> Dept. ESAT/SCD-COSIC, Katholieke Universiteit Leuven, Belgium  
s.i.nikova@utwente.nl

<sup>4</sup> Department of EEE, The University of Melbourne, Australia  
yee.wei.law@gmail.com

**Abstract.** Resource-efficient cryptographic primitives are essential for realizing both security and efficiency in embedded systems like RFID tags and sensor nodes. Among those primitives, lightweight block cipher plays a major role as a building block for security protocols. In this paper, we describe a new family of lightweight block ciphers named KLEIN, which is designed for resource-constrained devices such as wireless sensors and RFID tags. Compared to related proposals, KLEIN has advantage in the software performance on legacy sensor platforms, while its hardware implementation can be compact as well.

- AES-like
- Works on nibbles
- Involution Sbox

## The LED Block Cipher\*

Jian Guo<sup>1</sup>, Thomas Peyrin<sup>2,†</sup>, Axel Poschmann<sup>2,†</sup>, and Matt Robshaw<sup>3,‡</sup>

<sup>1</sup> Institute for Infocomm Research, Singapore

<sup>2</sup> Nanyang Technological University, Singapore

<sup>3</sup> Applied Cryptography Group, Orange Labs, France

{ntu.guo,thomas.peyrin}@gmail.com

aposchmann@ntu.edu.sg

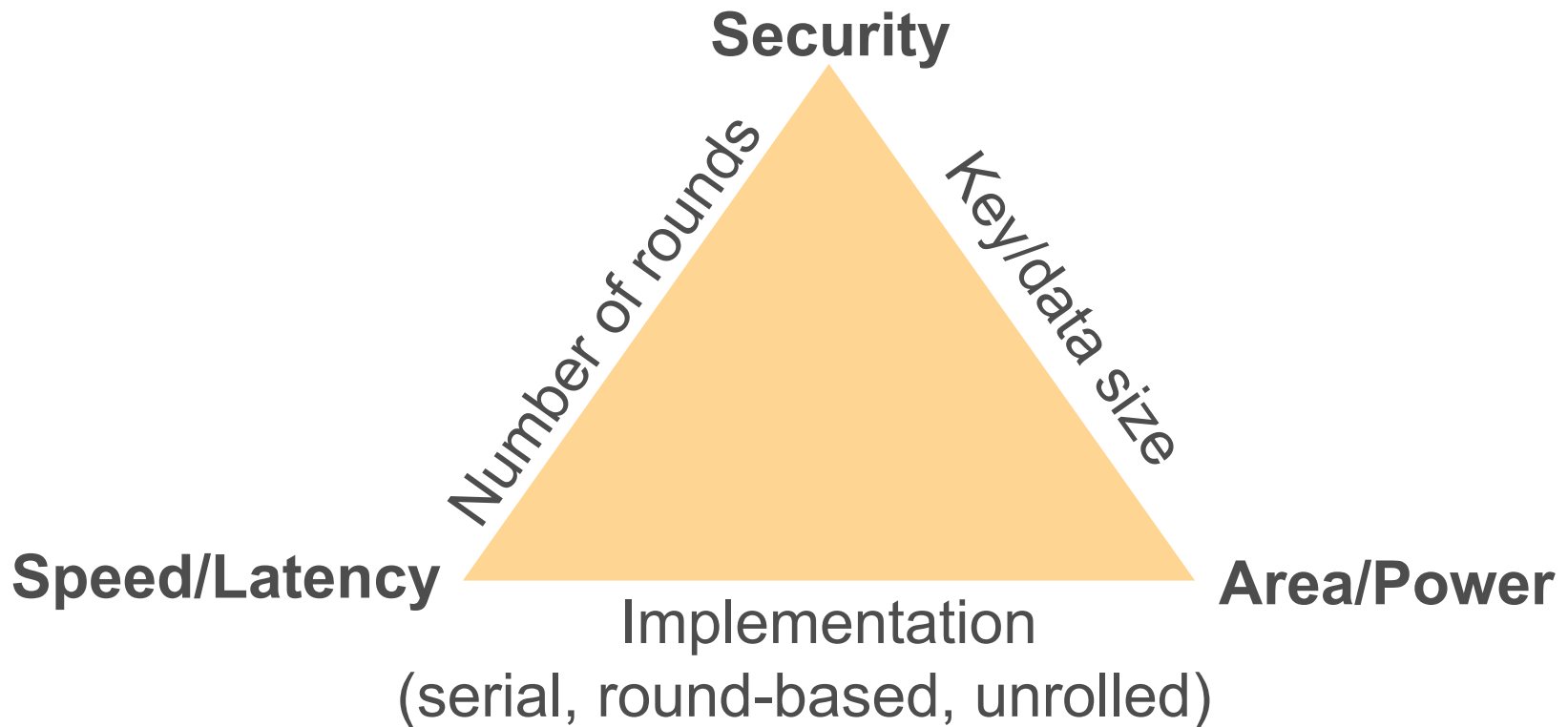
matt.robshaw@orange.com

**Abstract.** We present a new block cipher LED. While dedicated to compact hardware implementation, and offering the smallest silicon footprint among comparable block ciphers, the cipher has been designed to simultaneously tackle three additional goals. First, we explore the role of an ultra-light (in fact non-existent) key schedule. Second, we consider the resistance of ciphers, and LED in particular, to related-key attacks: we are able to derive simple yet interesting AES-like security proofs for LED regarding related- or single-key attacks. And third, while we provide a block cipher that is very compact in hardware, we aim to maintain a reasonable performance profile for software implementation.

- AES-like
- Uses PRESENT Sbox
- Consists of steps
  - Number based on key size
  - Each step 4 rounds



- Initial proposals mostly address area in hardware / speed in software
- Other important metrics?



- **Area**
  - Usually measured in  $\mu\text{m}^2$ , but depends on technology and the standard cell library
  - Hence stated in gate equivalents (GE) independent of the technology and library
  - One GE is equivalent to the area required to implement the two-input NAND gate (area derived by dividing the area in  $\mu\text{m}^2$  by the area of a two-input NAND gate)
- **Cycles**
  - # of clock cycles required to compute and output the results
- **Time**
  - Required time for a certain operation, i. e., # of cycles divided by operating frequency

\* Shahram Rasoolzadeh, Hardware-Oriented SPN Block Ciphers, PhD Thesis, RUB, 2020

- **Throughput**
  - Bit rate production of a new output w.r.t., i. e., # of output bits divided by time (expressed in bits per second – bps –)
- **Power**
  - Usually the power consumption estimated on the gate level by the synthesizer tool (typically in  $\mu\text{W}$ )
  - Power estimations on transistor level are more accurate (more steps in design flow)
- **Energy**
  - Power consumption over a certain time period, i. e., multiplying the power consumption with the required time (typically in  $\mu\text{J}$ )

---

\* Shahram Rasoolzadeh, Hardware-Oriented SPN Block Ciphers, PhD Thesis, RUB, 2020

- **Unrolled**
  - Whole encryption or decryption process is computed within only one clock cycle without using any registers in combinatorial circuit
  - Low-latency
  
- **Pipelined**
  - Circuit for whole encryption or decryption process is implemented (similar to unrolled), some registers are inserted in the critical path (path with maximum delay) to increase
  - Higher throughput rate but with the cost of higher area and power consumption

---

\* Shahram Rasoolzadeh, Hardware-Oriented SPN Block Ciphers, PhD Thesis, RUB, 2020

- **Round-based**
  - Each round function of the cipher is computed within one clock cycle
  - Reduces area and power at cost of decreasing throughput
  
- **Serialized**
  - Each round function computed in several clock cycles, and in each clock cycle, a small part of the round function is computed (e. g., only one S-box, or only one word of the linear layer)
  - Lower area & power consumption, but also lowest throughput
  - After a point, implementing control logic may require more overhead than before

---

\* Shahram Rasoolzadeh, Hardware-Oriented SPN Block Ciphers, PhD Thesis, RUB, 2020

# Proposals vs. Metrics

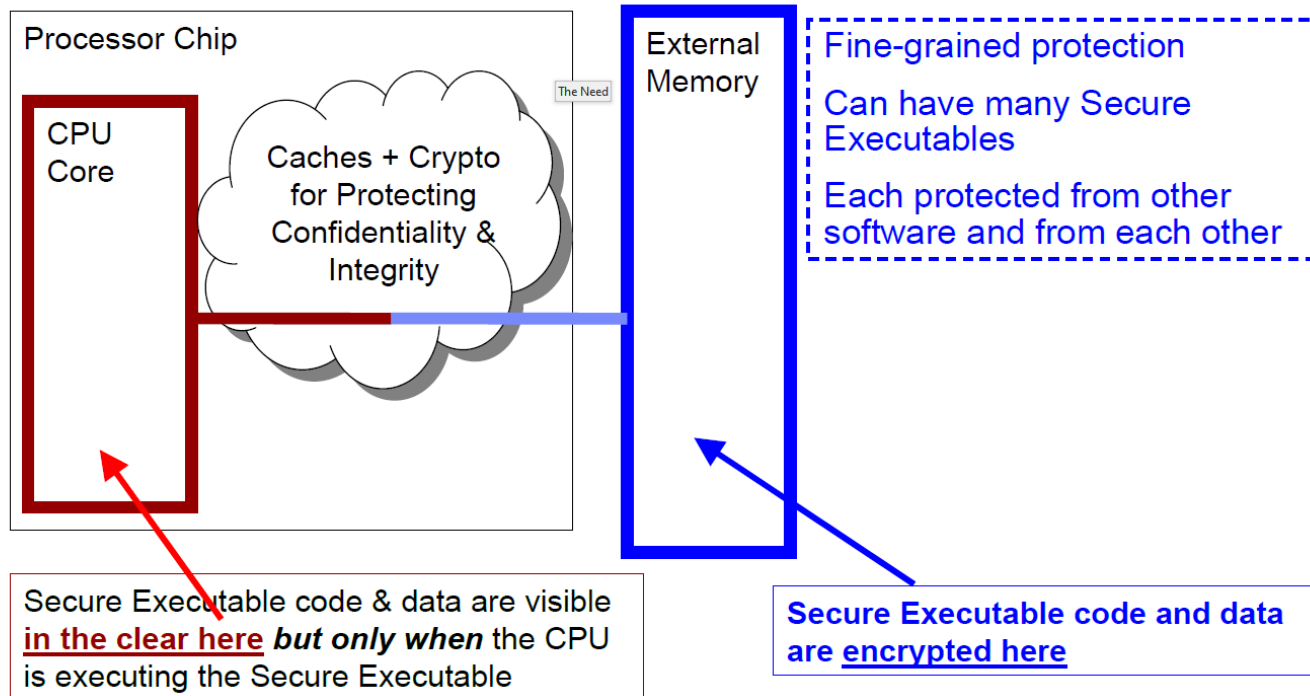
	Hardware	Software
Area/ Code size	mCrypton LBlock HIGHT KLEIN KATAN CLEFIA TWINE Piccolo PRESENT KTANTAN LED SIMON	ITUBee KLEIN SEA SPECK
Latency/ Execution time	?	LBlock TWINE KLEIN SPECK

## IBM

**NSA** | TRUSTED  
COMPUTING  
Conference & Exposition

### New Twist on **SecureBlue: SecureBlue++**

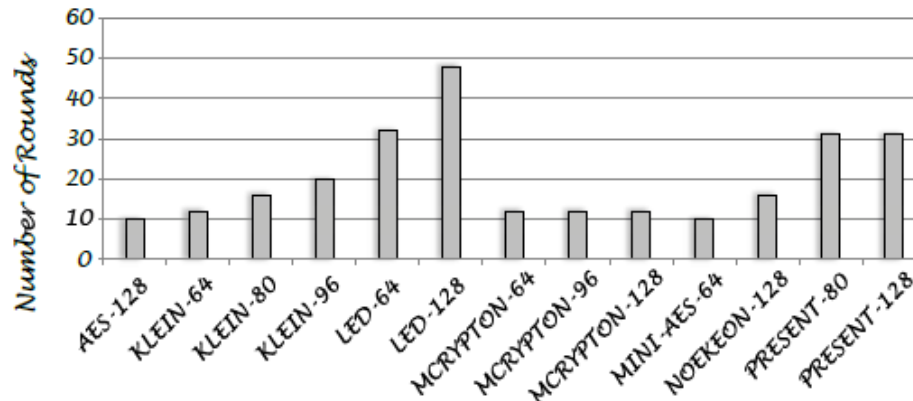
- Like SecureBlue, but provides more fine-grained SecureBlue-like crypto protection to protect information in one program from other S/W (including OS)
  - Protect confidentiality & integrity of information so other S/W cannot read it or undetectably tamper with it



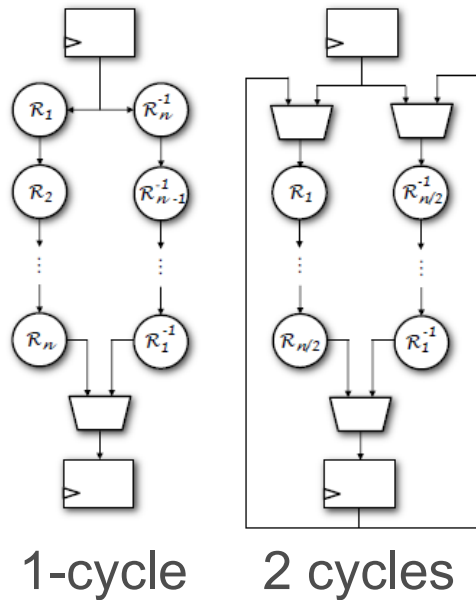
10

Also,

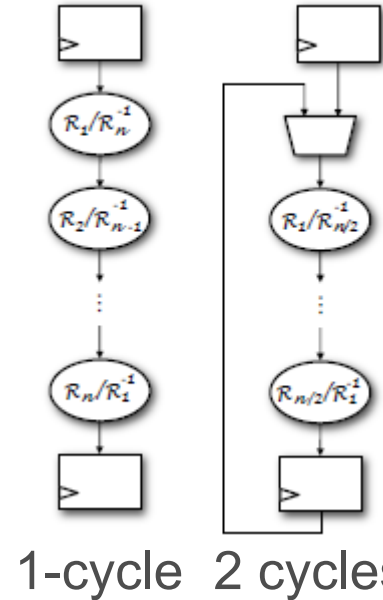
- Intel SGX
- AMD SEV



## Unrolled



Enc/Dec

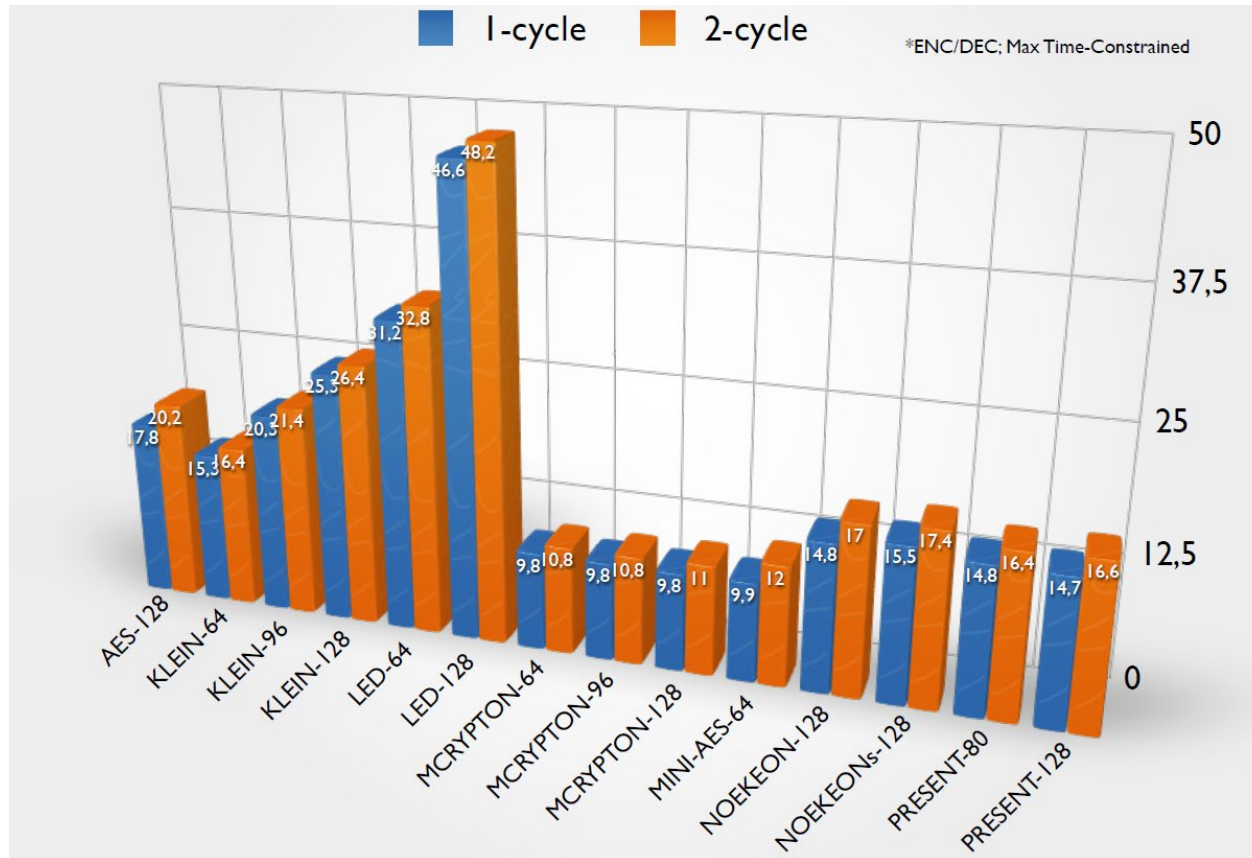


Enc/Dec  
Shared Datapath

\* Knezevic et al., Low-Latency Encryption – Is “Lightweight = Light + Wait”?, CHES, 2012

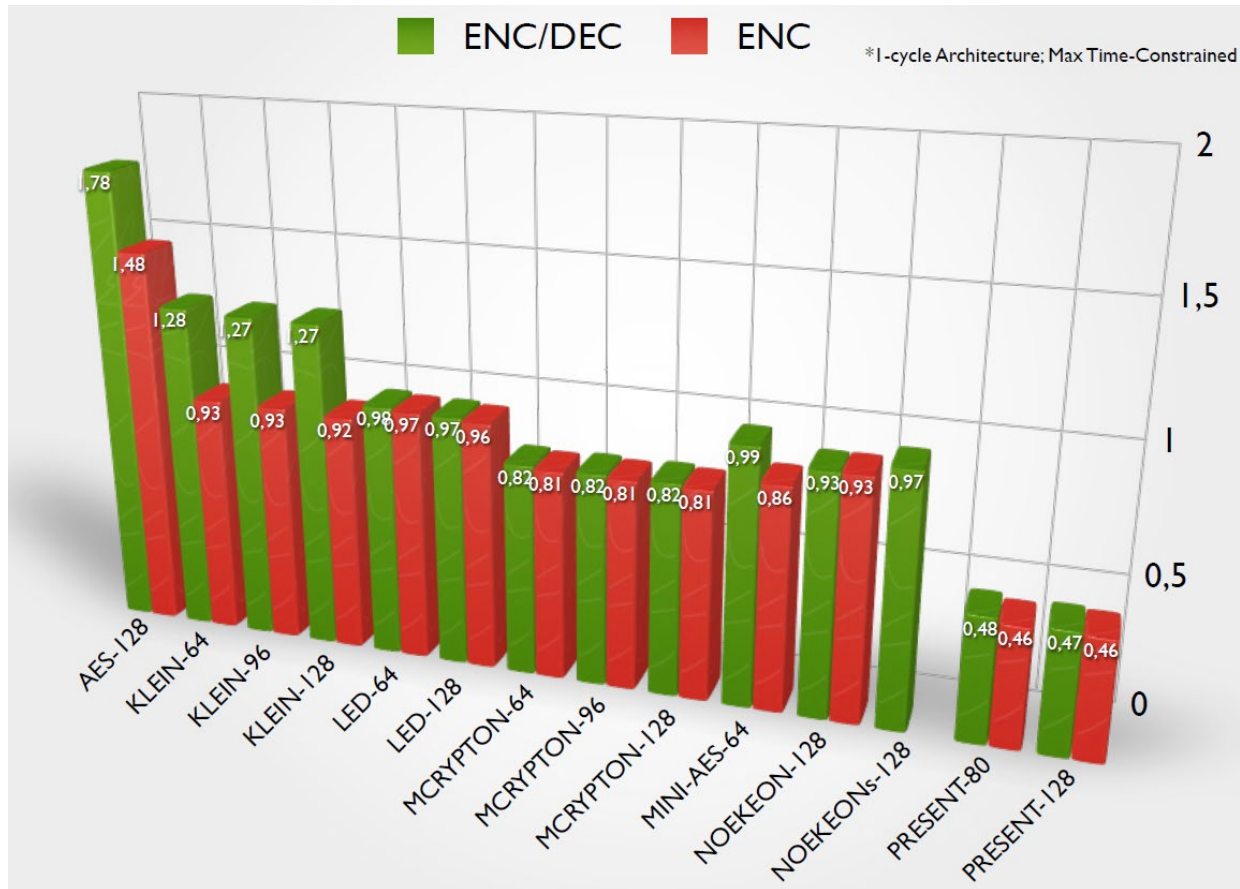


## Latency: Time to encrypt one block of data (ns)



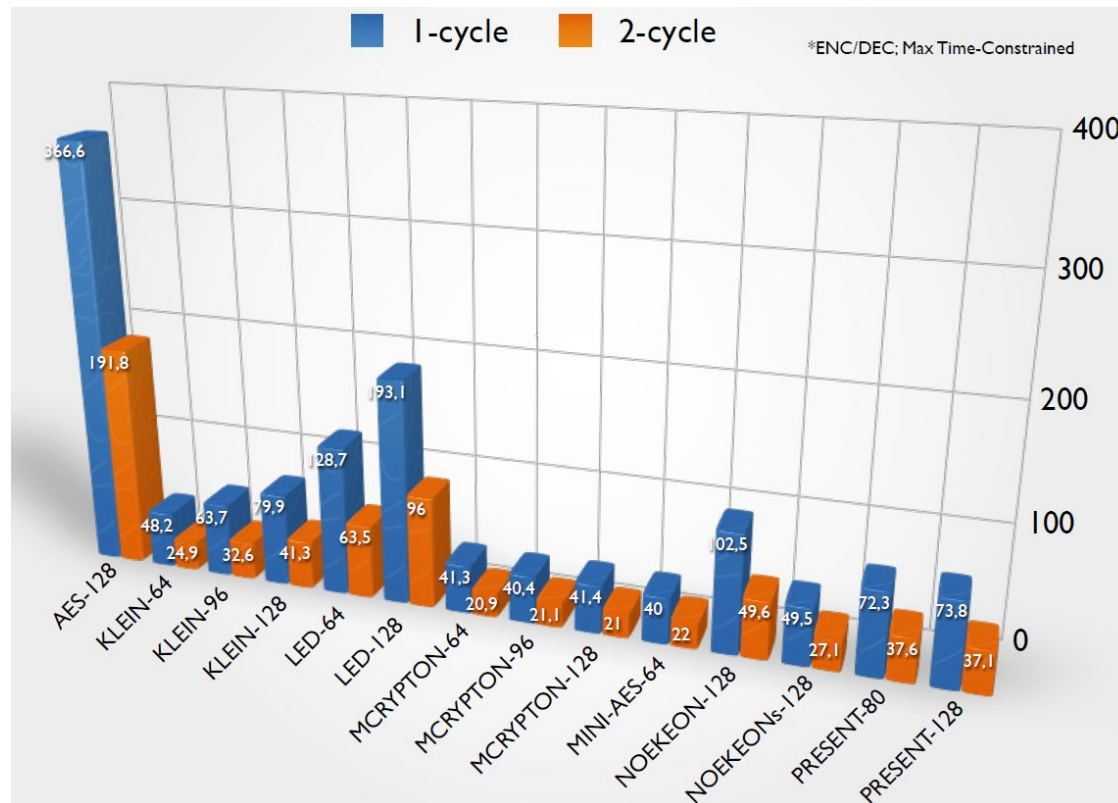
\* Knezevic et al., Low-Latency Encryption – Is “Lightweight = Light + Wait”?, CHES, 2012

Latency: Time to encrypt one block of data (ns) → per round



\* Knezevic et al., Low-Latency Encryption – Is “Lightweight = Light + Wait”?, CHES, 2012

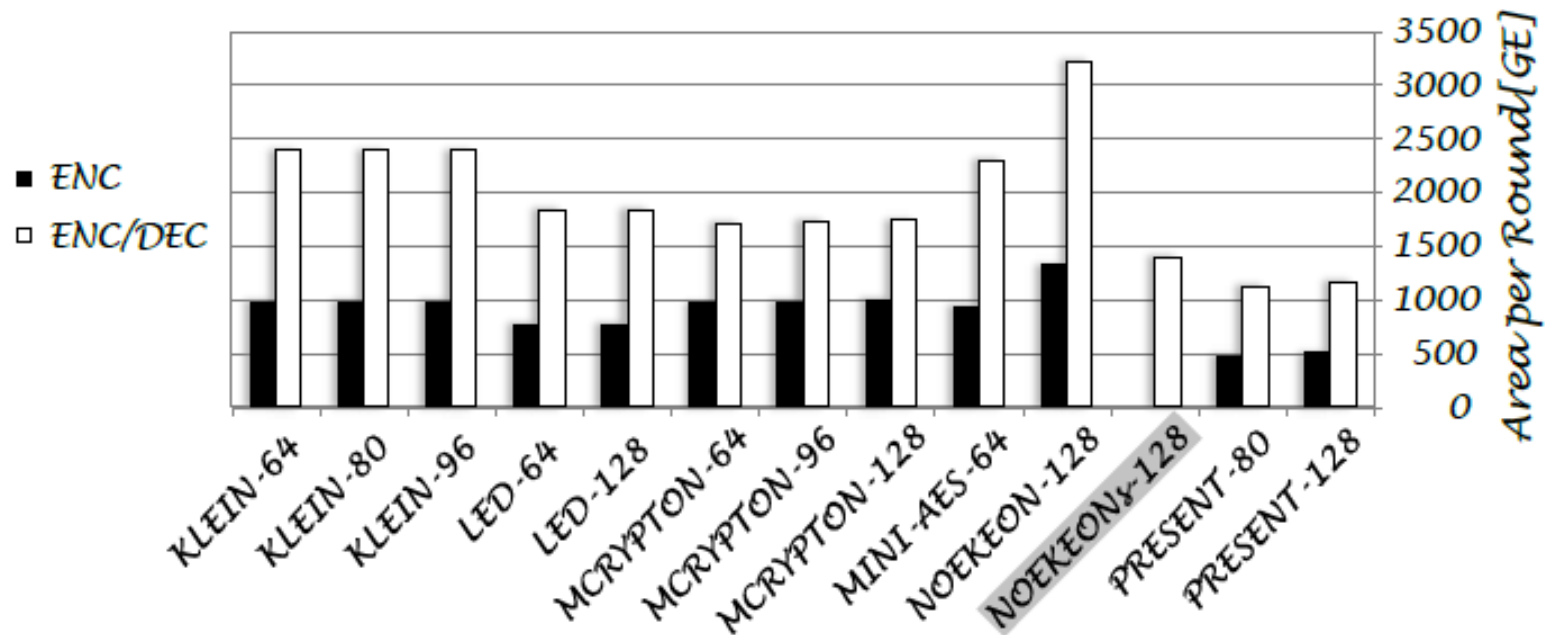
## Latency: Time to encrypt one block of data (Corresponding area cost in GE)



**Less area possible for encryption of one block of data?**

\* Knezevic et al., Low-Latency Encryption – Is “Lightweight = Light + Wait”?, CHES, 2012

Latency: Time to encrypt one block of data  
(Corresponding area cost in GE  $\rightarrow$  per round)



**Less area possible for encryption of one block of data?**

\* Knezevic et al., Low-Latency Encryption – Is “Lightweight = Light + Wait”?, CHES, 2012

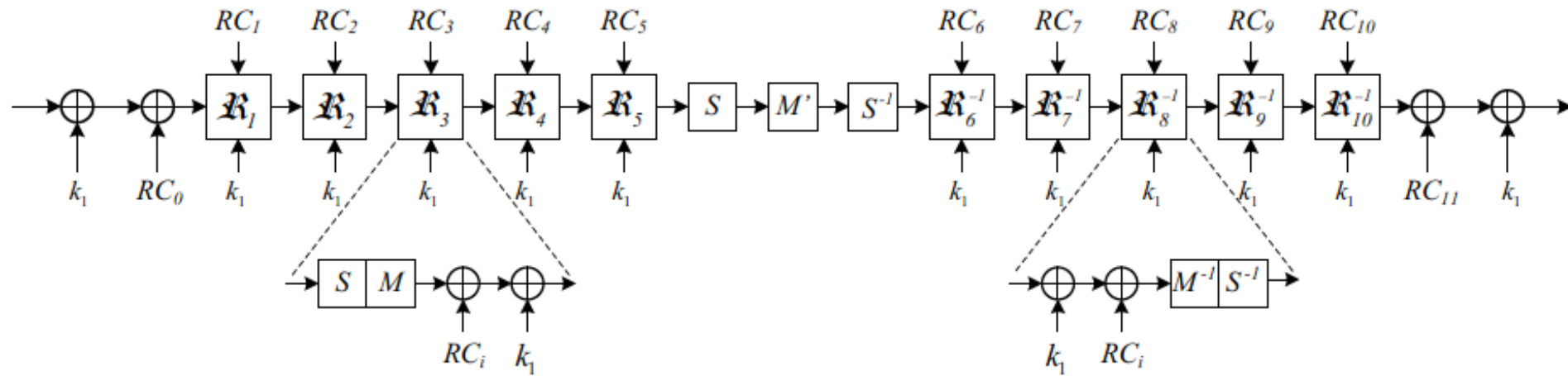
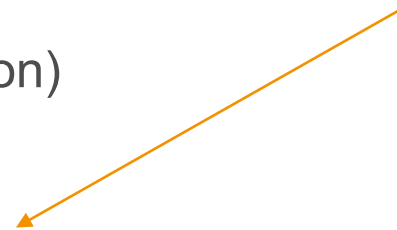
- Keep the hardware cost of one round as low as possible
  - Main savings in Sbox, smaller (3/4 bit Sboxes better)
  - Even among these there are significant differences
- All rounds are unrolled
  - Cipher can be thought as one big round
  - Number of rounds hence is important, should be minimized
- All rounds same, decreases cost
  - Less round complexity as well based on components, not too low

- Slightly heavy round with less/balanced number of rounds
- Simpler key schedule
  - Should be independent of number of rounds
  - Constant addition instead of key schedule should be preferred, if possible
- Minimum overhead for encryption and decryption
  - Use involution

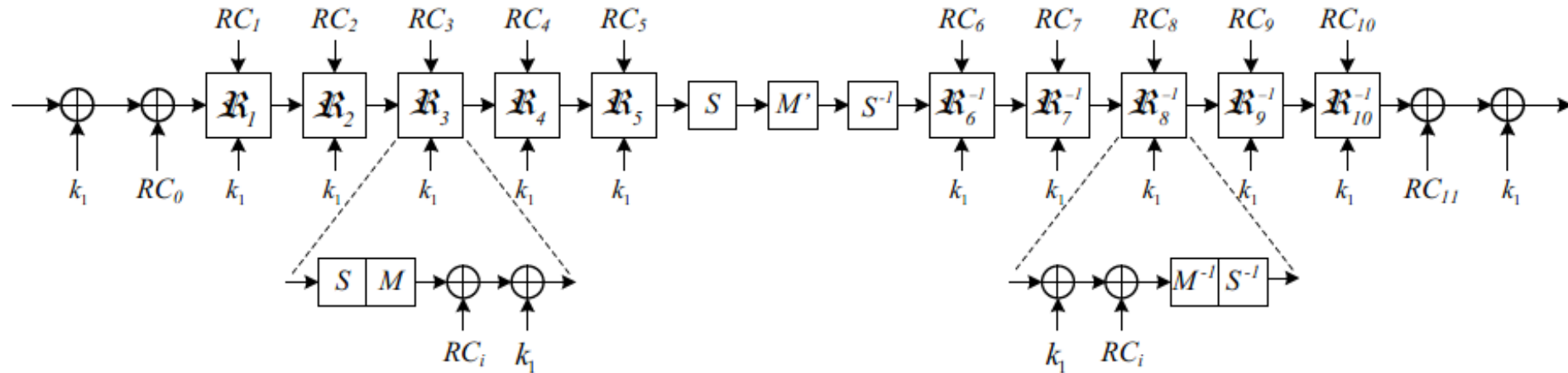
# Proposals vs. Metrics

	Hardware	Software
Area/ Code size	mCrypton KLEIN TWINE PRESENT LBlock KATAN KTANTAN Piccolo LED SIMON HIGHT CLEFIA	ITUBee SEA KLEIN SPECK
Latency/ Execution time	<b>PRINCE</b> ?	LBlock KLEIN TWINE SPECK

- 64-bit block, 128-bit key
- Core cipher with 64-bit key
- 64-bit whitening keys (FX construction)
- 12 rounds







$$R = SR \circ MC \circ SB, \quad R'_{\text{PRINCE}} = SB^{-1} \circ MC \circ SB \quad \text{and} \quad R^{-1} = SB^{-1} \circ MC \circ SR^{-1}$$

$$\oplus_{k_i}(x) := x + k_i$$

$$\oplus_{RC_i}(x) = x + RC_i$$

$$R = SR \circ MC \circ SB, \quad R'_{\text{PRINCE}} = SB^{-1} \circ MC \circ SB \quad \text{and} \quad R^{-1} = SB^{-1} \circ MC \circ SR^{-1}$$

$$\oplus_{k_i}(x) := x + k_i$$

$$\oplus_{RC_i}(x) = x + RC_i$$

## SB Sbox

$x$	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S[x]$	B	F	3	2	A	C	9	1	6	7	8	0	E	5	D	4

$$\widehat{M}^{(0)} = \begin{pmatrix} M_1 & M_2 & M_3 & M_4 \\ M_2 & M_3 & M_4 & M_1 \\ M_3 & M_4 & M_1 & M_2 \\ M_4 & M_1 & M_2 & M_3 \end{pmatrix}, \quad \widehat{M}^{(1)} = \begin{pmatrix} M_2 & M_3 & M_4 & M_1 \\ M_3 & M_4 & M_1 & M_2 \\ M_4 & M_1 & M_2 & M_3 \\ M_1 & M_2 & M_3 & M_4 \end{pmatrix}, \quad M' = \begin{pmatrix} \widehat{M}^{(0)} & 0 & 0 & 0 \\ 0 & \widehat{M}^{(1)} & 0 & 0 \\ 0 & 0 & \widehat{M}^{(1)} & 0 \\ 0 & 0 & 0 & \widehat{M}^{(0)} \end{pmatrix}$$

$M_i$  is the  $4 \times 4$  identity matrix

MC-layer multiplies the state with  $M'$

involution

### Permutation of SR

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	5	10	15	4	9	14	3	8	13	2	7	12	1	6	11

# Low-latency: PRINCE Core Steps

$$R = SR \circ MC \circ SB, \quad R'_{\text{PRINCE}} = SB^{-1} \circ MC \circ SB \quad \text{and} \quad R^{-1} = SB^{-1} \circ MC \circ SR^{-1}$$

$$\oplus_{k_i}(x) := x + k_i$$

$$\oplus_{RC_i}(x) = x + RC_i$$

$RC_0$	0000000000000000
$RC_1$	13198a2e03707344
$RC_2$	a4093822299f31d0
$RC_3$	082efa98ec4e6c89
$RC_4$	452821e638d01377
$RC_5$	be5466cf34e90c6c
$RC_6$	7ef84f78fd955cb1
$RC_7$	85840851f1ac43aa
$RC_8$	c882d32f25323c54
$RC_9$	64a51195e0e3610d
$RC_{10}$	d3b5a399ca0c2399
$RC_{11}$	c0ac29b7c97c50dd

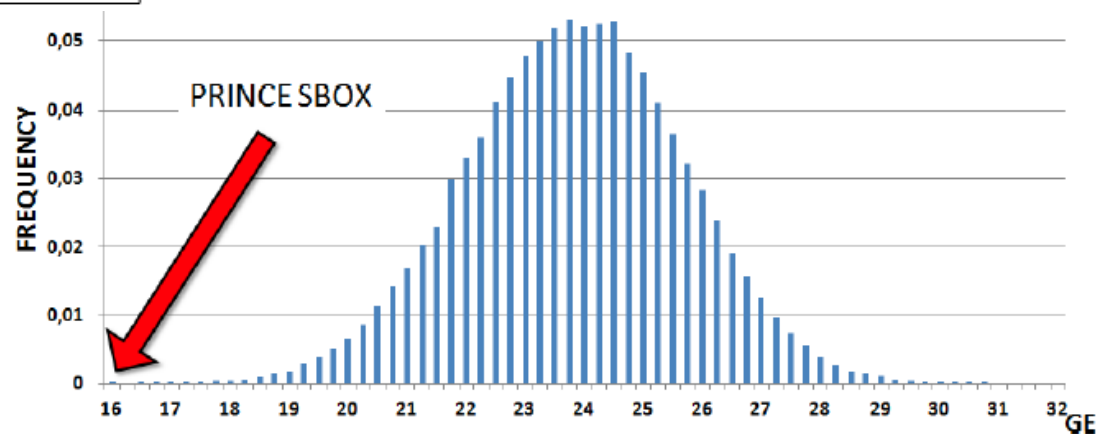
→  $\alpha = c0ac29b7c97c50dd$ ,

$$(k_0 || k_1) \mapsto (k_0 || P(k_0) || k_1)$$

$$P(x) = (x \ggg 1) \oplus (x \ggg 63)$$

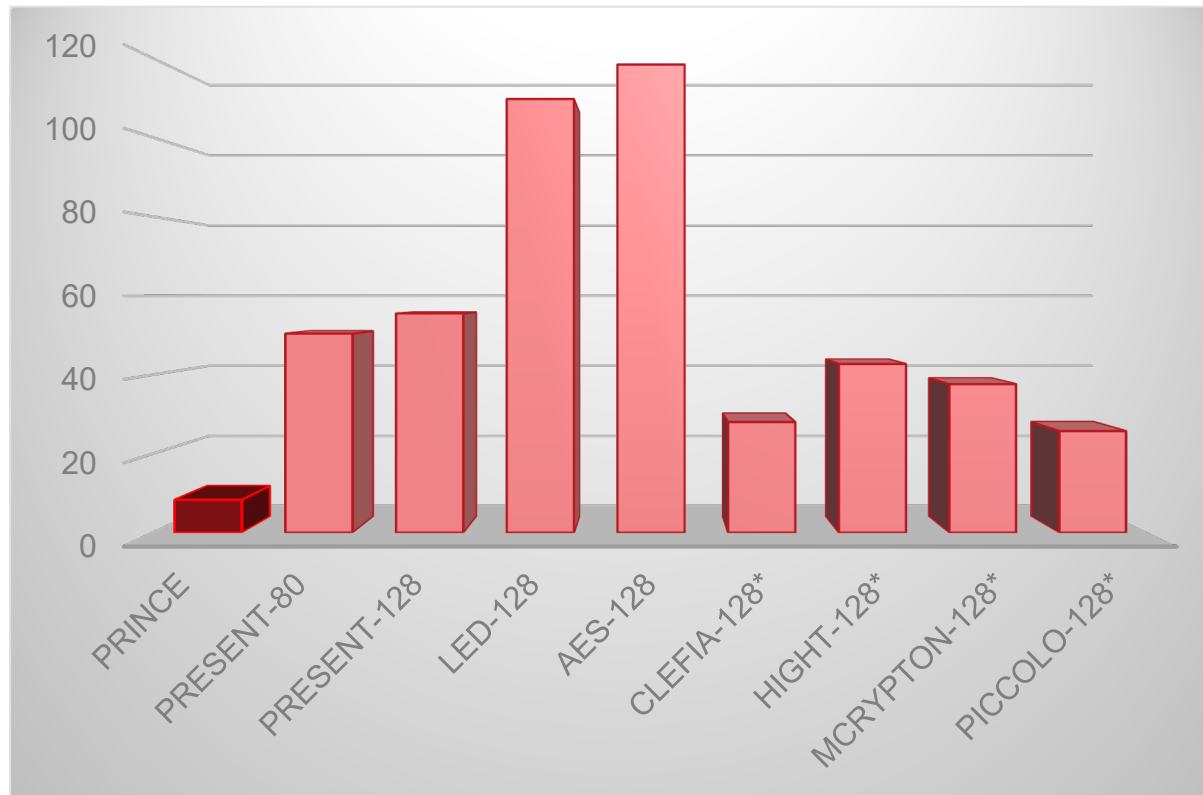
- 64000 Sboxes (and their inverses) with good cryptographic criteria are implemented and synthesized to obtain average gate counts
- Smallest Sbox selected

$S_0$	0x0, 0x1, 0x2, 0xD, 0x4, 0x7, 0xF, 0x6, 0x8, 0xC, 0x5, 0x3, 0xA, 0xE, 0xB, 0x9
$S_1$	0x0, 0x1, 0x2, 0xD, 0x4, 0x7, 0xF, 0x6, 0x8, 0xC, 0x9, 0xB, 0xA, 0xE, 0x5, 0x3
$S_2$	0x0, 0x1, 0x2, 0xD, 0x4, 0x7, 0xF, 0x6, 0x8, 0xC, 0xB, 0x9, 0xA, 0xE, 0x3, 0x5
$S_3$	0x0, 0x1, 0x2, 0xD, 0x4, 0x7, 0xF, 0x6, 0x8, 0xC, 0xB, 0x9, 0xA, 0xE, 0x5, 0x3
$S_4$	0x0, 0x1, 0x2, 0xD, 0x4, 0x7, 0xF, 0x6, 0x8, 0xC, 0xE, 0xB, 0xA, 0x9, 0x3, 0x5
$S_5$	0x0, 0x1, 0x2, 0xD, 0x4, 0x7, 0xF, 0x6, 0x8, 0xE, 0xB, 0xA, 0x5, 0x9, 0xC, 0x3
$S_6$	0x0, 0x1, 0x2, 0xD, 0x4, 0x7, 0xF, 0x6, 0x8, 0xE, 0xB, 0xA, 0x9, 0x3, 0xC, 0x5
$S_7$	0x0, 0x1, 0x2, 0xD, 0x4, 0x7, 0xF, 0x6, 0x8, 0xE, 0xC, 0x9, 0x5, 0xB, 0xA, 0x3



Area distribution of *good* Sboxes (90 nm)

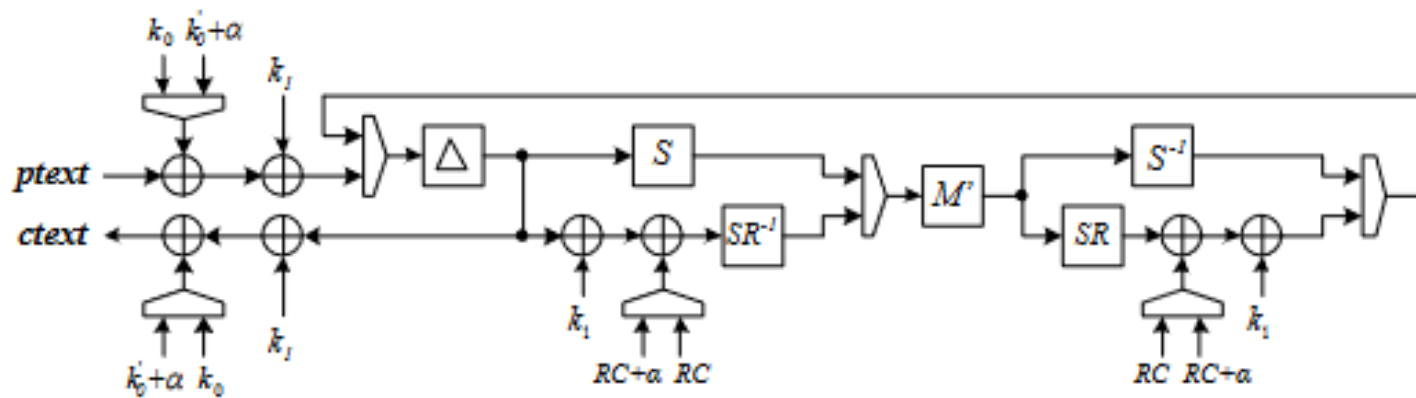
## Results



## Results

	Tech.	Nangate 45nm Generic			UMC 90nm Faraday			UMC 130nm Faraday		
	Constr.(UD)	1000	3162	10000	1000	3162	10000	1000	3162	10000
PRINCE <sup>-</sup>	Area(GE)	8260	8263	8263	7996	7996	7996	8679	8679	8679
	Power(mW)	38.5	17.9	8.3	26.3	10.9	3.9	29.8	11.8	4.1
PRESENT-80	Area(GE)	63942	51631	50429	113062	49723	49698	119196	51790	51790
	Power(mW)	1304.6	320.9	98.0	1436.9	144.9	45.5	1578.4	134.9	42.7
PRESENT-128	Area(GE)	68908	56668	55467	120271	54576	54525	126351	56732	56722
	Power(mW)	1327.1	330.4	99.1	1491.1	149.9	47.8	1638.7	137.4	43.6
LED-128	Area(GE)	109811	109958	109697	281240	286779	98100	236770	235106	111496
	Power(mW)	2470.7	835.7	252.3	5405.0	1076.3	133.7	5274.8	1133.9	163.6
AES-128	Area (GE)	135051	135093	118440	421997	130835	118522	347860	141060	130764
	Power (mW)	3265.8	1165.7	301.6	8903.2	587.4	186.8	8911.2	876.8	229.1

# Low-latency: PRINCE Cipher (Round-based)

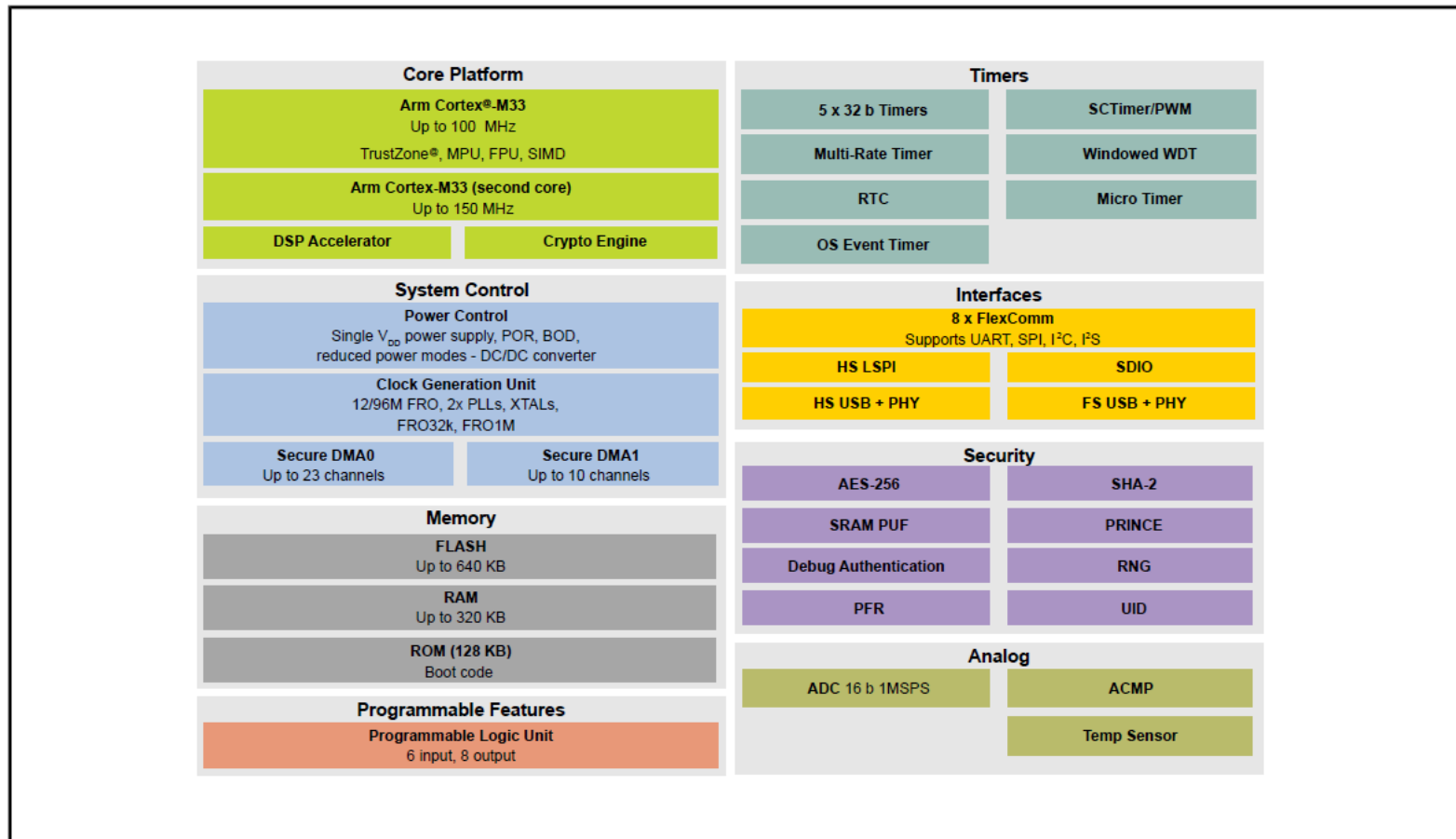


## Results

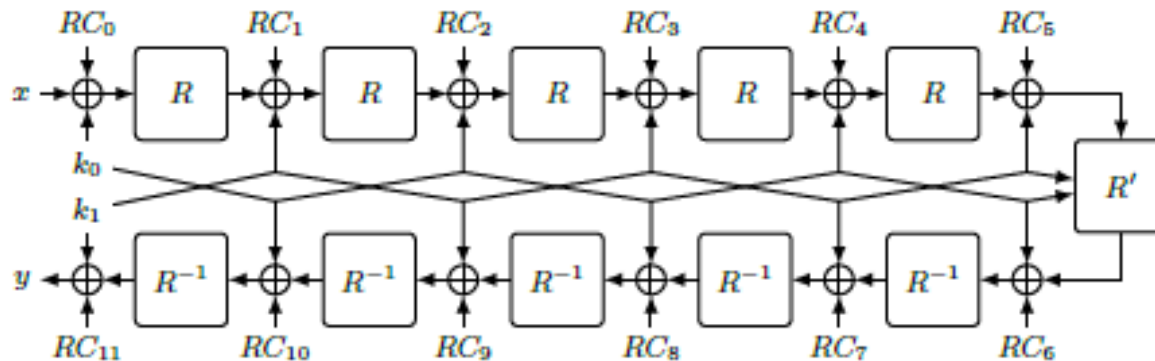
	Nangate 45nm Generic				UMC 90nm Faraday				UMC 130nm Faraday			
	Area (GE)	Freq. (MHz)	Power (mW)	Tput (Gbps)	Area (GE)	Freq. (MHz)	Power (mW)	Tput (Gbps)	Area (GE)	Freq. (MHz)	Power (mW)	Tput (Gbps)
PRINCE	3779	666.7	5.7	3.56	3286	188.7	4.5	1.00	3491	153.8	5.8	0.82
PRESENT-80	3105	833.3	1.2	1.67	2795	222.2	2.1	0.44	2909	196.1	2.5	0.39
PRESENT-128	3707	833.3	1.6	1.67	3301	294.1	3.4	0.59	3458	196.1	2.9	0.39
LED-128	3309	312.5	0.5	0.41	3076	103.1	1.9	0.13	3407	78.13	2.4	0.10
AES-128	15880	250.0	5.8	2.91	14691	78.1	14.3	0.91	16212	61.3	18.8	0.71



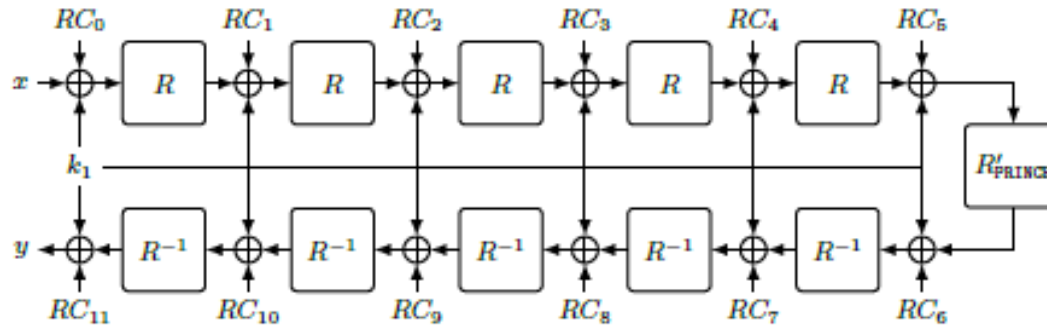
## LPC55S6x MCU Block Diagram



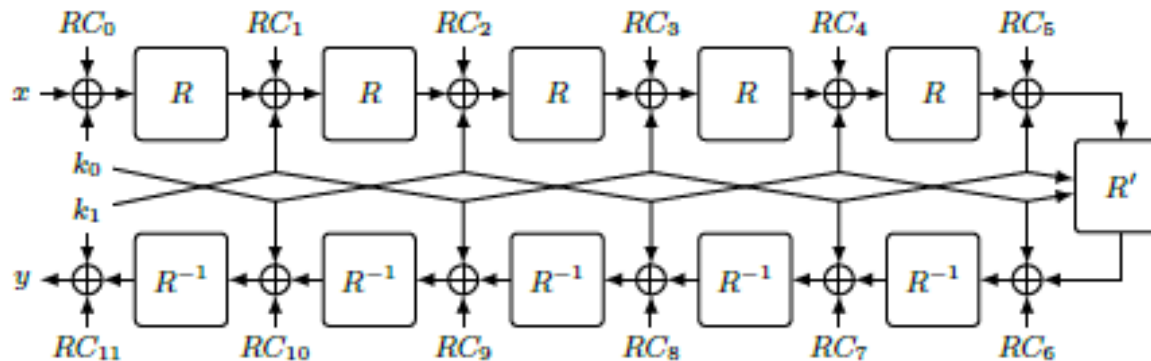
- NIST lightweight security requirement:
  - 112-bit security with at most  $2^{50}$  bytes of chosen data
  - *PRINCE cannot reach*: Data complexity  $2^n$ , time complexity  $2^{126-n}$
- PRINCEv2
  - 64-bit block, 128-bit key
  - Core cipher with 64-bit key
  - 12 rounds



\* Bozilov et al., PRINCEv2: More Security for (Almost) No Overhead, SAC, 2020



- Differences to PRINCE
  - FX construction and alpha reflection removed
  - Key schedule changed, new round constant introduced
  - Keyed middle layer



\* Bozilov et al., PRINCEv2: More Security for (Almost) No Overhead, SAC, 2020

- Differences to PRINCE
  - FX construction and alpha reflection removed
  - Key schedule changed, new round constant introduced
  - Keyed middle layer

$$R' = SB^{-1} \circ \oplus_{RC_{11}+k_1} \circ MC \circ \oplus_{k_0} \circ SB$$

$$\text{Swap}(k_0, k_1, \text{dec}) = \begin{cases} k_0, k_1 & \text{if dec} = 0 \\ k_1 \oplus \beta, k_0 \oplus \alpha & \text{if dec} = 1 \end{cases}$$

Constants	
$RC_0 = 0000000000000000$	$RC_6 = 7ef84f78fd955cb1 = RC_5 \oplus \alpha$
$RC_1 = 13198a2e03707344$	$RC_7 = 7aacf4538d971a60 = RC_4 \oplus \beta$
$RC_2 = a4093822299f31d0$	$RC_8 = c882d32f25323c54 = RC_3 \oplus \alpha$
$RC_3 = 082efa98ec4e6c89$	$RC_9 = 9b8ded979cd838c7 = RC_2 \oplus \beta$
$RC_4 = 452821e638d01377$	$RC_{10} = d3b5a399ca0c2399 = RC_1 \oplus \alpha$
$RC_5 = be5466cf34e90c6c$	$RC_{11} = 3f84d5b5b5470917 = RC_0 \oplus \beta$
$\alpha = c0ac29b7c97c50dd$	$\beta = 3f84d5b5b5470917$

\* Bozilov et al., PRINCEv2: More Security for (Almost) No Overhead, SAC, 2020

- Differences to PRINCE

- PRINCE round keys

$k_0 \oplus k_1, k_1, k_1, k_1, k_1, k_1, k_1 \oplus \alpha, k_1 \oplus \alpha, k_1 \oplus \alpha, k_1 \oplus \alpha, k_1 \oplus \alpha, k_0' \oplus k_1 \oplus \alpha$

- PRINCEv2 round keys

$k_0, k_1, k_0, k_1, k_0, k_1, k_0, k_1 \oplus \beta, k_0 \oplus \alpha, k_1 \oplus \beta, k_0 \oplus \alpha, k_1 \oplus \beta, k_0 \oplus \alpha, k_1 \oplus \beta$

$$\begin{array}{cccccccc}
 k_0 & \rightarrow & k_1 & \rightarrow & k_0 & \rightarrow & k_1 & \rightarrow & k_0 \\
 & & & & & & & & \downarrow \\
 k_1 \oplus \beta & \leftarrow & k_0 \oplus \alpha & \leftarrow & k_1 \oplus \beta & \leftarrow & k_0 \oplus \beta & \leftarrow & k_1 \oplus \beta & \leftarrow & k_0 \oplus \alpha & \leftarrow & k_1 \oplus \beta
 \end{array}$$

$k_0 \leftarrow k_1 \oplus \beta$  and  $k_1 \leftarrow k_0 \oplus \alpha$ :

$$\begin{array}{cccccccc}
 k_1 \oplus \beta & \rightarrow & k_0 \oplus \alpha & \rightarrow & k_1 \oplus \beta & \rightarrow & k_0 \oplus \beta & \rightarrow & k_1 \oplus \beta & \rightarrow & k_0 \oplus \alpha & \rightarrow & k_1 \oplus \beta \\
 & & & & & & & & \downarrow & & & & \\
 k_0 \oplus \alpha & \leftarrow & k_1 \oplus \beta & \leftarrow & k_0 \oplus \beta & \leftarrow & k_1 \oplus \beta & \leftarrow & k_0 \oplus \alpha & \leftarrow & k_1 \oplus \beta & \leftarrow & k_0 \oplus \beta \\
 \alpha \oplus \beta & & \alpha \oplus \beta & & \alpha \oplus \beta & & \alpha \oplus \beta & & \alpha \oplus \beta & & \alpha \oplus \beta & & \alpha \oplus \beta
 \end{array}$$

\* Bozilov et al., PRINCEv2: More Security for (Almost) No Overhead, SAC, 2020

## Minimum latency constrained

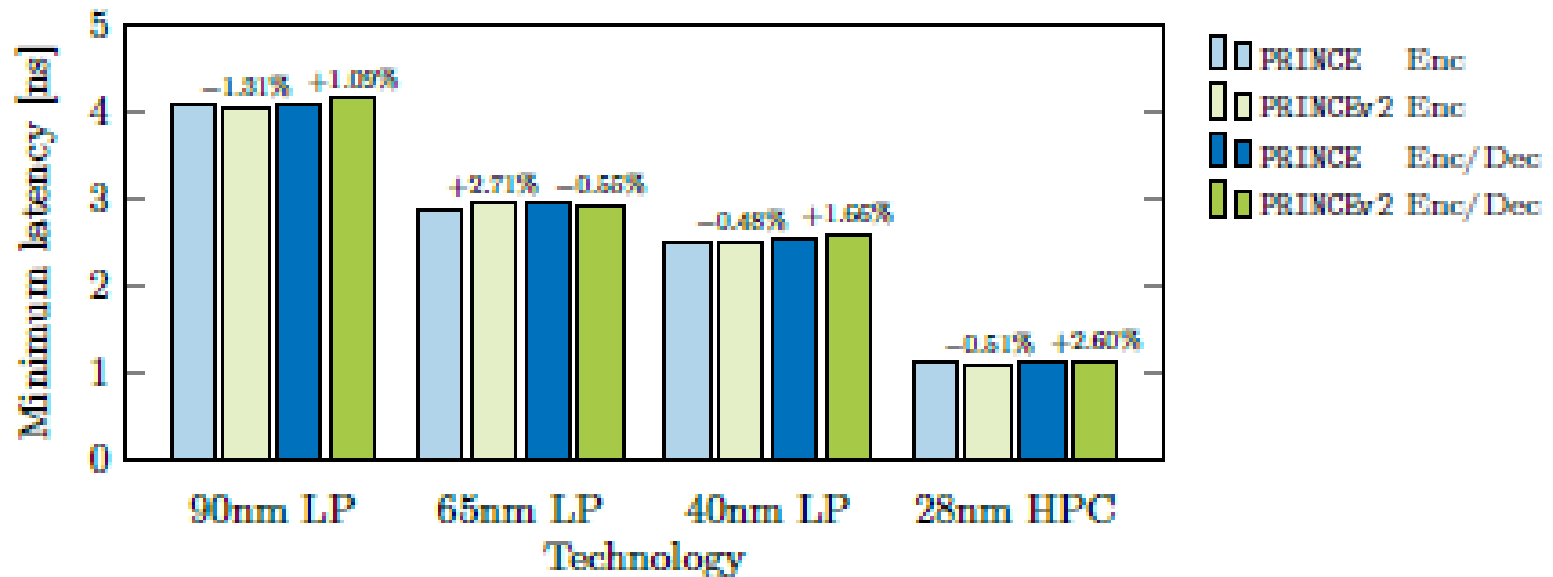
Techn.	Mode	Cipher	Area [GE]	Latency [ns]	Energy [pJ]
90nm LP*	ENC	PRINCE	16244.25	<b>4.101177</b>	1.993172
		PRINCEv2	17661.25	<b>4.047311</b>	2.230068
	ENC/DEC	PRINCE	17808.00	<b>4.106262</b>	2.213275
		PRINCEv2	18888.75	<b>4.151113</b>	2.424250
65nm LP*	ENC	PRINCE	19877.75	<b>2.866749</b>	1.602513
		PRINCEv2	18798.25	<b>2.944367</b>	1.492794
	ENC/DEC	PRINCE	19966.00	<b>2.946442</b>	1.594025
		PRINCEv2	21171.25	<b>2.930153</b>	1.696559
40nm LP*	ENC	PRINCE	17177.00	<b>2.521302</b>	0.617719
		PRINCEv2	16556.50	<b>2.509131</b>	0.592155
	ENC/DEC	PRINCE	17377.50	<b>2.541220</b>	0.630223
		PRINCEv2	17799.50	<b>2.583466</b>	0.648450
28nm HPC**	ENC	PRINCE	38145.33	<b>1.108886</b>	1.258586
		PRINCEv2	33470.33	<b>1.103273</b>	1.108789
	ENC/DEC	PRINCE	35297.67	<b>1.119593</b>	1.181171
		PRINCEv2	38962.33	<b>1.148693</b>	1.299172

\* LP = Low Power

\*\* HPC = High Performance Computing

\* Bozilov et al., PRINCEv2: More Security for (Almost) No Overhead, SAC, 2020

## Minimum latency constrained



\* Bozilov et al., PRINCEv2: More Security for (Almost) No Overhead, SAC, 2020

## Minimum area constrained

Techn.	Mode	Cipher	Area [GE]	Latency [ns]	Energy [pJ]
90 nm LP*	ENC	PRINCE	<b>7937.50</b>	12.859 908	0.569694
		PRINCEv2	<b>8111.25</b>	12.856 450	0.574683
	ENC/DEC	PRINCE	<b>8183.00</b>	14.015 245	0.616671
		PRINCEv2	<b>8440.75</b>	15.513 536	0.628298
65 nm LP*	ENC	PRINCE	<b>8316.00</b>	11.434 771	0.433378
		PRINCEv2	<b>8385.25</b>	11.504 968	0.430286
	ENC/DEC	PRINCE	<b>8547.75</b>	12.349 355	0.440872
		PRINCEv2	<b>8792.75</b>	13.376 949	0.456154
40 nm LP*	ENC	PRINCE	<b>8563.75</b>	10.144 847	0.212027
		PRINCEv2	<b>8608.50</b>	10.063 908	0.207317
	ENC/DEC	PRINCE	<b>8780.00</b>	10.886 960	0.217739
		PRINCEv2	<b>9039.75</b>	11.798 657	0.226534
28 nm HPC**	ENC	PRINCE	<b>8197.00</b>	3.599 936	0.127798
		PRINCEv2	<b>8292.00</b>	3.682 593	0.127786
	ENC/DEC	PRINCE	<b>8426.33</b>	4.260 999	0.131239
		PRINCEv2	<b>8844.67</b>	4.323 993	0.134909

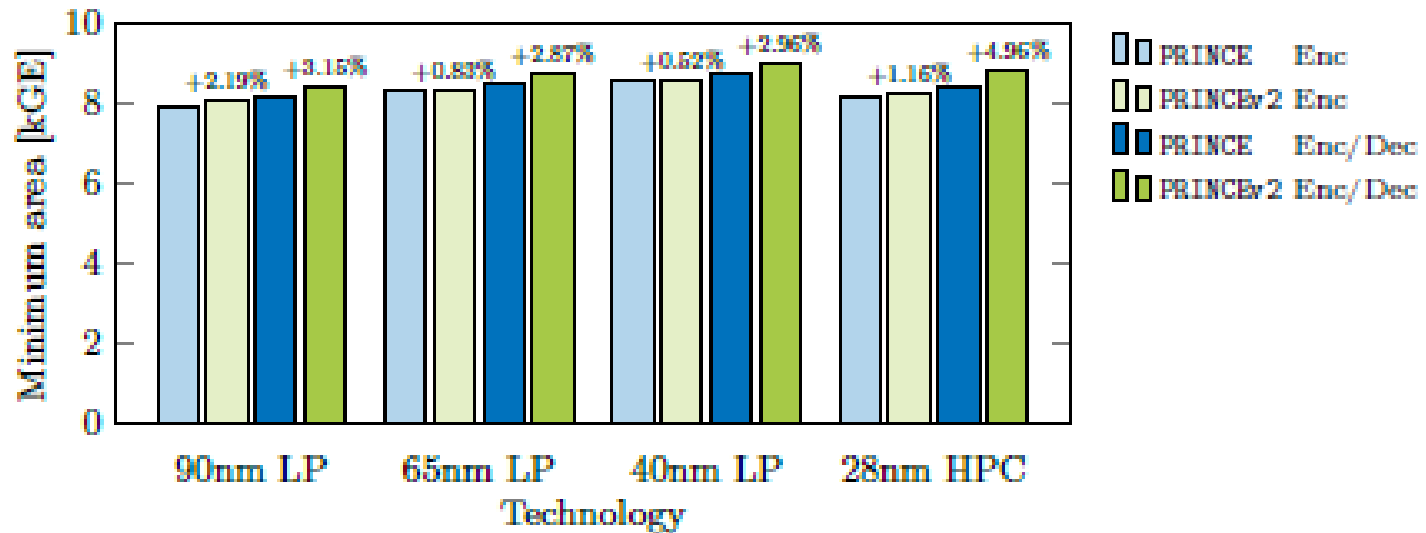
\* LP = Low Power

\*\* HPC = High Performance Computing

\* Bozilov et al., PRINCEv2: More Security for (Almost) No Overhead, SAC, 2020

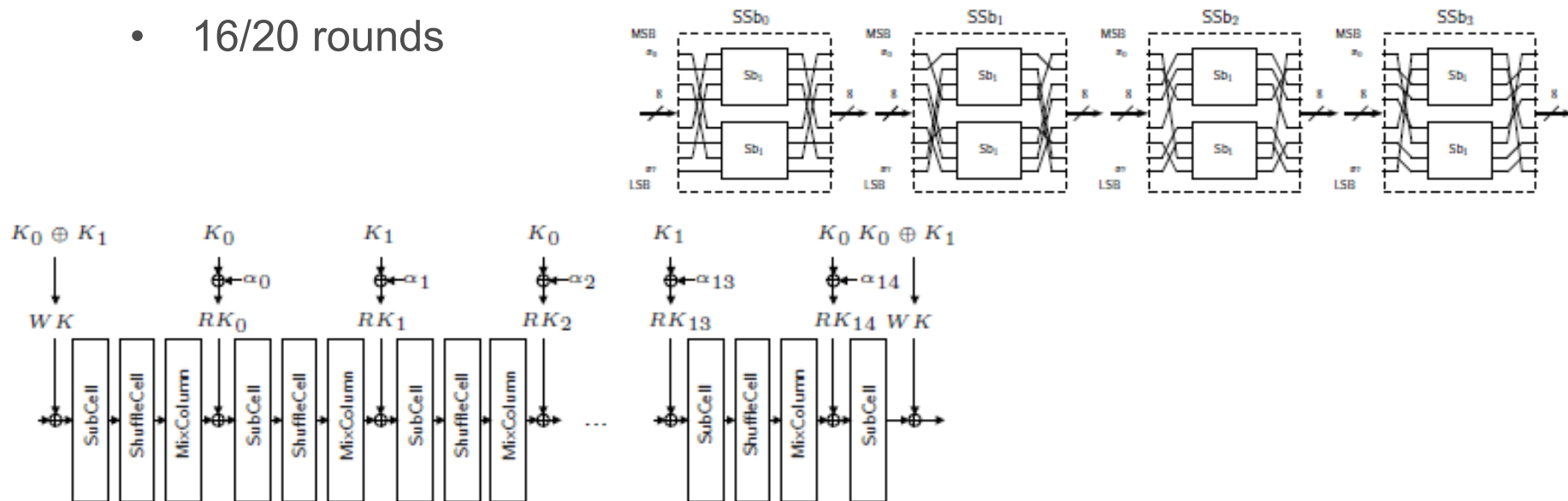


## Minimum area constrained



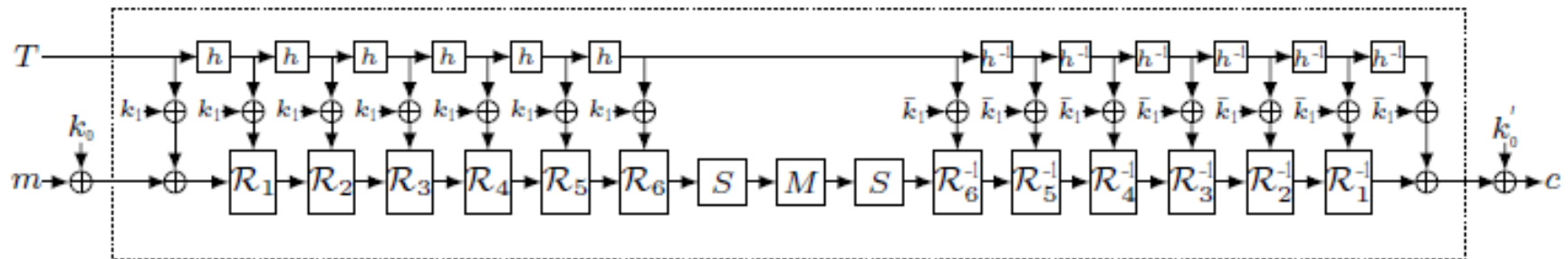
\* Bozilov et al., PRINCEv2: More Security for (Almost) No Overhead, SAC, 2020

- Low energy oriented design
  - Not necessarily for low latency but compared with PRINCE
- Cipher specifics
  - 64/128-bit block, 128-bit key
  - SPN: 2 bijective Sboxes (nonlinear) and involutive binary matrix (linear)
  - 16/20 rounds



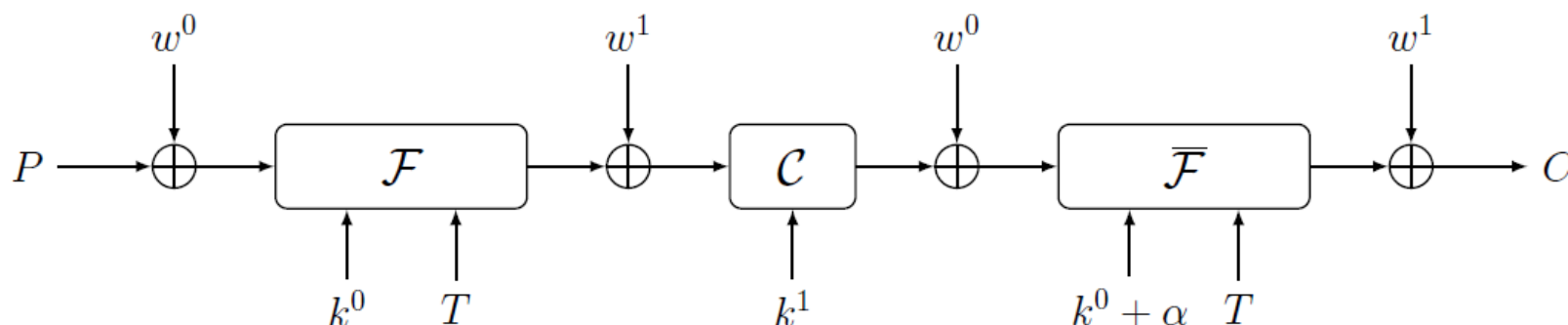
\* Banik et al., Midori: A Block Cipher for Low Energy, ASIACRYPT, 2015

- Turning PRINCE into a tweakable block cipher
  - Well understood when TWEAKEY framework is used
- Cipher specifics
  - 64-bit block, 128-bit key, 64-bit tweak
  - FX-design, SPN
    - Midori Sbox used as it has better latency than PRINCE Sbox
  - 14 rounds (PRINCE-like middle)

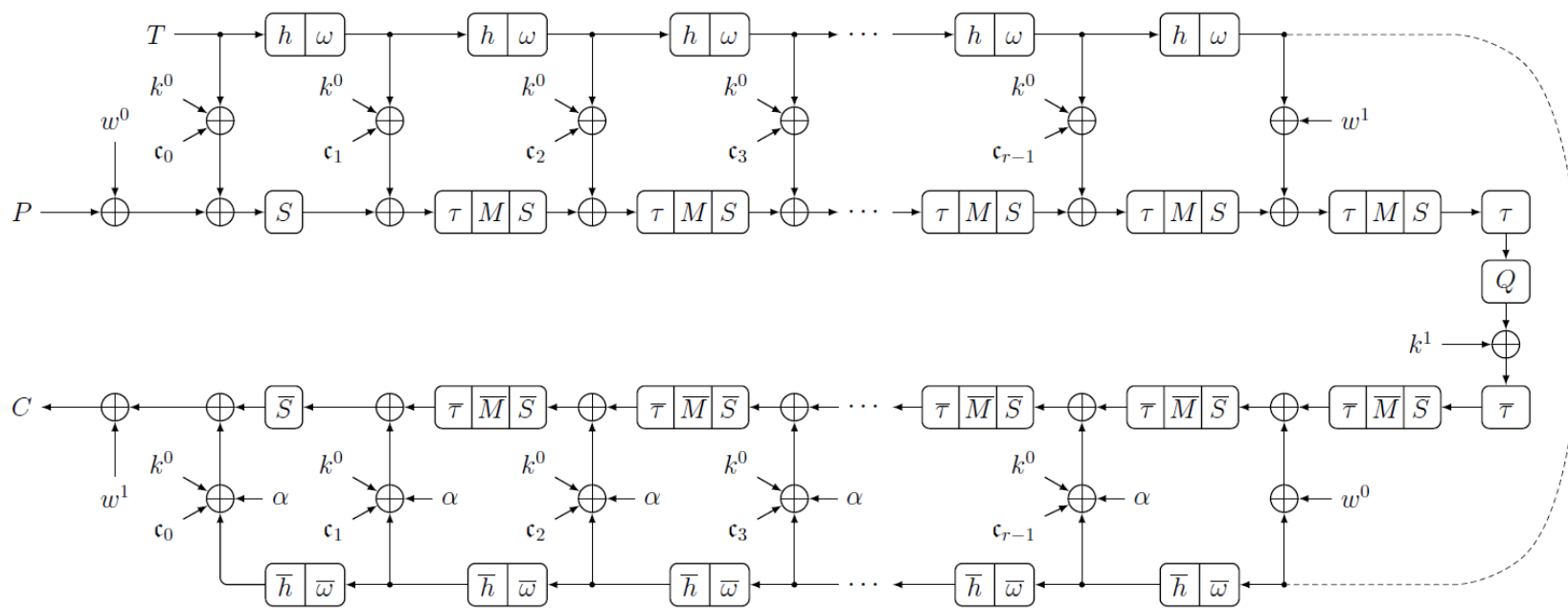


\* Beierle et al., The SKINNY Family of Block Ciphers and its Low-Latency Variant MANTIS, CRYPTO, 2016

- QARMA (from Qualcomm ARM Authenticator)
  - Lightweight tweakable block cipher
  - Primarily known for its use in the ARMv8 architecture
    - For protection of software as a cryptographic hash for the Pointer Authentication Code
- Cipher specifics
  - 64/128-bit block, 128/256-bit key (round numbers 7/10 in permutation)
  - An Even-Mansour cipher using three stages, with whitening keys  $w^0$  and  $w^1$  XORed in between permutation  $\mathcal{F}$  (and its inverse) which uses using core key  $k^0$  and parameterized by a tweak  $T$  and "central" permutation  $\mathcal{C}$  which uses key  $k^1$  and is designed to be reversible via a simple key transformation

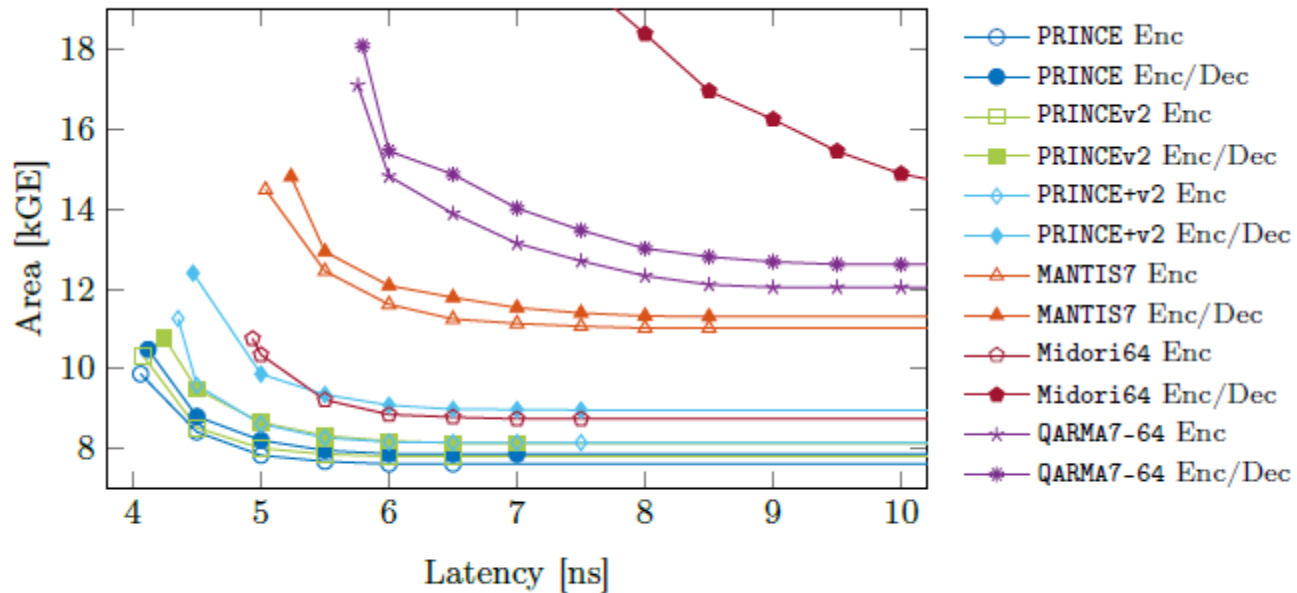


\* Roberto Avanzi, The QARMA Block Cipher Family, ToSC, vol. 17, 2017



\* Roberto Avanzi, The QARMA Block Cipher Family, ToSC, vol. 17, 2017

# Comparison so far...



Comparison of unrolled block ciphers in NanGate 45 nm Open Cell Library.

\* Bozilov et al., PRINCEv2: More Security for (Almost) No Overhead, SAC, 2020

## K-Cipher: A Low Latency, Bit Length Parameterizable Cipher

Michael Kounavis, Sergej Deutsch, Santosh Ghosh, and David Durham

Intel Labs, Intel Corporation, 2111, NE 25th Avenue, Hillsboro, OR 97124  
Email: {michael.e.kounavis, sergej.deutsch, santosh.ghosh, david.durham}@intel.com

**Abstract**—We present the design of a novel low latency, bit length parameterizable cipher, called the “K-Cipher”. K-Cipher is particularly useful to applications that need to support ultra low latency encryption at arbitrary ciphertext lengths. We can think of a range of networking, gaming and computing applications that may require encrypting data at unusual block lengths for many different reasons, such as to make space for other unencrypted state values. Furthermore, in modern applications, encryption is typically required to complete inside stringent time frames in order not to affect performance. K-Cipher has been designed to meet these requirements. In the paper we present the K-Cipher design and discuss its rationale. We also present results from our ongoing security analysis which suggest that only 2 to 4 rounds are sufficient to make the cipher operate securely. Finally, we present synthesis results from 2-round 32-bit and 64-bit K-Cipher encrypt datapaths, produced using Intel’s  $\text{\textcircled{R}}$  10 nm process technology. Our results show that the encrypt datapaths can complete in no more than 767 psec, or 3 clocks in 3.9-4.9 GHz frequencies, and are associated with a maximum area requirement of  $1875 \mu\text{m}^2$ .

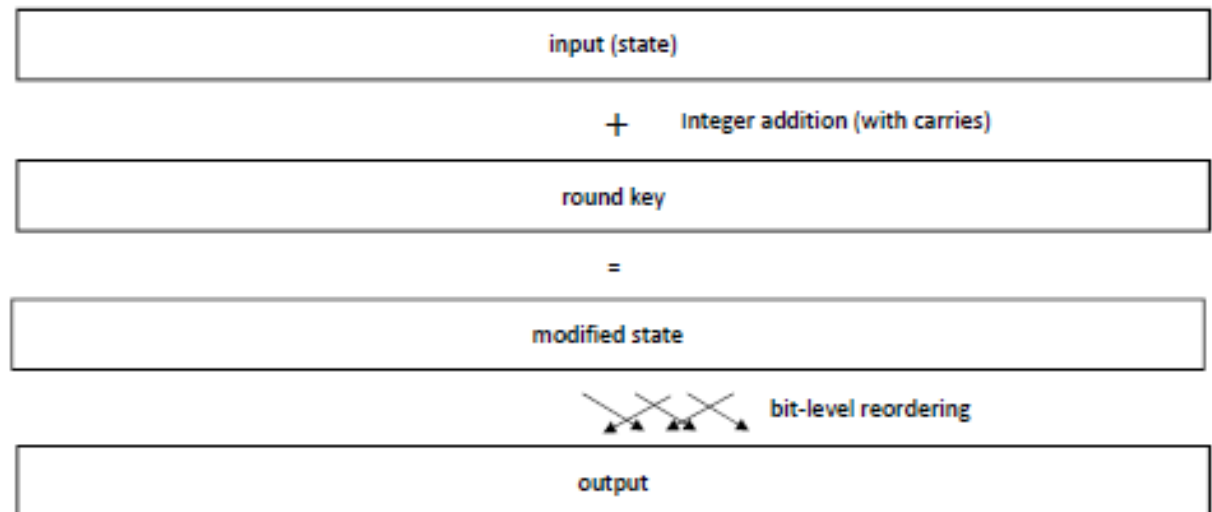


Fig. 1. The Aggressive Adder Component of the K-Cipher Round

## K-Cipher: A Low Latency, Bit Length Parameterizable Cipher

Michael Kounavis, Sergej Deutsch, Santosh Ghosh, and David Durham

Intel Labs, Intel Corporation, 2111, NE 25th Avenue, Hillsboro, OR 97124  
Email: {michael.e.kounavis, sergej.deutsch, santosh.ghosh, david.durham}@intel.com

**Abstract**—We present the design of a novel low latency, bit length parameterizable cipher, called the “K-Cipher”. K-Cipher is particularly useful to applications that need to support ultra low latency encryption at arbitrary ciphertext lengths. We can think of a range of networking, gaming and computing applications that may require encrypting data at unusual block lengths for many different reasons, such as to make space for other unencrypted state values. Furthermore, in modern applications, encryption is typically required to complete inside stringent time frames in order not to affect performance. K-Cipher has been designed to meet these requirements. In the paper we present the K-Cipher design and discuss its rationale. We also present results from our ongoing security analysis which suggest that only 2 to 4 rounds are sufficient to make the cipher operate securely. Finally, we present synthesis results from 2-round 32-bit and 64-bit K-Cipher encrypt datapaths, produced using Intel’s <sup>®</sup> 10 nm process technology. Our results show that the encrypt datapaths can complete in no more than 767 psec, or 3 clocks in 3.9-4.9 GHz frequencies, and are associated with a maximum area requirement of 1875  $\mu\text{m}^2$ .

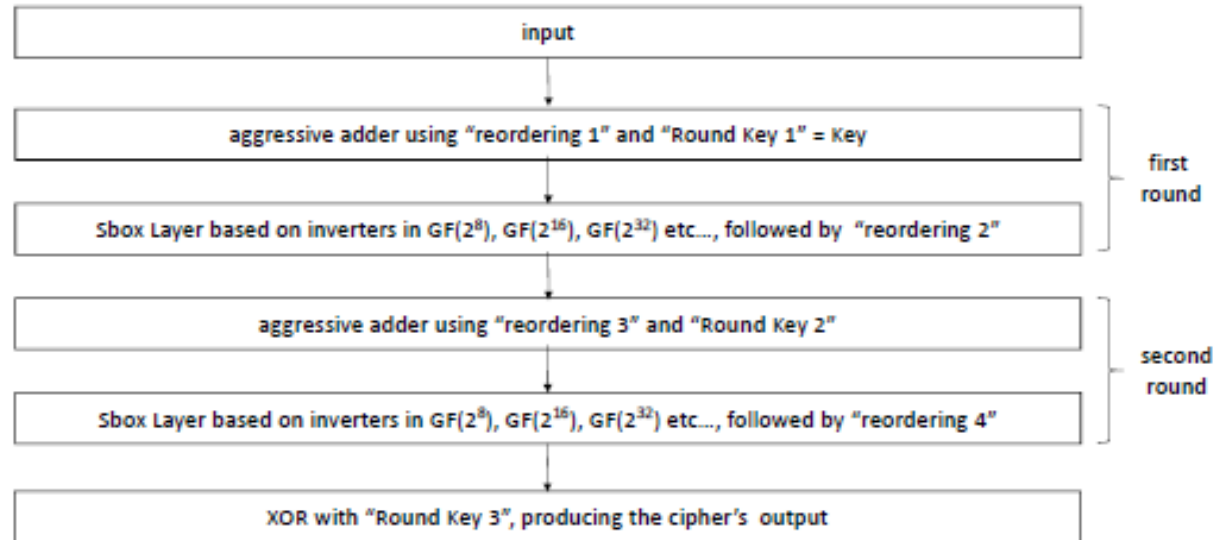


Fig. 2. Two Round K-Cipher Specification

cipher	area ( $\mu\text{m}^2$ )	latency (psec)	number of clocks	freq.
K-Cipher Enc-32, $r = 2$	614	613	3	4.9 GHz
K-Cipher Enc-64, $r = 2$	1875	767	3	3.9 GHz



## The SPEEDY Family of Block Ciphers

Engineering an Ultra Low-Latency Cipher from Gate Level for Secure Processor Architectures

Gregor Leander<sup>1</sup>, Thorben Moos<sup>1</sup>, Amir Moradi<sup>1</sup> and Shahram Rasoolzadeh<sup>\*2</sup>

<sup>1</sup> Ruhr University Bochum, Horst Görtz Institute for IT Security, Bochum, Germany

[firstname.lastname@rub.de](mailto:firstname.lastname@rub.de)

<sup>2</sup> Radboud University, Nijmegen, The Netherlands

[firstname.lastname@ru.nl](mailto:firstname.lastname@ru.nl)

**Abstract.** We introduce **SPEEDY**, a family of ultra low-latency block ciphers. We mix engineering expertise into each step of the cipher's design process in order to create a secure encryption primitive with an extremely low latency in CMOS hardware. The centerpiece of our constructions is a high-speed 6-bit substitution box whose coordinate functions are realized as two-level NAND trees. In contrast to other low-latency block ciphers such as **PRINCE**, **PRINCEv2**, **MANTIS** and **QARMA**, we neither constrain ourselves by demanding decryption at low overhead, nor by requiring a super low area or energy. This freedom together with our gate- and transistor-level considerations allows us to create an ultra low-latency cipher which outperforms all known solutions in single-cycle encryption speed. Our main result, **SPEEDY-6-192**, is a 6-round 192-bit block and 192-bit key cipher which can be executed faster in hardware than any other known encryption primitive (including **Gin11** in Even-Mansour scheme and the **Orthros** pseudorandom function) and offers 128-bit security. One round more, i.e., **SPEEDY-7-192**, provides full 192-bit security. **SPEEDY** primarily targets hardware security solutions embedded in high-end CPUs, where area and energy restrictions are secondary while high performance is the number one priority.

**Keywords:** Low-Latency Cryptography, High-Speed Encryption, Block Cipher

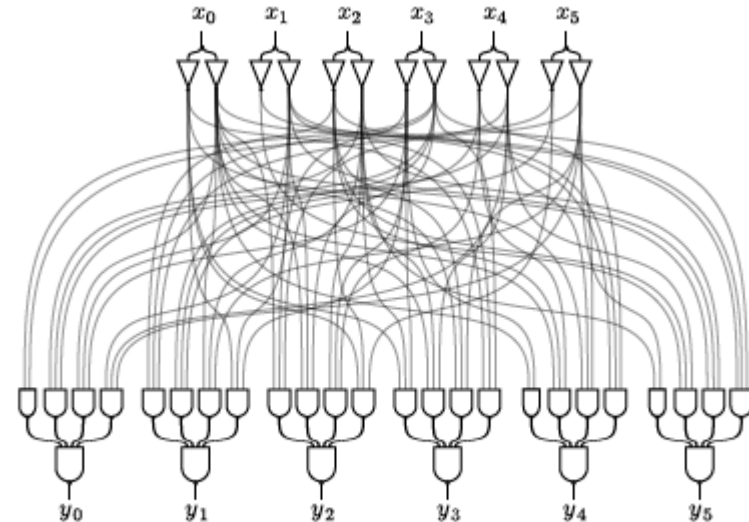
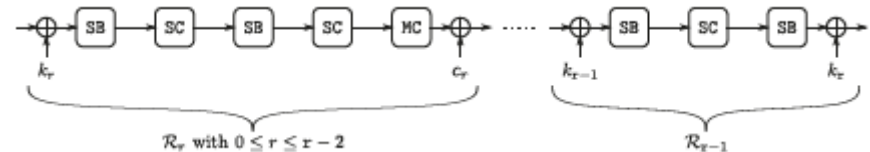


Figure 3: Implementation of the 6-bit S-box of **SPEEDY** based on two-level NAND trees.



Cipher	Minimum Latency [ns]					
	Commercial Foundry				NanGate OCL	
	90 nm LP	65 nm LP	40 nm LP	28 nm HPC	45 nm	15 nm
<b>GIN11 E-M</b>	4.532467	3.330192	2.794736	1.178424	4.537304	0.435069
<b>NANTIS<sub>6</sub></b>	4.625529	3.405490	2.891383	1.278725	4.479773	0.437595
<b>NANTIS<sub>7</sub></b>	5.201681	3.722473	3.234409	1.421365	5.074452	0.492703
<b>NANTIS<sub>8</sub></b>	5.823127	4.233543	3.631438	1.594997	5.739020	0.552384
<b>Nider1</b>	5.061255	3.582221	3.142355	1.362237	4.934847	0.481522
<b>Orthros</b>	3.862139	2.678637	2.401275	1.087139	3.774836	0.369497
<b>PRINCE</b>	4.101177	2.866749	2.521302	1.108886	4.059997	0.389144
<b>PRINCEv2</b>	4.047311	2.944367	2.509131	1.103273	4.077636	0.387146
<b>QARMA<sub>5</sub>-64-<math>\sigma_0</math></b>	4.075846	2.920377	2.498908	1.134901	4.014516	0.385281
<b>QARMA<sub>6</sub>-64-<math>\sigma_0</math></b>	4.770325	3.418600	2.951308	1.308331	4.554445	0.448931
<b>QARMA<sub>7</sub>-64-<math>\sigma_0</math></b>	5.449707	3.909138	3.389576	1.538606	5.336362	0.517093
<b>QARMA<sub>8</sub>-64-<math>\sigma_0</math></b>	6.103768	4.396543	3.814078	1.697027	5.966323	0.575525
<b>QARMA<sub>5</sub>-64-<math>\sigma_1</math></b>	4.515514	3.284252	2.815788	1.219624	4.367899	0.408580
<b>QARMA<sub>6</sub>-64-<math>\sigma_1</math></b>	5.297867	3.808675	3.271455	1.388353	4.944635	0.472798
<b>QARMA<sub>7</sub>-64-<math>\sigma_1</math></b>	6.014477	4.371963	3.745959	1.601572	5.800633	0.542712
<b>QARMA<sub>8</sub>-64-<math>\sigma_1</math></b>	6.720944	4.904521	4.202632	1.797539	6.498429	0.608985
<b>SPEEDY-5-192</b>	2.994643	2.178075	1.867064	0.847761	3.187368	0.300466
<b>SPEEDY-6-192</b>	3.637978	2.639186	2.277422	1.032206	3.848132	0.366762
<b>SPEEDY-7-192</b>	4.261928	3.087257	2.663004	1.217946	4.515505	0.431032
<b>SPEEDY-5-192 *</b>	2.941130	2.121748	1.820950	0.826217	2.817971	0.290961
<b>SPEEDY-6-192 *</b>	3.559981	2.573561	2.223863	1.011173	3.382270	0.353391
<b>SPEEDY-7-192 *</b>	4.174183	3.029217	2.620612	1.186598	3.995325	0.413950

\* = Optimized HDL code with direct instantiation of library cells based on Figures 3 and 4.

Cipher	Area [GE]					
	Commercial Foundry				NanGate OCL	
	90 nm LP	65 nm LP	40 nm LP	28 nm HPC	45 nm	15 nm
<b>GIN11 E-M</b>	72644.00	82781.00	63100.50	144036.33	52038.67	57551.25
<b>NANTIS<sub>6</sub></b>	21045.75	23264.50	20448.25	36073.33	12660.67	15954.00
<b>NANTIS<sub>7</sub></b>	23229.25	26385.75	23192.50	43220.33	14225.67	17522.50
<b>NANTIS<sub>8</sub></b>	26365.75	30316.75	25429.75	50793.00	15663.33	19707.50
<b>Nider1</b>	18678.50	21964.00	17562.25	41450.67	10675.33	13927.25
<b>Orthros</b>	49639.75	61657.00	44715.75	74384.67	31317.33	39165.00
<b>PRINCE</b>	16244.25	19877.75	17177.00	38145.33	9873.33	13291.00
<b>PRINCEv2</b>	17661.25	18798.25	16556.50	33470.33	10332.00	13069.50
<b>QARMA<sub>5</sub>-64-<math>\sigma_0</math></b>	19590.75	21706.75	20255.00	31703.00	11824.67	14880.75
<b>QARMA<sub>6</sub>-64-<math>\sigma_0</math></b>	22624.25	25349.50	22689.00	38813.67	14165.67	17621.75
<b>QARMA<sub>7</sub>-64-<math>\sigma_0</math></b>	25614.00	29323.00	24656.25	40494.33	15769.33	19770.25
<b>QARMA<sub>8</sub>-64-<math>\sigma_0</math></b>	28813.75	32780.75	28262.75	47952.33	17908.00	22074.00
<b>QARMA<sub>5</sub>-64-<math>\sigma_1</math></b>	20264.75	23753.00	20202.25	34302.00	12350.33	15588.75
<b>QARMA<sub>6</sub>-64-<math>\sigma_1</math></b>	23162.25	26941.25	23333.75	45419.00	15066.00	18164.00
<b>QARMA<sub>7</sub>-64-<math>\sigma_1</math></b>	26563.75	31495.00	27059.50	52108.00	16641.00	20670.25
<b>QARMA<sub>8</sub>-64-<math>\sigma_1</math></b>	30534.50	35787.75	29116.50	54967.00	18963.67	22761.75
<b>SPEEDY-5-192</b>	47364.00	53856.00	47528.50	74467.00	27903.33	34649.00
<b>SPEEDY-6-192</b>	57322.00	64438.25	56816.00	88932.00	34085.00	41443.25
<b>SPEEDY-7-192</b>	68370.00	75273.00	65422.00	95235.67	39853.33	48727.75
<b>SPEEDY-5-192 *</b>	49902.00	58796.25	55846.75	80313.33	29839.00	38075.25
<b>SPEEDY-6-192 *</b>	59688.00	70653.00	66553.00	98950.00	36523.33	46266.50
<b>SPEEDY-7-192 *</b>	73397.75	84745.00	77519.75	111754.33	42813.33	54193.25

\* = Optimized HDL code with direct instantiation of library cells based on Figures 3 and 4.

## Orthros: A Low-Latency PRF

Subhadeep Banik<sup>1</sup> and Takanori Isobe<sup>2,3,4</sup> and  
Fukang Liu<sup>2,5</sup> and Kazuhiko Minematsu<sup>6</sup> and Kosei Sakamoto<sup>2</sup>

<sup>1</sup> LASEC, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland.  
[subhadeep.banik@epfl.ch](mailto:subhadeep.banik@epfl.ch)

<sup>2</sup> University of Hyogo, Kobe, Japan. [takanori.isobe@ai.u-hyogo.ac.jp](mailto:takanori.isobe@ai.u-hyogo.ac.jp),  
[liufukangs@gmail.com](mailto:liufukangs@gmail.com), [k.sakamoto0728@gmail.com](mailto:k.sakamoto0728@gmail.com)

<sup>3</sup> NICT, Tokyo, Japan

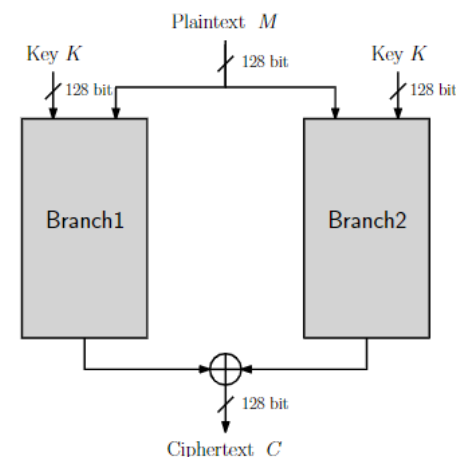
<sup>4</sup> PRESTO, Japan Science and Technology Agency, Tokyo, Japan

<sup>5</sup> East China Normal University, Shanghai, China

<sup>6</sup> NEC, Kawasaki, Japan [k-minematsu@nec.com](mailto:k-minematsu@nec.com)

**Abstract.** We present Orthros, a 128-bit block pseudorandom function. It is designed with primary focus on latency of fully unrolled circuits. For this purpose, we adopt a parallel structure comprising two keyed permutations. The round function of each permutation is similar to Midori, a low-energy block cipher, however we thoroughly revise it to reduce latency, and introduce different rounds to significantly improve cryptographic strength in a small number of rounds. We provide a comprehensive, dedicated security analysis. For hardware implementation, Orthros achieves the lowest latency among the state-of-the-art low-latency primitives. For example, using the STM 90nm library, Orthros achieves a minimum latency of around 2.4 ns, while other constructions like PRINCE, Midori-128 and QARMA<sub>0</sub>-128- $\sigma_0$  achieve 2.56 ns, 4.10 ns, 4.38 ns respectively.

**Keywords:** Pseudorandom Function · Low Latency · Lightweight Cryptography · Sum of Permutations



## Orthros: A Low-Latency PRF

Subhadeep Banik<sup>1</sup> and Takanori Isobe<sup>2,3,4</sup> and  
Fukang Liu<sup>2,5</sup> and Kazuhiko Minematsu<sup>6</sup> and Kosei Sakamoto<sup>2</sup>

<sup>1</sup> LASEC, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland.  
[subhadeep.banik@epfl.ch](mailto:subhadeep.banik@epfl.ch)

<sup>2</sup> University of Hyogo, Kobe, Japan. [takanori.isobe@ai.u-hyogo.ac.jp](mailto:takanori.isobe@ai.u-hyogo.ac.jp),  
[liufukangs@gmail.com](mailto:liufukangs@gmail.com), [k.sakamoto0728@gmail.com](mailto:k.sakamoto0728@gmail.com)

<sup>3</sup> NICT, Tokyo, Japan

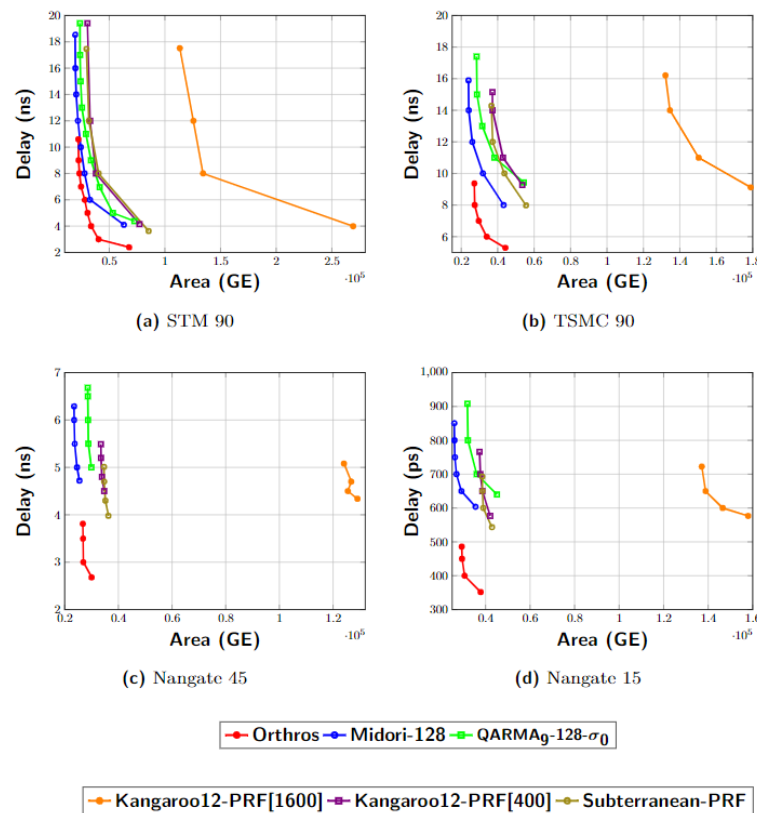
<sup>4</sup> PRESTO, Japan Science and Technology Agency, Tokyo, Japan

<sup>5</sup> East China Normal University, Shanghai, China

<sup>6</sup> NEC, Kawasaki, Japan [k-minematsu@nec.com](mailto:k-minematsu@nec.com)

**Abstract.** We present Orthros, a 128-bit block pseudorandom function. It is designed with primary focus on latency of fully unrolled circuits. For this purpose, we adopt a parallel structure comprising two keyed permutations. The round function of each permutation is similar to Midori, a low-energy block cipher, however we thoroughly revise it to reduce latency, and introduce different rounds to significantly improve cryptographic strength in a small number of rounds. We provide a comprehensive, dedicated security analysis. For hardware implementation, Orthros achieves the lowest latency among the state-of-the-art low-latency primitives. For example, using the STM 90nm library, Orthros achieves a minimum latency of around 2.4 ns, while other constructions like PRINCE, Midori-128 and QARMA<sub>9</sub>-128- $\sigma_0$  achieve 2.56 ns, 4.10 ns, 4.38 ns respectively.

**Keywords:** Pseudorandom Function · Low Latency · Lightweight Cryptography · Sum of Permutations



## SCARF – A Low-Latency Block Cipher for Secure Cache-Randomization

Federico Canale<sup>1</sup>, Tim Güneysu<sup>1,4</sup>, Gregor Leander<sup>1</sup>, Jan Philipp Thoma<sup>1</sup>,  
Yosuke Todo<sup>2</sup>, and Rei Ueno<sup>3</sup>

<sup>1</sup> Ruhr University Bochum, Bochum, Germany [firstname.lastname@rub.de](mailto:firstname.lastname@rub.de)

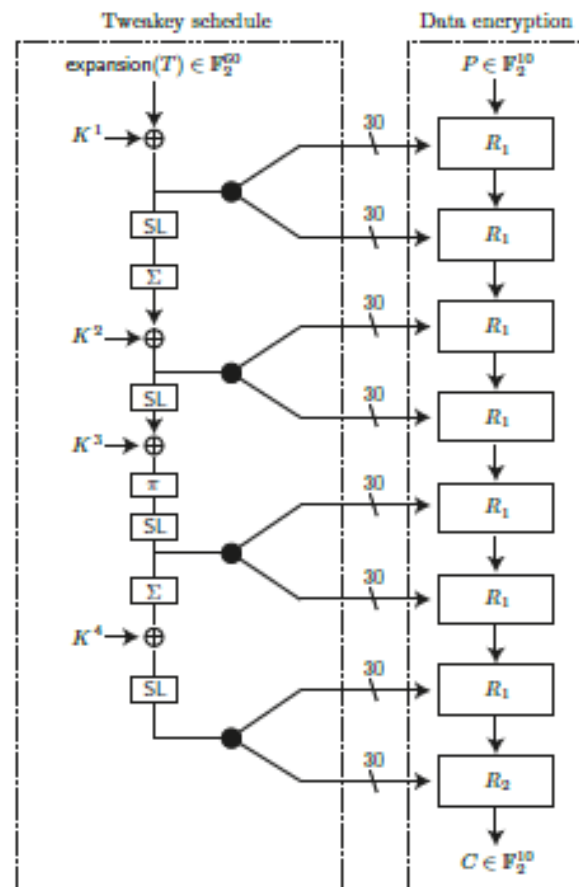
<sup>2</sup> NTT Social Informatics Laboratories, Tokyo, Japan [yosuke.todo@ntt.com](mailto:yosuke.todo@ntt.com)

<sup>3</sup> Tohoku University, Sendai-shi, Japan. [re1.ueno.a8@tohoku.ac.jp](mailto:re1.ueno.a8@tohoku.ac.jp)

<sup>4</sup> DFKI, Bremen, Germany.

**Abstract.** Randomized cache architectures have proven to significantly increase the complexity of contention-based cache side-channel attacks and therefore present an important building block for side-channel secure microarchitectures. By randomizing the address-to-cache-index mapping, attackers can no longer trivially construct minimal eviction sets which are fundamental for contention-based cache attacks. At the same time, randomized caches maintain the flexibility of traditional caches, making them broadly applicable across various CPU types. This is a major advantage over cache partitioning approaches.

A large variety of randomized cache architectures has been proposed. However, the actual randomization function received little attention and is often neglected in these proposals. Since the randomization operates directly on the critical path of the cache lookup, the function needs to have extremely low latency. At the same time, attackers must not be able to bypass the randomization which would nullify the security benefit of the randomized mapping. In this paper, we propose SCARF (Secure Cache Randomization Function), the first dedicated cache randomization cipher which achieves low latency and is cryptographically secure in the cache attacker model. The design methodology for this dedicated cache cipher enters new territory in the field of block ciphers with a small 10-bit block length and heavy key-dependency in few rounds.



## SCARF – A Low-Latency Block Cipher for Secure Cache-Randomization

Federico Canale<sup>1</sup>, Tim Güneysu<sup>1,4</sup>, Gregor Leander<sup>1</sup>, Jan Philipp Thoma<sup>1</sup>,  
Yosuke Todo<sup>2</sup>, and Rei Ueno<sup>3</sup>

<sup>1</sup> Ruhr University Bochum, Bochum, Germany [firstname.lastname@rub.de](mailto:firstname.lastname@rub.de)

<sup>2</sup> NTT Social Informatics Laboratories, Tokyo, Japan [yosuke.todo@ntt.com](mailto:yosuke.todo@ntt.com)

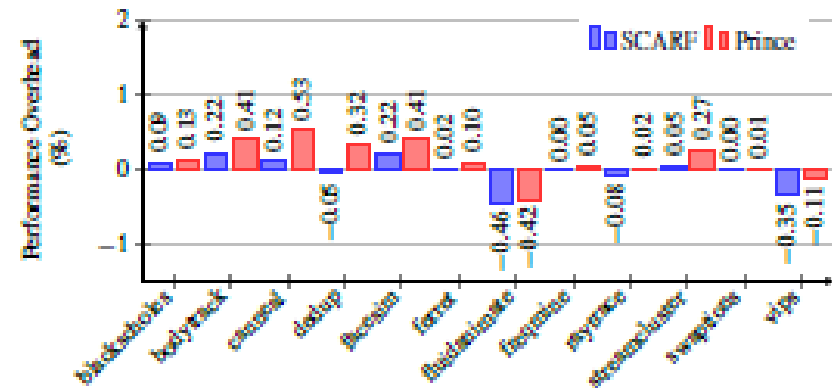
<sup>3</sup> Tohoku University, Sendai-shi, Japan. [re1.ueno.a@tohoku.ac.jp](mailto:re1.ueno.a@tohoku.ac.jp)

<sup>4</sup> DFKI, Bremen, Germany.


**Abstract.** Randomized cache architectures have proven to significantly increase the complexity of contention-based cache side-channel attacks and therefore present an important building block for side-channel secure microarchitectures. By randomizing the address-to-cache-index mapping, attackers can no longer trivially construct minimal eviction sets which are fundamental for contention-based cache attacks. At the same time, randomized caches maintain the flexibility of traditional caches, making them broadly applicable across various CPU types. This is a major advantage over cache partitioning approaches.

A large variety of randomized cache architectures has been proposed. However, the actual randomization function received little attention and is often neglected in these proposals. Since the randomization operates directly on the critical path of the cache lookup, the function needs to have extremely low latency. At the same time, attackers must not be able to bypass the randomization which would nullify the security benefit of the randomized mapping. In this paper, we propose SCARF (Secure Cache Randomization Function), the first dedicated cache randomization cipher which achieves low latency and is cryptographically secure in the cache attacker model. The design methodology for this dedicated cache cipher enters new territory in the field of block ciphers with a small 10-bit block length and heavy key-dependency in few rounds.

Technology	45 nm		15 nm	
	Latency [ns]	Area [GE]	Latency [ps]	Area [GE]
PRINCE	4.74	12,554	628.49	17,484
MANTIS6	4.73	13,129	630.07	17,641
QARMA5	4.40	13,915	563.62	18,455
SCARF	2.26	7,335	305.76	8,118



## LLWBC: A New Low-Latency Light-Weight Block Cipher

Lei Zhang<sup>1,2</sup> , Ruichen Wu<sup>1</sup>, Yuhan Zhang<sup>1</sup>, Yafei Zheng<sup>1,2</sup>,  
and Wenling Wu<sup>1</sup>

<sup>1</sup> Institute of Software, Chinese Academy of Sciences, Beijing 100190, China  
zhanglei@iscas.ac.cn

<sup>2</sup> State Key Laboratory of Cryptology, P. O. Box 5159, Beijing, China

**Abstract.** Lightweight cipher suitable for resource constrained environment is crucial to the security of applications such as RFID, Internet of Things, etc. Moreover, in recent years low-latency is becoming more important and highly desirable by some specific applications which need instant response and real-time security. In this paper, we propose a new low-latency block cipher named LLLWBC. Similar to other known low-latency block ciphers, LLLWBC preserves the important  $\alpha$ -reflection property, namely the decryption for a key  $K$  is equal to encryption with a key  $K \oplus \alpha$  where  $\alpha$  is a fixed constant. However, instead of the normally used SP-type construction, the core cipher employs a variant of generalized Feistel structure called extended GFS. It has 8 branches and employs byte-wise round function and nibble-wise round permutation iterated for 21 rounds. We choose the round permutations carefully together with a novel key schedule to guarantee the  $\alpha$ -reflection property. This allows an efficient fully unrolled implementation of LLLWBC in hardware and the overhead of decryption on top of encryption is negligible. Moreover, because of the involutory property of extended GFS, the inverse round function is not needed, which makes it possible to be implemented in round-based architecture with a competitive area cost. Furthermore, our security evaluation shows that LLLWBC can achieve enough security margin within the constraints of security claims. Finally, we evaluate the hardware and software performances of LLLWBC on various platforms and a brief comparison with other low-latency ciphers is also presented.

**Keywords:** Block cipher · Low-latency · Lightweight · Extended GFS

**Table 7.** Performance results of fully unrolled version of LLLWBC and other ciphers.

Cipher	Technology	Latency(ns)	Area(GE)	Source
LLWBC	NanGate 45 nm Generic	11.76	8226.85	This paper
PRINCE	NanGate 45 nm Generic	–	8263	[6]
MANTIS <sub>7</sub>	UMC L180 0.18 $\mu$ m 1P6M	20.50	11209	[4]
QARMA <sub>7</sub>	FinFet 7 nm	6.04	17109	[1]

**Table 8.** Performance results of round-based version of LLLWBC and PRINCE.

Cipher	Technology	Latency(ns)	Area(GE)	Source
LLWBC	NanGate 45 nm Generic	0.64	1024.10	This paper
PRINCE	NanGate 45 nm Generic	–	3779	[6]

## Introducing two Low-Latency Cipher Families: Sonic and SuperSonic

Yanis Belkheyar<sup>1</sup>, Joan Daemen<sup>1</sup>, Christoph Dobraunig<sup>2</sup>, Santosh Ghosh<sup>2</sup>  
and Shahram Rasoolzadeh<sup>1</sup>

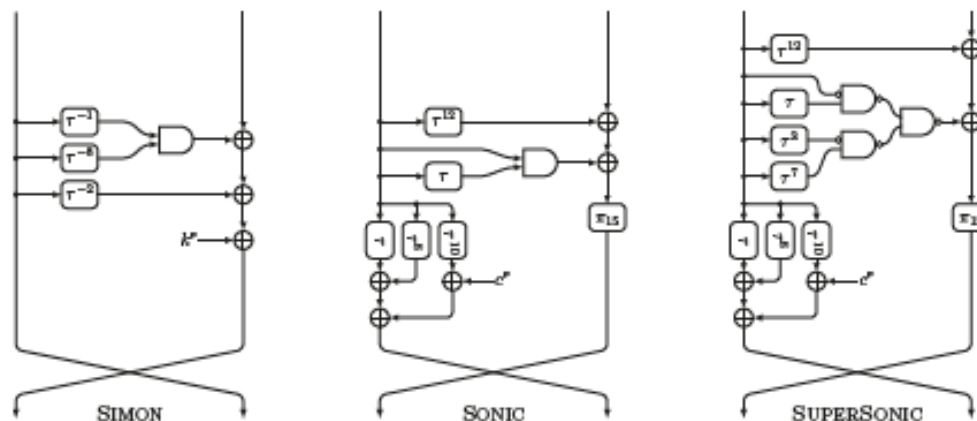
<sup>1</sup> Digital Security Group, Radboud University, Nijmegen, The Netherlands  
[firstname.lastname@ru.nl](mailto:firstname.lastname@ru.nl)

<sup>2</sup> Intel Labs, Hillsboro, USA [firstname.lastname@intel.com](mailto:firstname.lastname@intel.com)

**Abstract.** For many latency-critical operations in computer systems, like memory reads/writes, adding encryption can have a big impact on the performance. Hence, the existence of cryptographic primitives with good security properties and minimal latency is a key element in the wide-spread implementation of such security measures. In this paper, we introduce two new families of low-latency permutations/block ciphers called SONIC and SUPERSONIC, inspired by the SIMON block ciphers.

**Keywords:** low-latency, Simon, Sonic, SuperSonic, Feistel structure, gate-delay-balanced Feistel, block cipher

cipher	word size ( $w$ ) [bits]	block size ( $b$ ) [bits]	number of rounds	target security [bits]
SONIC256	128	256	24	128
SONIC512	256	512	24	128
SUPERSONIC256	128	256	21	128
SUPERSONIC512	256	512	21	128



**Figure 1:** Comparison of round functions of SIMON, SONIC, and SUPERSONIC.



## BipBip: A Low-Latency Tweakable Block Cipher with Small Dimensions

Yanis Belkheayar<sup>1</sup>, Joan Daemen<sup>1</sup>, Christoph Dobraunig<sup>2</sup>, Santosh Ghosh<sup>2</sup>  
and Shahram Rasoolzadeh<sup>1</sup>

<sup>1</sup> Digital Security Group, Radboud University, Nijmegen, The Netherlands  
[firstname.lastname@ru.nl](mailto:firstname.lastname@ru.nl)

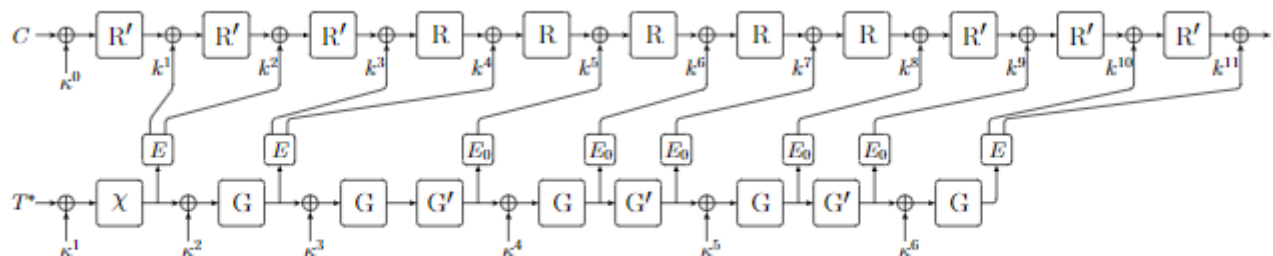
<sup>2</sup> Intel Labs, Hillsboro, USA [firstname.lastname@intel.com](mailto:firstname.lastname@intel.com)

**Abstract.** Recently, a memory safety concept called Cryptographic Capability Computing ( $C^3$ ) has been proposed.  $C^3$  is the first memory safety mechanism that works without requiring extra storage for metadata and hence, has the potential to significantly enhance the security of modern IT-systems at a rather low cost. To achieve this,  $C^3$  heavily relies on ultra-low-latency cryptographic primitives. However, the most crucial primitive required by  $C^3$  demands uncommon dimensions. To partially encrypt 64-bit pointers, a 24-bit tweakable block cipher with a 40-bit tweak is needed. The research on low-latency tweakable block ciphers with such small dimensions is not very mature. Therefore, designing such a cipher provides a great research challenge, which we take on with this paper. As a result, we present BipBip, a 24-bit tweakable block cipher with a 40-bit tweak that allows for ASIC implementations with a latency of 3 cycles at a 4.5 GHz clock frequency on a modern 10 nm CMOS technology.

**Keywords:** BipBip · low-latency · tweakable block cipher

**Table 9:** Unrolled implementation results on Intel 10nm with 0.85 V and 100°C.

Cipher	Critical Path		Area [GE]	Power [mW]
	Gate Levels	Delay [ps]		
Prince Enc/Dec	74	853	7542	42.71
<b>BipBip Dec</b>	<b>48</b>	<b>622</b>	<b>5741</b>	15.91
BipBip Enc	148	1523	10776	19.23



**Figure 2:** Structure of BipBip.



## IoVCipher: A low-latency lightweight block cipher for internet of vehicles

Xiantong Huang<sup>a,b</sup>, Lang Li<sup>a,b,\*</sup>, Hong Zhang<sup>a,b</sup>, Jinling Yang<sup>a,b</sup>, Juanli Kuang<sup>a,b,c</sup>

<sup>a</sup> College of Computer Science and Technology, Hengyang Normal University, Hengyang 421002, China

<sup>b</sup> Hunan Provincial Key Laboratory of Intelligent Information Processing and Application, Hengyang Normal University, Hengyang 421002, China

<sup>c</sup> Faculty of Innovation Engineering, Macau University of Science and Technology, 999078, Macao Special Administrative Region of China

### ARTICLE INFO

#### Keywords:

Internet of vehicles  
Low latency  
Lightweight block cipher  
Automotive security  
Electronic control units

### ABSTRACT

The data security of CAN bus system is receiving increasing attention with the rapid development of Internet of Vehicles (IoV). However, traditional ciphers are not the best choice due to the limitations of computation, real-time, and resources of Electronic Control Units in vehicles. Thus, this paper proposes a lightweight block cipher IoVCipher to protect the security of IoV. It is designed focus on the latency and area in round-based architectures (both encryption and decryption) to meet this resource-constrained environments. For this purpose, two S-boxes with low latency and tiny area are constructed in this paper, one involution and one non-involution. Considering the decryption latency, a low latency subkey generation method is designed. In addition, this paper proposes a new extended MISTY structure that makes the encryption and decryption of hardware implementations similar. In comparison to other low-latency lightweight block ciphers such as PRINCE, QARMA, MANTIS and LLLWBC, IoVCipher achieves an effective balance between latency and area in the round-based architecture, and IoVCipher has low latency, low area, and low energy in the fully unrolled architecture. Finally, IoVCipher is implemented on a real-time speed acquisition and encryption testbed to simulate encrypted transmission of real-time speed in a CAN bus environment.

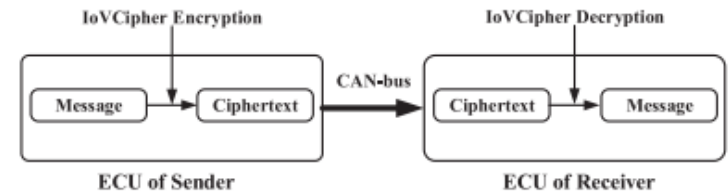
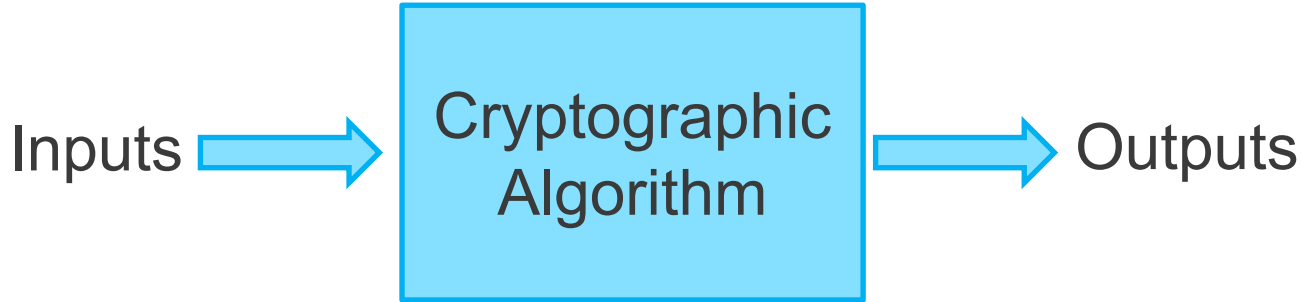


Fig. 4. Encryption and decryption on CAN bus messages.

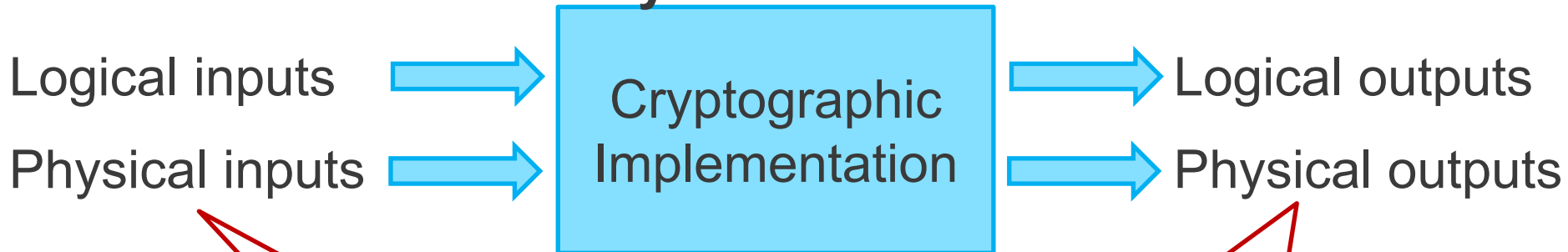
Implementation results for fully unrolled architecture (encryption-only). Estimated for 100 MHz operation.

Algorithm	Area (GEs)	Latency (ns)	Power (mW)	Energy (pJ)	Throughput (MBit/s)
QARMA <sub>6</sub> -σ <sub>0</sub> [18]	14 165.67	4.55	0.41	40.99	14 065.93
QARMA <sub>7</sub> -σ <sub>0</sub> [18]	15 769.33	5.34	0.45	45.29	11 985.02
QARMA <sub>8</sub> -σ <sub>0</sub> [18]	17 908.00	5.97	0.51	51.21	10 720.27
QARMA <sub>6</sub> -σ <sub>1</sub> [18]	15 066.00	4.94	0.44	43.50	12 955.47
QARMA <sub>7</sub> -σ <sub>1</sub> [18]	16 641.00	5.80	0.48	47.69	11 034.48
QARMA <sub>8</sub> -σ <sub>1</sub> [18]	18 963.67	6.50	0.54	54.18	9846.15
MANTIS <sub>6</sub> [18]	12 660.67	4.48	0.37	36.80	14 285.71
MANTIS <sub>7</sub> [18]	14 225.67	5.07	0.41	41.07	12 623.27
MANTIS <sub>8</sub> [18]	15 663.33	5.74	0.45	44.79	11 149.83
PRINCE [19]	12 554.00	4.74	0.36	36.41	13 502.11
LLLWBC	17 093.19	8.20	0.50	49.57	7804.88
<b>IoVCipher</b>	<b>8868.00</b>	<b>4.07</b>	<b>0.26</b>	<b>25.72</b>	<b>15 724.82</b>

## Classical attacks

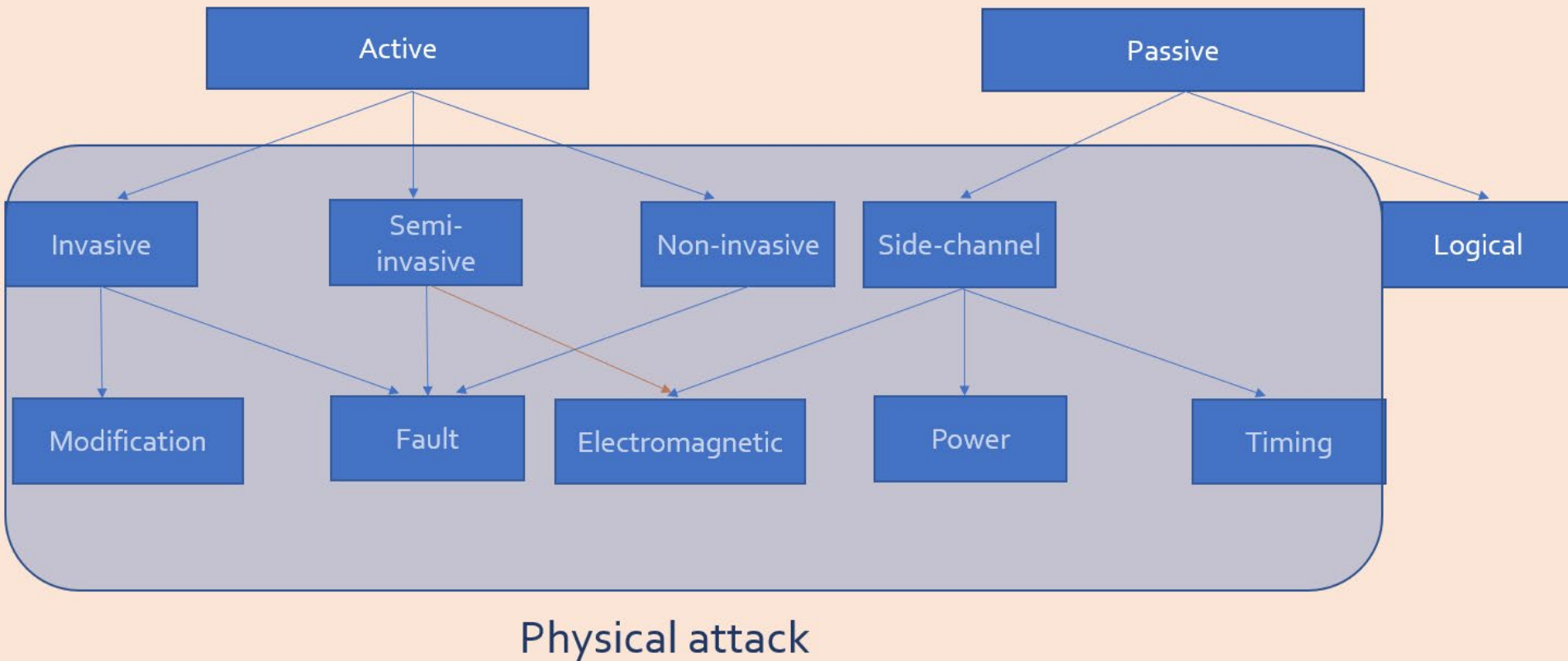


## Physical attacks



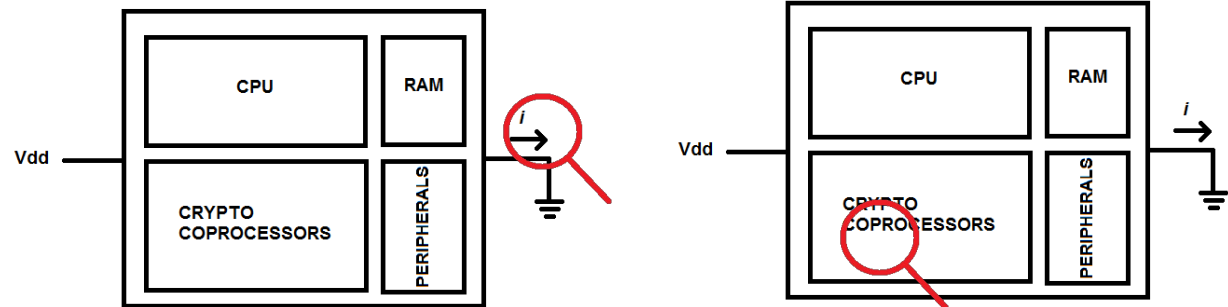
**Fault-injection attacks**

**Side-channel attacks**



## Attack-resistant Lightweight Crypto Implementations

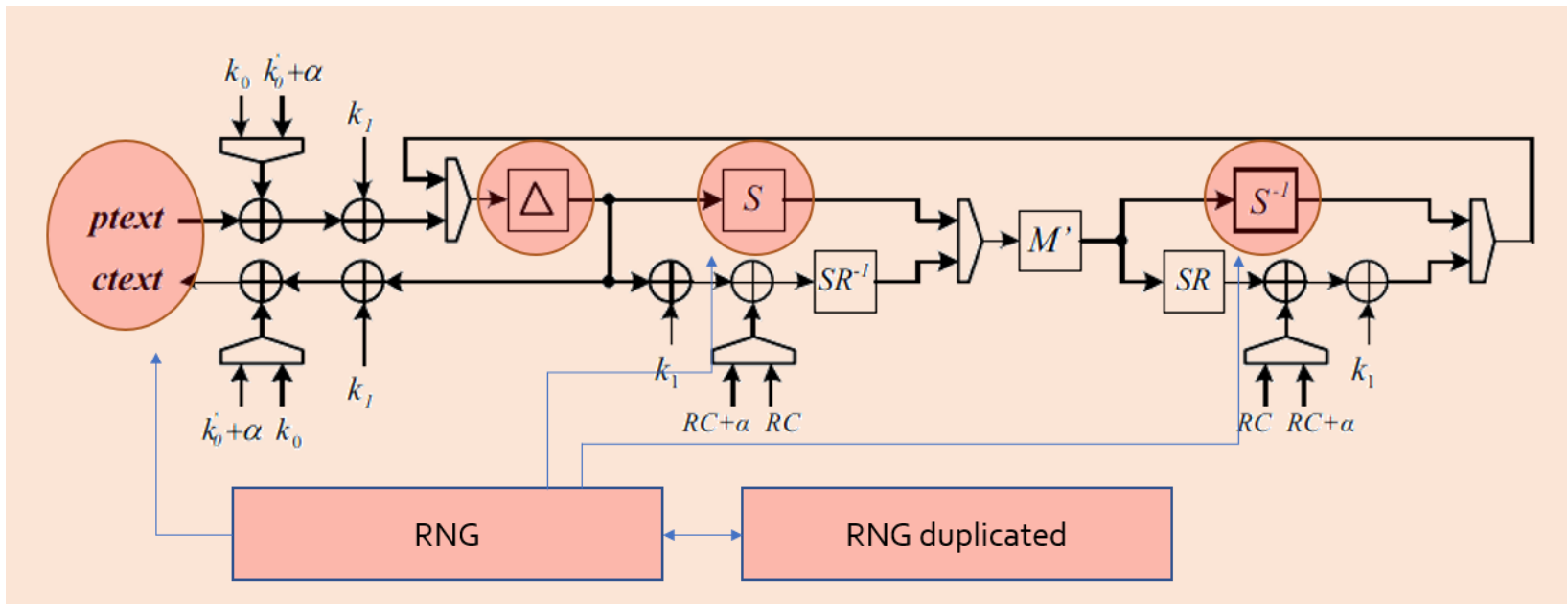
- Only efficiency and functionality not adequate
- Should also be resistant against physical attacks
  - Side-channel attacks (SCA)
  - Fault attacks
- Countermeasures
  - Masking: Threshold implementations, etc.
  - Redundancy



## Countermeasures

- Masking – Novel techniques
  - Threshold Implementations
  - Domain-oriented masking
  - Comes with cost, sharing of secret triples/quadruples costs (even for first-order security)
- Brings additional randomness as well
  - Cost of random number generators – non-linear shift registers (NLFSRs)
- Redundancy against fault attacks
  - For critical parts in the design (NLFSRs)

## PRINCE: When Protected



## SCA-resistant “Threshold Implementation” of PRINCE

Bozilov et al (KU Leuven) at LWC Workshop 2016

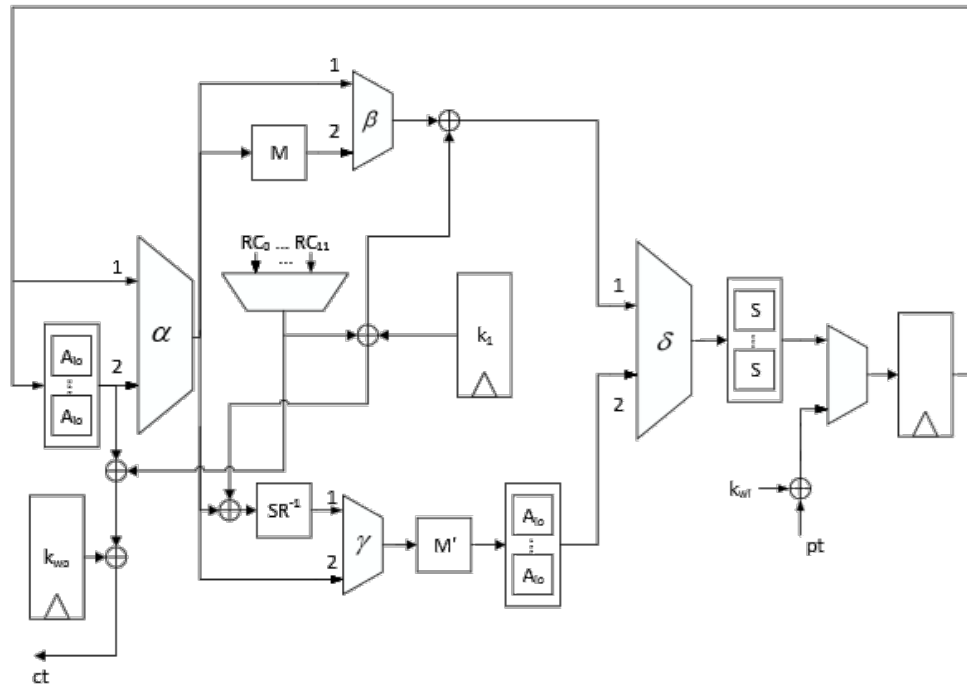
### Threshold Implementations of PRINCE: The Cost of Physical Security

**Abstract.** Threshold implementations have recently emerged as one of the most popular masking countermeasures for hardware implementations of cryptographic primitives. In the original version of TI, the number of input shares was dependant on both security order  $d$  and algebraic degree of a function  $t$ , namely  $td + 1$ . At CRYPTO 2015 Reparaz et al. presented a way to perform  $d$ -th order secure implementation using  $d + 1$  shares. Here we analyze  $d + 1$  and  $td + 1$  TI versions for first and second order secure implementations of the PRINCE block cipher. We compare a plain round-based implementation of PRINCE with its secured versions and we report hardware figures to indicate the overhead introduced by adding a side channel protection.



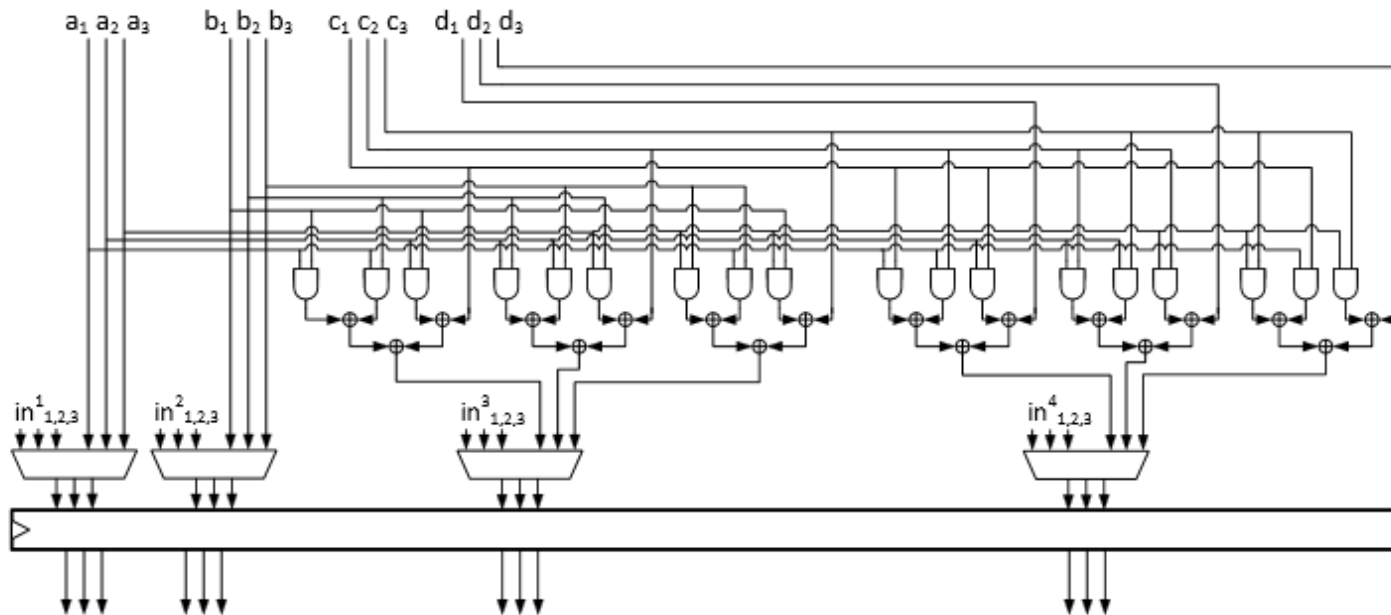
## SCA-resistant “Threshold Implementation” of PRINCE

- Applied on PRINCE Sbox: Algebraic degree 3, Class  $Q_{294}$
- Unprotected, round-based PRINCE



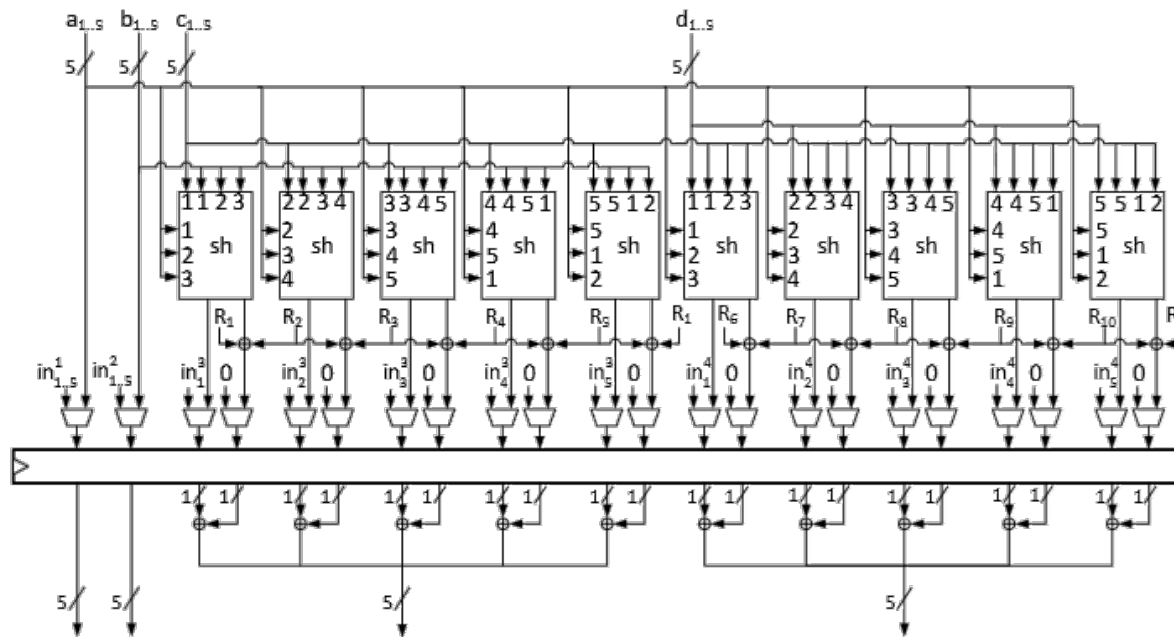
## SCA-resistant “Threshold Implementation” of PRINCE

- Class  $Q_{294}$  sharing, first-order secure, 3 by 3 sharing
- No re-masking, sharing is uniform



## SCA-resistant “Threshold Implementation” of PRINCE

- Class  $Q_{294}$  sharing, second-order secure, 5 by 10 sharing
- Re-masking applied



## SCA-resistant “Threshold Implementation” of PRINCE

### Results

PRINCE-128 (round-based implementation) unprotected

Technology	Area (GE)
ASIC, 90nm	3589

PRINCE-128 (round-based implementation) 1<sup>st</sup>-order secure

Technology	Area (GE)
ASIC, 90nm	11958

PRINCE-128 (round-based implementation) 2<sup>nd</sup>-order secure

Technology	Area (GE)
ASIC, 90nm	21879

- Development of novel resource-efficient ciphers
  - Both SCA and fault attacks in mind
  - With the cost of randomness in mind
    - Needed for countermeasures
  - Still more on low-latency!

**Thanks for listening!**

Any questions?

([elif.kavun@uni-passau.de](mailto:elif.kavun@uni-passau.de))