Faculty of Computer Science and Mathematics

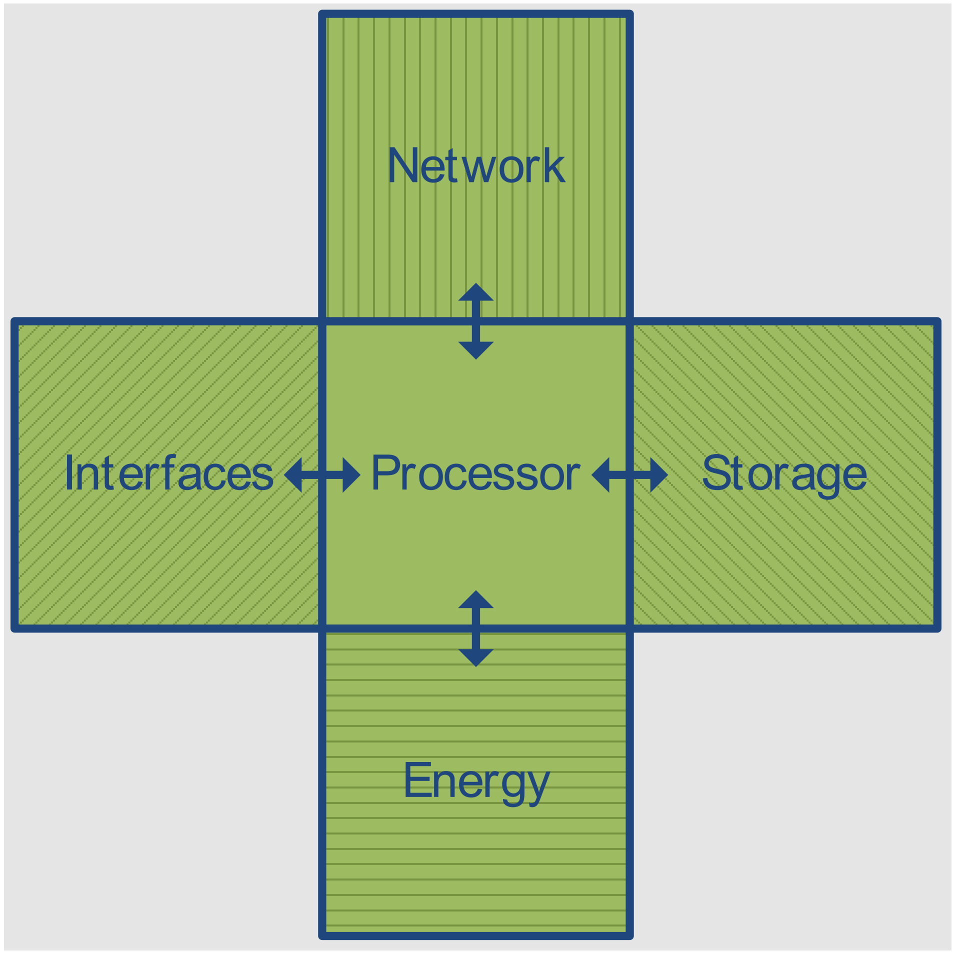
Resource-Efficient Security for Cyber Physical Systems

*internetofbusiness.com

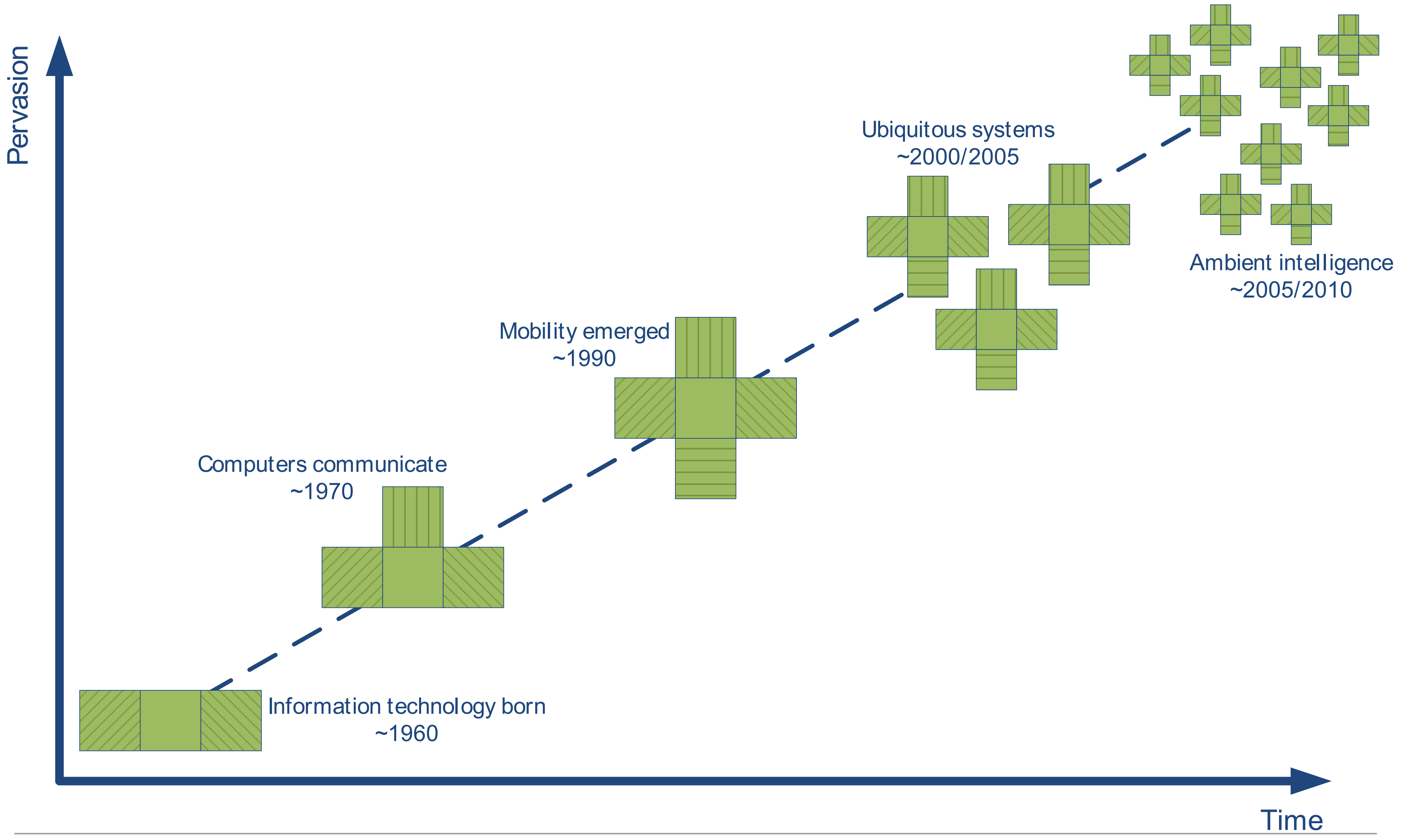
Elif Bilge Kavun

September 21, 2023

Ubiquitous Computing Era



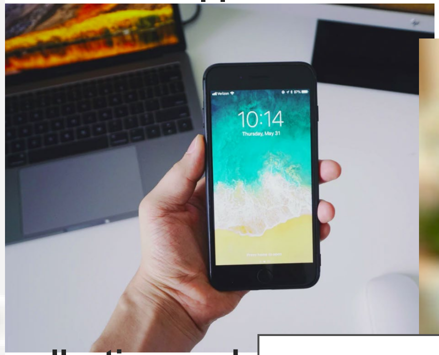
Ubiquitous Computing Era



Ubiquitous Computing Era

Electronic passports

Smartphones
Mobile applications



Smart home

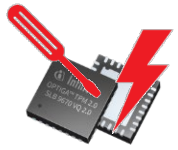
Medical & sensor systems

Payment & toll-collection cards

Products, IPs

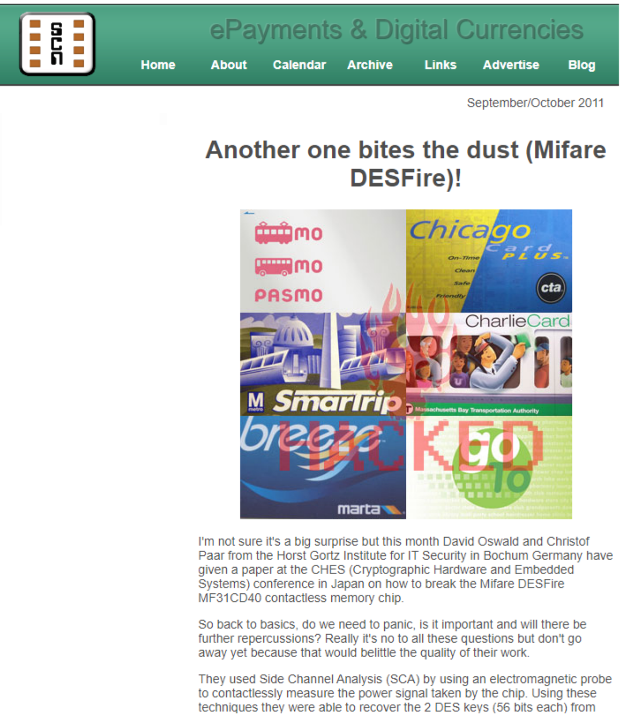
Automation components
Asset tracking systems

Car key systems



Security Concerns!!!

Ubiquitous Computing Era



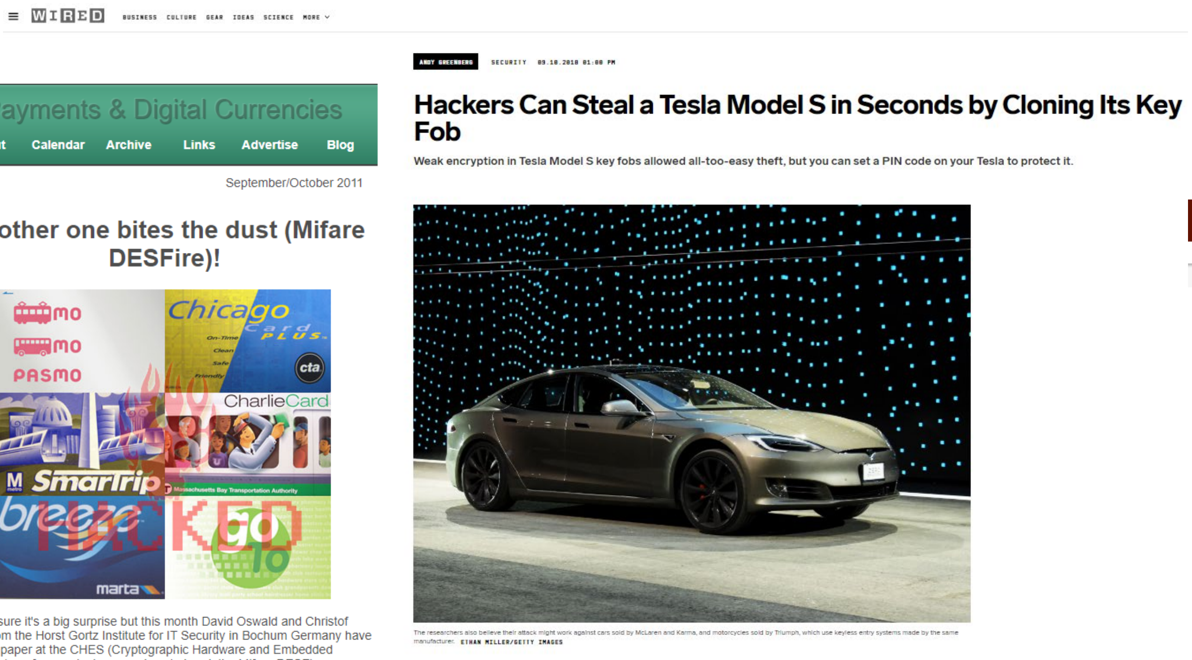
September/October 2011

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 MF31CD40 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

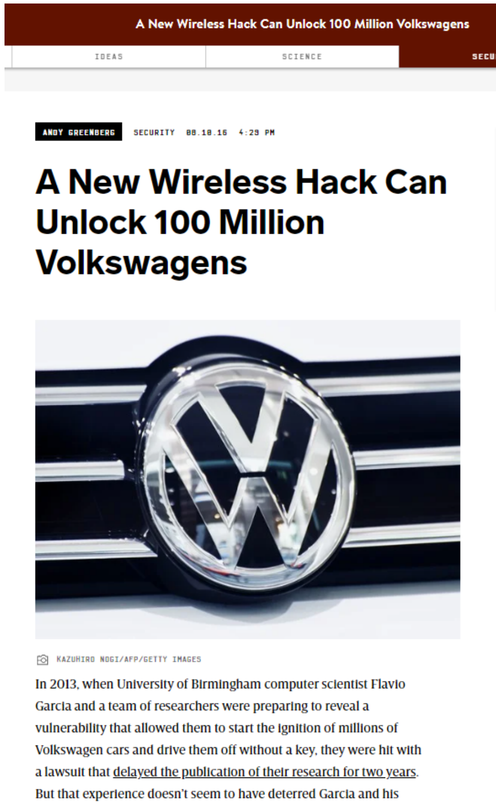


ANDY GREENBERG SECURITY 09.10.2018 01:00 PM

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 believe 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. ETHAN MILLER/GETTY IMAGES



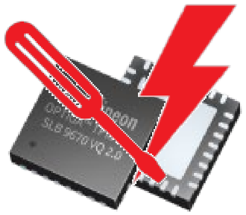
ANDY GREENBERG SECURITY 08.10.16 4:29 PM

A New Wireless Hack Can Unlock 100 Million Volkswagens

KAZUHIRO NOGI/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

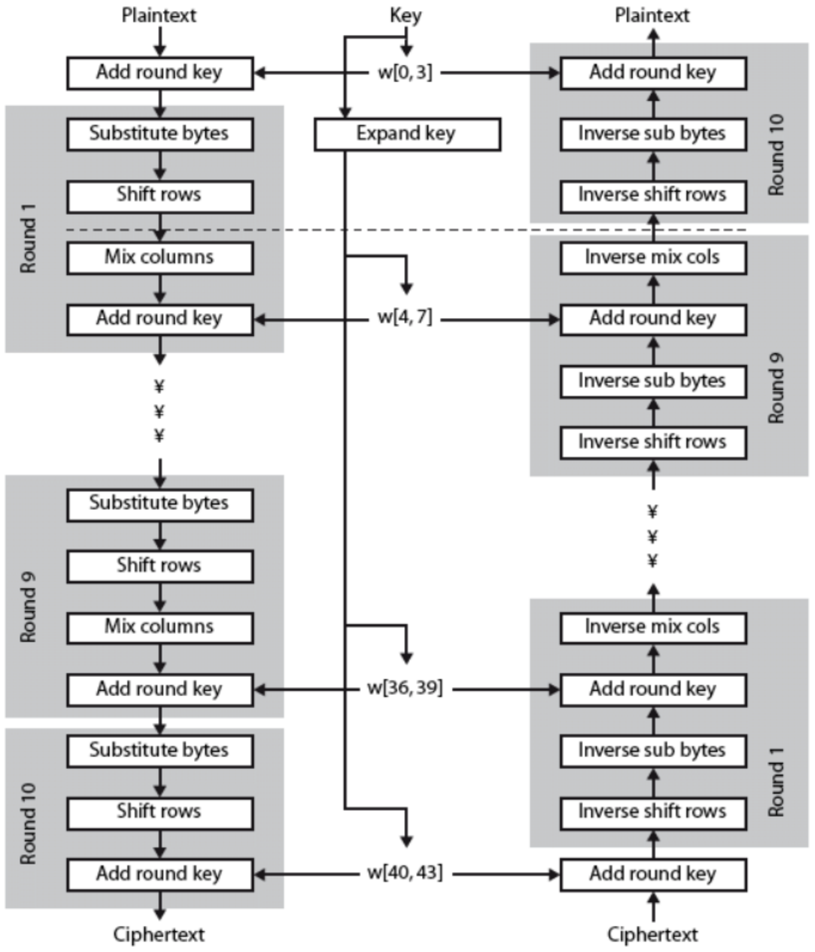
Ubiquitous Computing Era



Good security designs and architecture needed to resist attacks!

- Access control
- Data confidentiality
- Security
- Counterfeiting mitigations

Need for Security: Same Level?

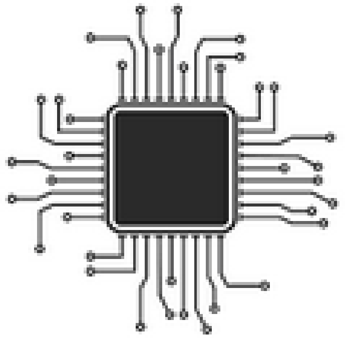


- **Conventional cryptography**
  - RSA
  - Standard block cipher solutions (AES, etc.)
- **Applications in**
  - Servers
  - PCs
  - “Strong” tablets, smartphones

Need for Security: Same Level?

UNIVERSITY OF PASSAU
Faculty of Computer Science and Mathematics

Embedded/IoT devices ➔ **Resource-constrained!**



Chip Area: Limited!

Power and Energy Consumption: Lowest Possible!

➔ to match these constraints:

Need for *Tailored* Cryptography: *Lightweight* Cryptography

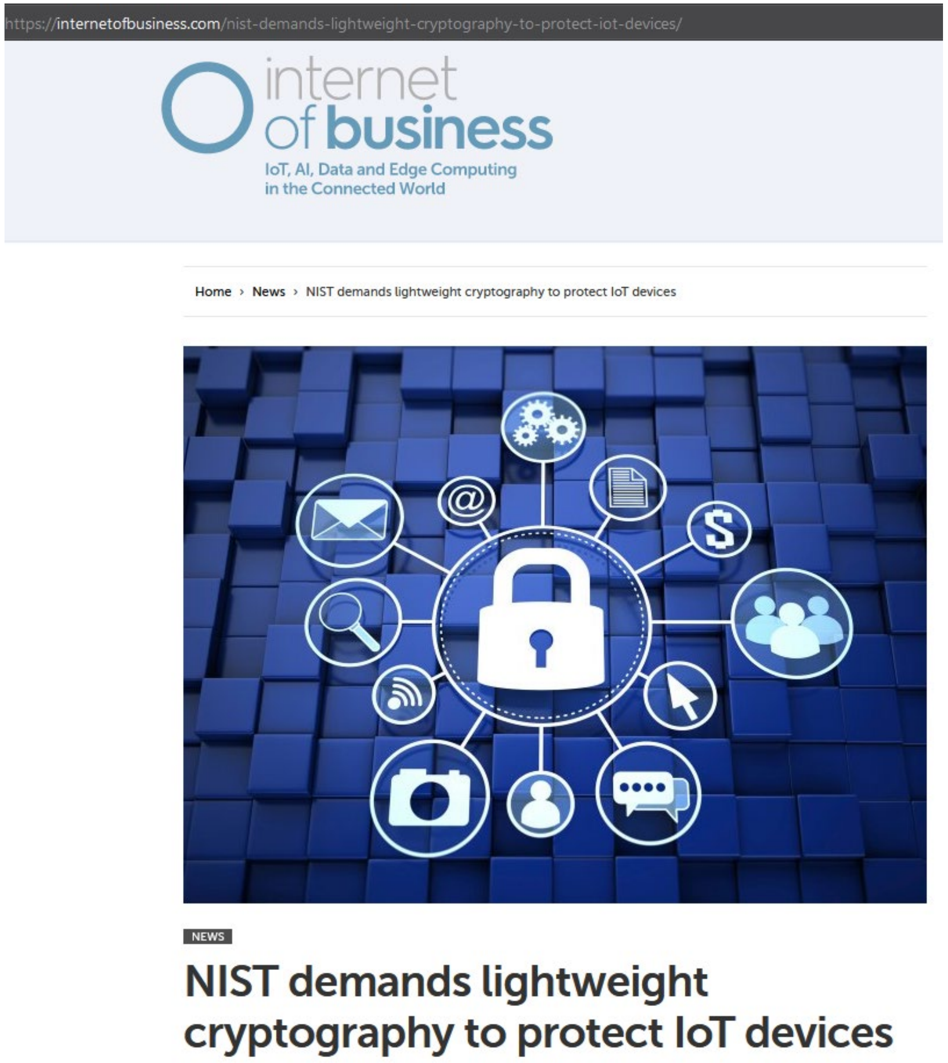
Resource-Efficient!

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.

Resource-Efficient Block Ciphers



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

internet of business

IoT, AI, Data and Edge Computing in the Connected World

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

NEWS

NIST demands lightweight cryptography to protect IoT devices

NIST LWC Standardization

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

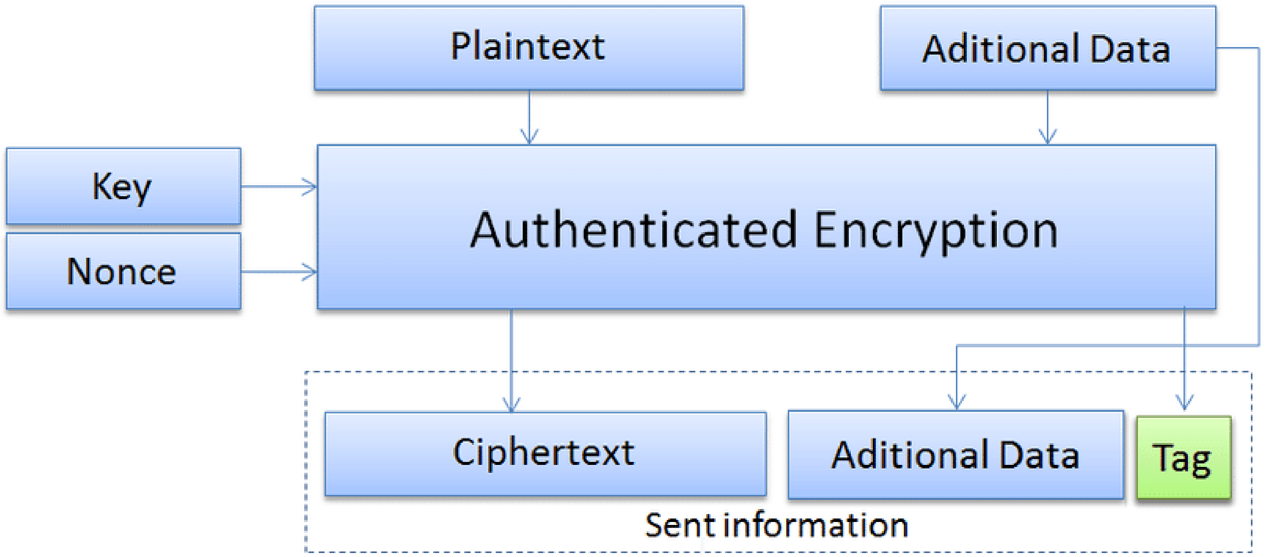


*Figure from "L. Cardoso Santos and J. López, Pipeline Oriented Implementation of NORX for ARM Processors, SBSEG'17"

NIST LWC Standardization

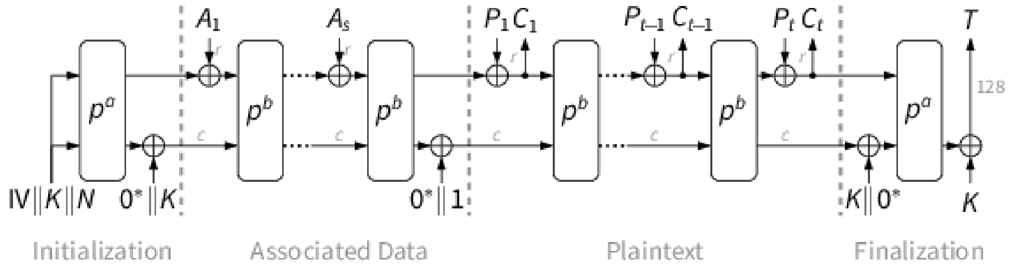
- In round format
- 10 finalists

  ASCON, Elephant, GIFT-COFB, Grain128-AEAD, ISAP, Photon-Beetle , Romulus, Sparkle , TinyJambu , Xoodyak
- ASCON selected
- Implementations on hardware
  - Benchmarking implementations targeting low-cost modern FPGAs: A call launched in 2022
  - FPGA benchmarking available for Round 2 candidates [1]

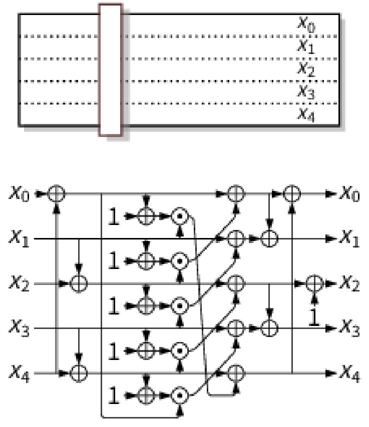
- Mohajerani et al., FPGA Benchmarking of Round 2 Candidates in the NIST Lightweight Cryptography Standardization Process: Methodology, Metrics, Tools, and Results, Cryptology ePrint Archive, Paper 2020/1207, 2020.

ASCON

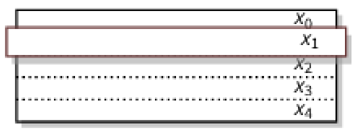
1. **Initialization**: initializes the state with the key K and nonce N.
2. **Associated Data Processing**: updates the state with associated data blocks A_i.
3. **Plaintext Processing**: injects plaintext blocks P_i into the state and extracts ciphertext blocks C_i.
4. **Finalization**: injects the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



The duplex sponge mode for ASCON authenticated encryption



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



$x_0 := x_0 \oplus (x_0 \ggg 19) \oplus (x_0 \ggg 28)$

$x_1 := x_1 \oplus (x_1 \ggg 61) \oplus (x_1 \ggg 39)$

$x_2 := x_2 \oplus (x_2 \ggg 1) \oplus (x_2 \ggg 6)$

$x_3 := x_3 \oplus (x_3 \ggg 10) \oplus (x_3 \ggg 17)$

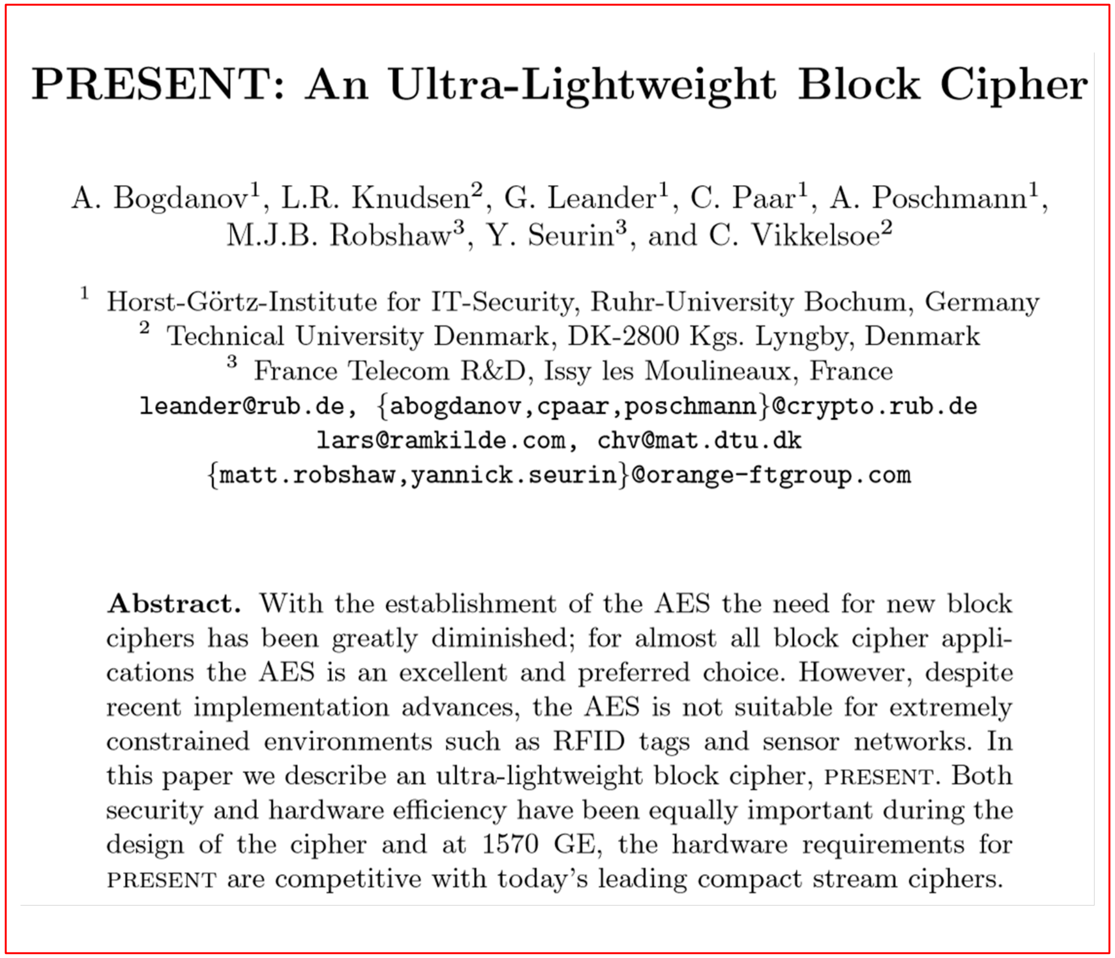
$x_4 := x_4 \oplus (x_4 \ggg 7) \oplus (x_4 \ggg 41)$

ASCON's linear layer

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

Initial Proposals

ISO/IEC 29192-2:2019
Information security — Lightweight cryptography — Part 2: Block ciphers



PRESENT: An Ultra-Lightweight Block Cipher

A. Bogdanov[1], L.R. Knudsen[2], G. Leander[1], C. Paar[1], A. Poschmann[1], M.J.B. Robshaw[3], Y. Seurin[3], and C. Vikkelsoe[2]

[1] Horst-Görtz-Institute for IT-Security, Ruhr-University Bochum, Germany
[2] Technical University Denmark, DK-2800 Kgs. Lyngby, Denmark
[3] 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

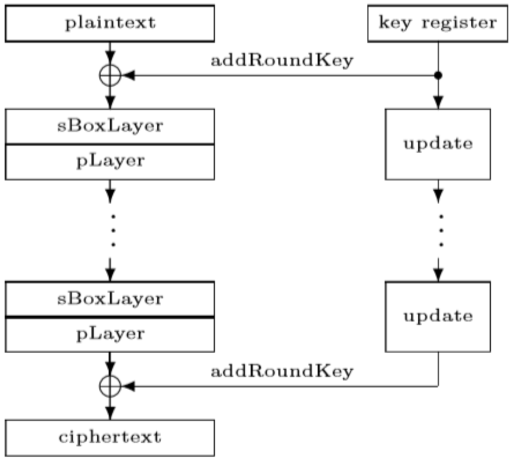
Abstract. With the establishment of the AES the need for new block ciphers has been greatly diminished; for almost all block cipher applications the AES is an excellent and preferred choice. However, despite recent implementation advances, the AES is not suitable for extremely constrained environments 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!

Initial Proposals

ISO/IEC 29192-2:2019
Information security — Lightweight cryptography — Part 2: Block ciphers

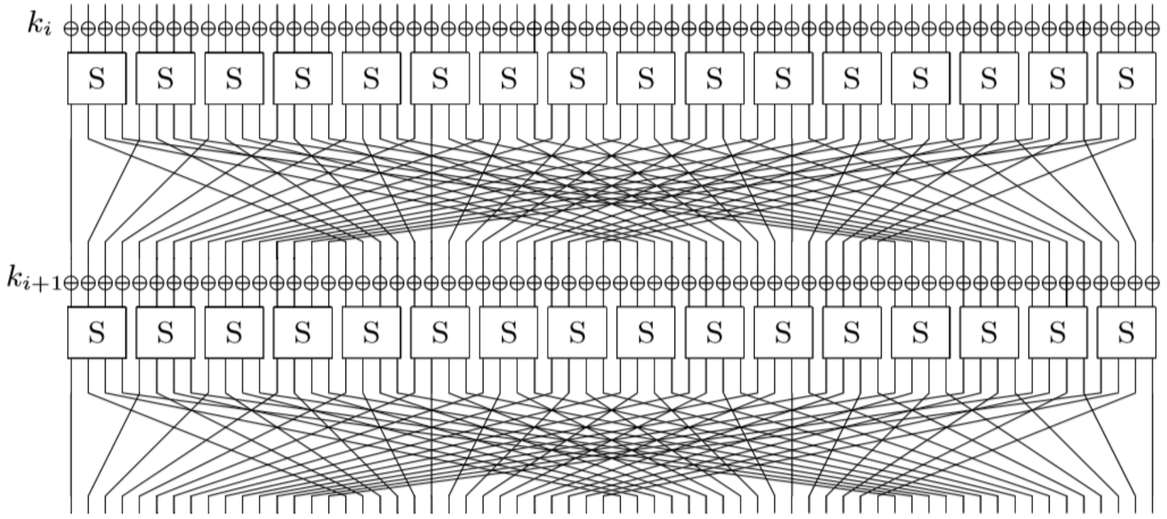
```
generateRoundKeys()
for $i = 1$ to 31 do
    addRoundKey(STATE,$K_i$)
    sBoxLayer(STATE)
    pLayer(STATE)
end for
addRoundKey(STATE,$K_{32}$)
```



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

Initial Proposals

ISO/IEC 29192-2:2019
Information security — Lightweight cryptography — Part 2: Block ciphers



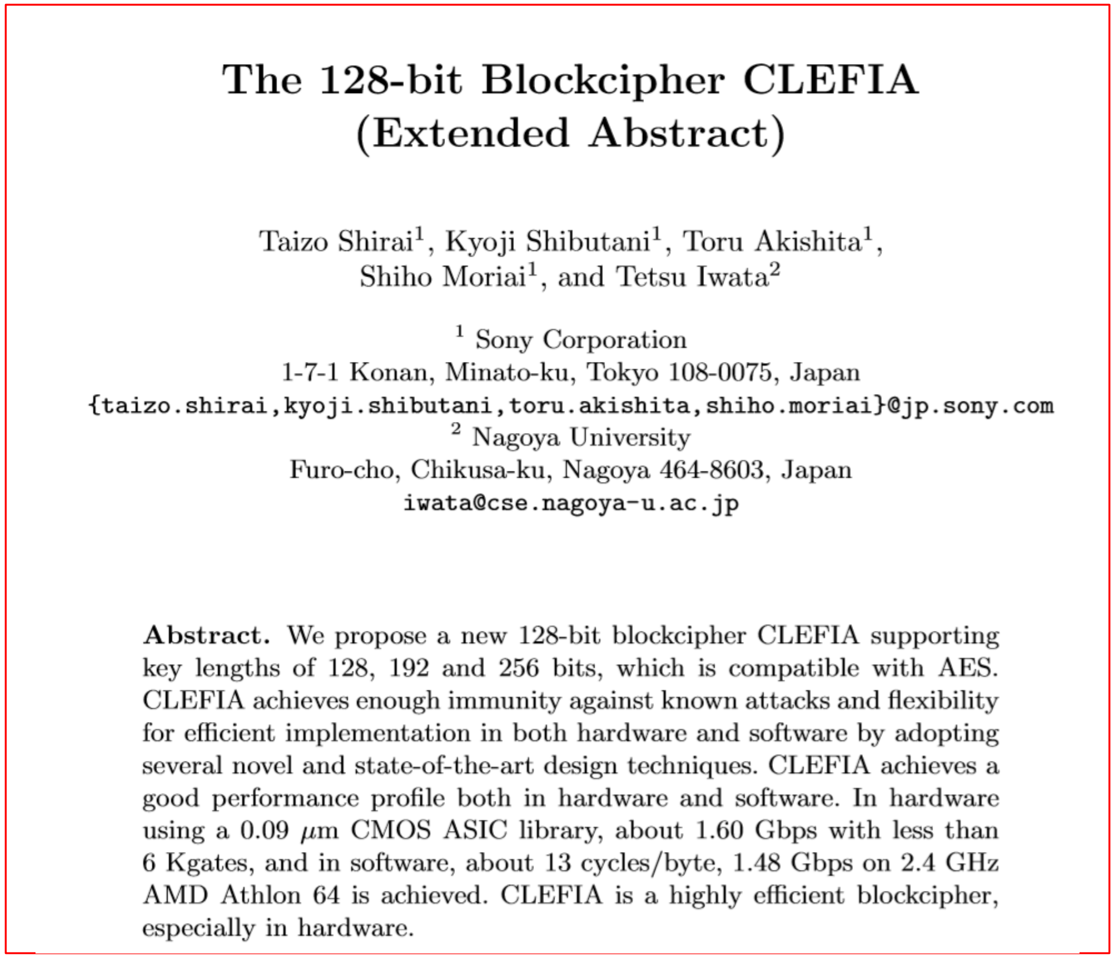
The S/P network for PRESENT.

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

Initial Proposals

UNIVERSITY OF PASSAU
Faculty of Computer Science and Mathematics

ISO/IEC 29192-2:2019
Information security — Lightweight cryptography — Part 2: Block ciphers



The 128-bit Blockcipher CLEFIA (Extended Abstract)

Taizo Shirai[1], Kyoji Shibutani[1], Toru Akishita[1], Shiho Moriai[1], and Tetsu Iwata[2]

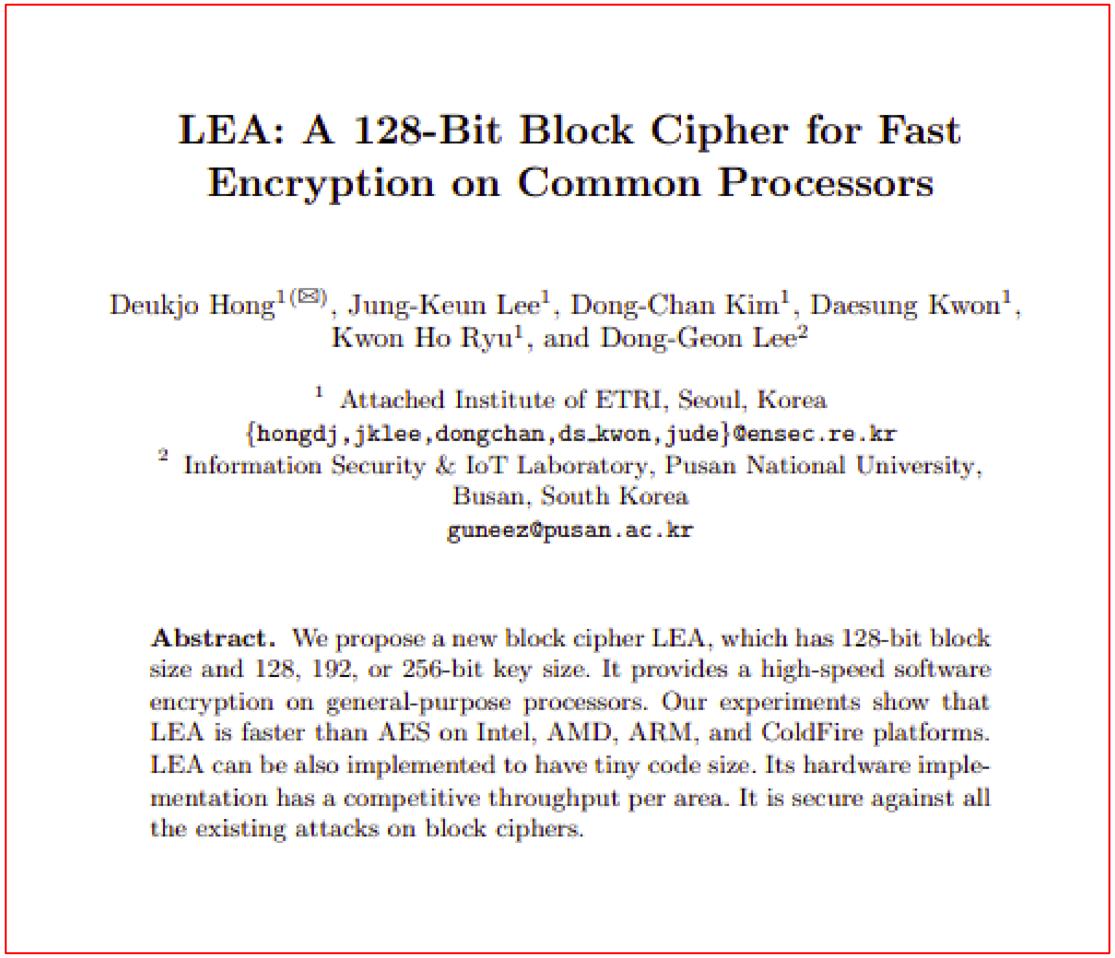
[1] Sony Corporation
1-7-1 Konan, Minato-ku, Tokyo 108-0075, Japan
{taizo.shirai,kyoji.shibutani,toru.akishita,shiho.moriai}@jp.sony.com
[2] Nagoya University
Furo-cho, Chikusa-ku, Nagoya 464-8603, Japan
iwata@cse.nagoya-u.ac.jp

Abstract. We propose a new 128-bit blockcipher CLEFIA supporting key lengths of 128, 192 and 256 bits, which is compatible with AES. CLEFIA achieves enough immunity against known attacks and flexibility for efficient implementation in both hardware and software by adopting several novel and state-of-the-art design techniques. CLEFIA achieves a good performance profile both in hardware and software. In hardware using a 0.09 μm CMOS ASIC library, about 1.60 Gbps with less than 6 Kgates, and in software, about 13 cycles/byte, 1.48 Gbps on 2.4 GHz AMD Athlon 64 is achieved. CLEFIA is a highly efficient blockcipher, especially in hardware.

- By SONY
- 128-bit key minimum
- Feistel network
- Balancing security, speed, cost
- Several Sboxes

Initial Proposals

ISO/IEC 29192-2:2019
Information security — Lightweight cryptography — Part 2: Block ciphers



LEA: A 128-Bit Block Cipher for Fast Encryption on Common Processors

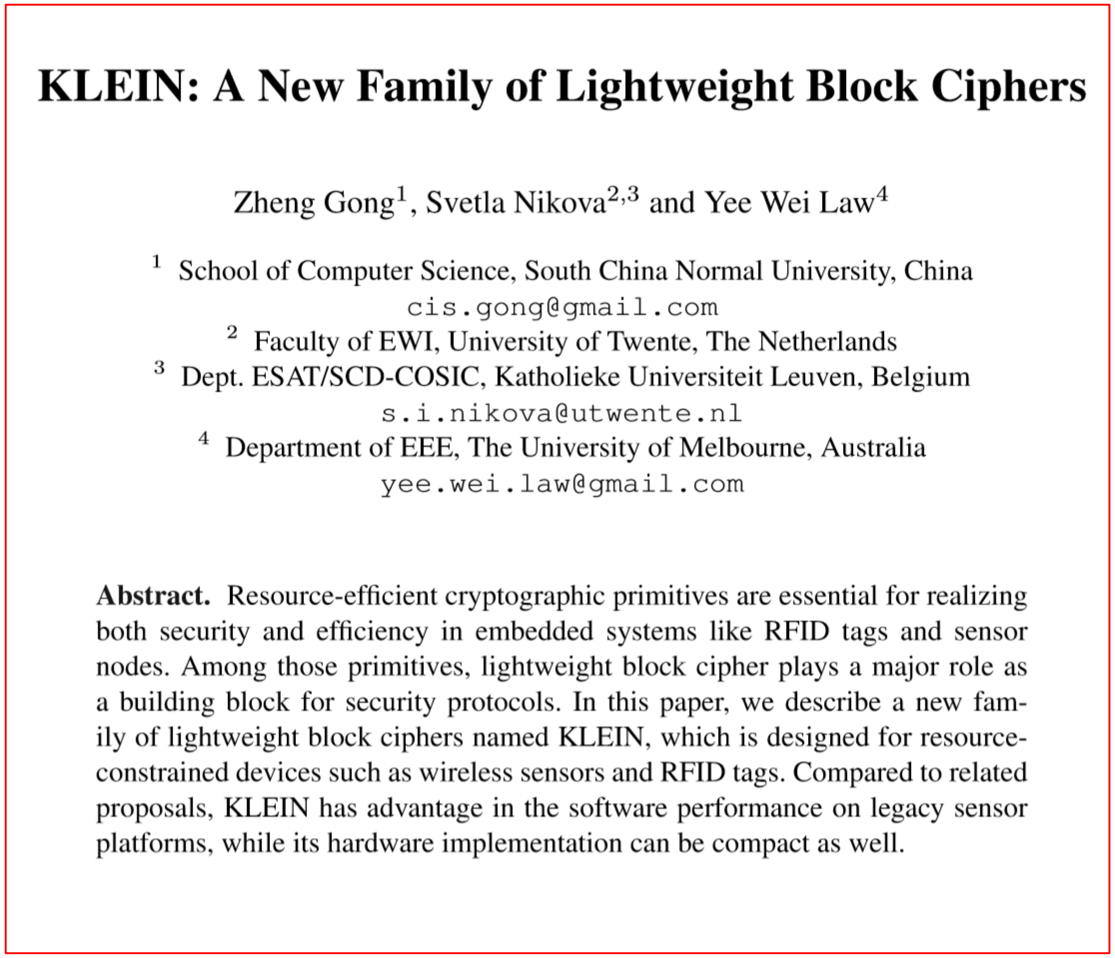
Deukjo Hong[1(✉)], Jung-Keun Lee[1], Dong-Chan Kim[1], Daesung Kwon[1], Kwon Ho Ryu[1], and Dong-Geon Lee[2]

[1] Attached Institute of ETRI, Seoul, Korea
{hongdj,jklee,dongchan,ds_kwon,jude}@ensec.re.kr
[2] Information Security & IoT Laboratory, Pusan National University, Busan, South Korea
guneez@pusan.ac.kr

Abstract. We propose a new block cipher LEA, which has 128-bit block size and 128, 192, or 256-bit key size. It provides a high-speed software encryption on general-purpose processors. Our experiments show that LEA is faster than AES on Intel, AMD, ARM, and ColdFire platforms. LEA can be also implemented to have tiny code size. Its hardware implementation has a competitive throughput per area. It is secure against all the existing attacks on block ciphers.

- 128-bit key minimum
- ARX
  - modular Addition, bitwise Rotation, and bitwise XOR
- Faster than AES in software
  - ~1.5-2 times

Initial Proposals



KLEIN: A New Family of Lightweight Block Ciphers

Zheng Gong[1], Svetla Nikova[2,3] and Yee Wei Law[4]

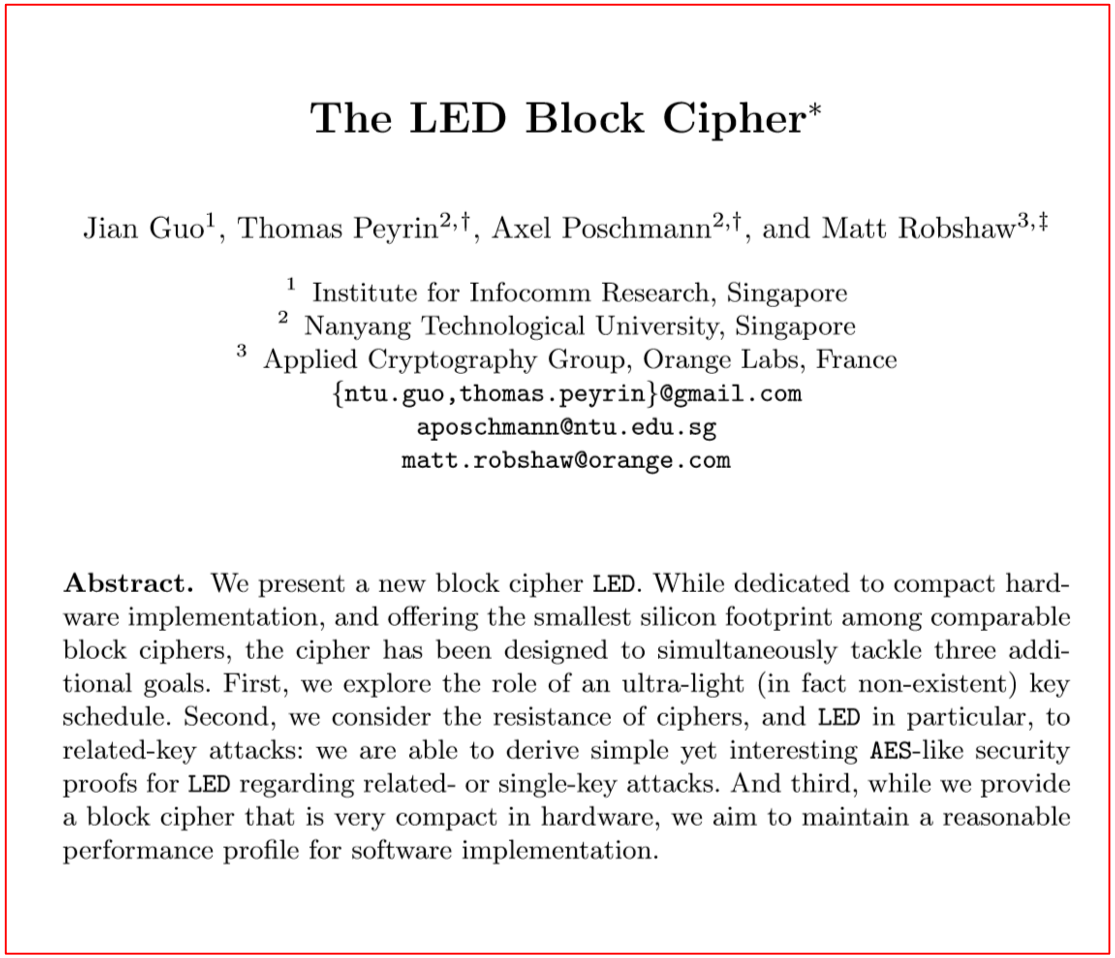
[1] School of Computer Science, South China Normal University, China
cis.gong@gmail.com
[2] Faculty of EWI, University of Twente, The Netherlands
[3] Dept. ESAT/SCD-COSIC, Katholieke Universiteit Leuven, Belgium
s.i.nikova@utwente.nl
[4] Department of EEE, The University of Melbourne, Australia
yee.wei.law@gmail.com

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

```
$sk^1 \leftarrow$ KEY;
STATE $\leftarrow$ PLAINTEXT;
for $i = 1$ to $N_R$ do
  $AddRoundKey$(STATE, $sk^i$);
  $SubNibbles$(STATE);
  $RotateNibbles$(STATE);
  $MixNibbles$(STATE);
  $sk^{i+1} = KeySchedule(sk^i, i)$;
end for
CIPHERTEXT $\leftarrow AddRoundKey$(STATE, $sk^{N_R+1}$);
```

Initial Proposals

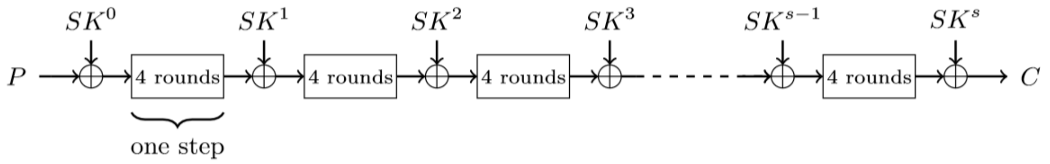

The LED Block Cipher*

Jian Guo[1], Thomas Peyrin[2,†], Axel Poschmann[2,†], and Matt Robshaw[3,‡]

[1] Institute for Infocomm Research, Singapore
[2] Nanyang Technological University, Singapore
[3] Applied Cryptography Group, Orange Labs, France
{ntu.guo,thomas.peyrin}@gmail.com
aposchmann@ntu.edu.sg
matt.robshaw@orange.com

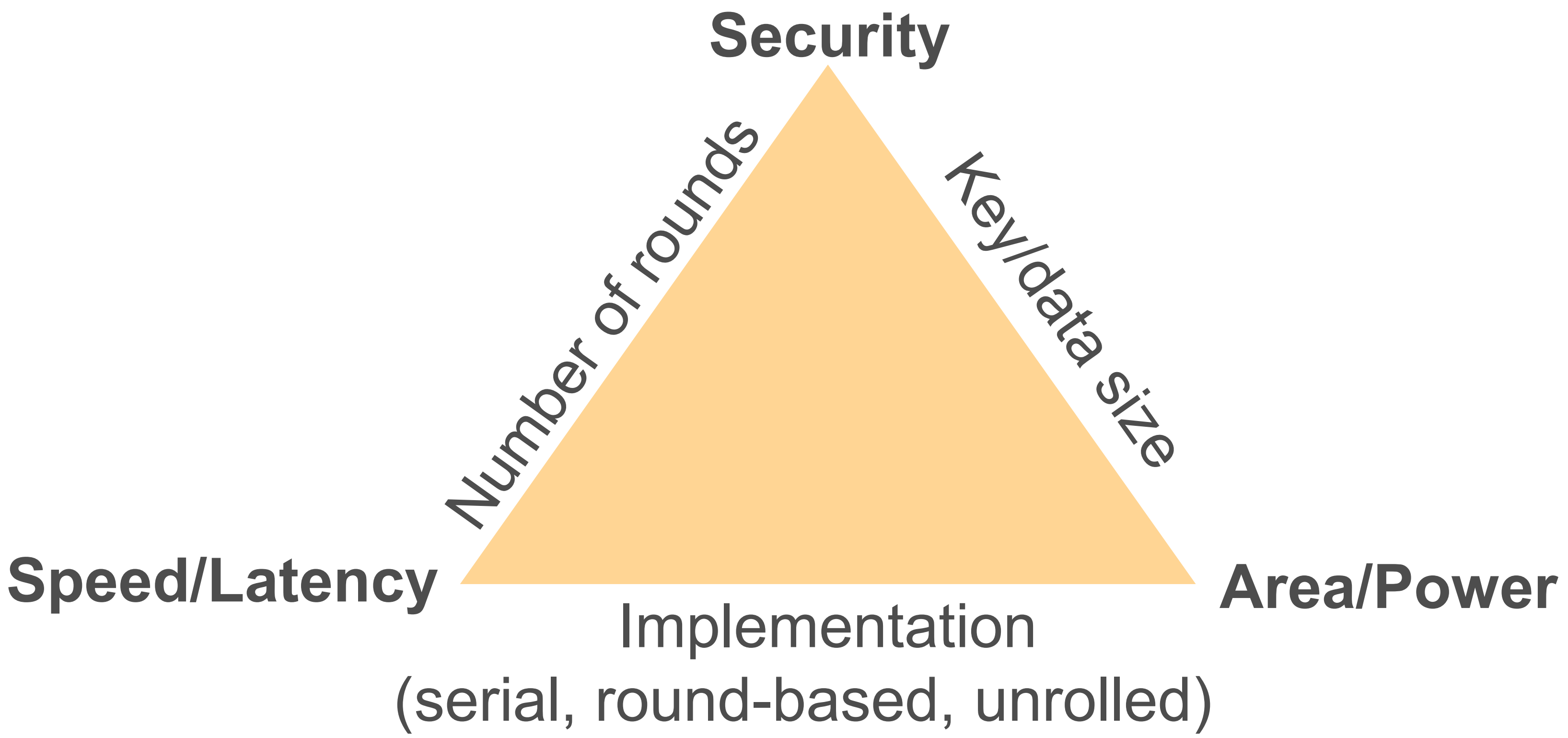
Abstract. We present a new block cipher LED. While dedicated to compact hardware implementation, and offering the smallest silicon footprint among comparable block ciphers, the cipher has been designed to simultaneously tackle three additional goals. First, we explore the role of an ultra-light (in fact 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



Proposals vs. Metrics

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



Hardware Metrics (*)

- Area
  - Usually measured in μm2, 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 μm2 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

Hardware Metrics (*)

UNIVERSITY OF PASSAU
Faculty of Computer Science and Mathematics

- 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 µW)
  - Power estimations on transistor level are more accurate (more steps in design flow)
- Energy
  - Power consumption over a certain time period, i. e., multiplicating the power consumption with the required time (typically in µJ)

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

Implementation Architectures (*)

- 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

Implementation Architectures (*)

UNIVERSITY OF PASSAU
Faculty of Computer Science and Mathematics

- 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

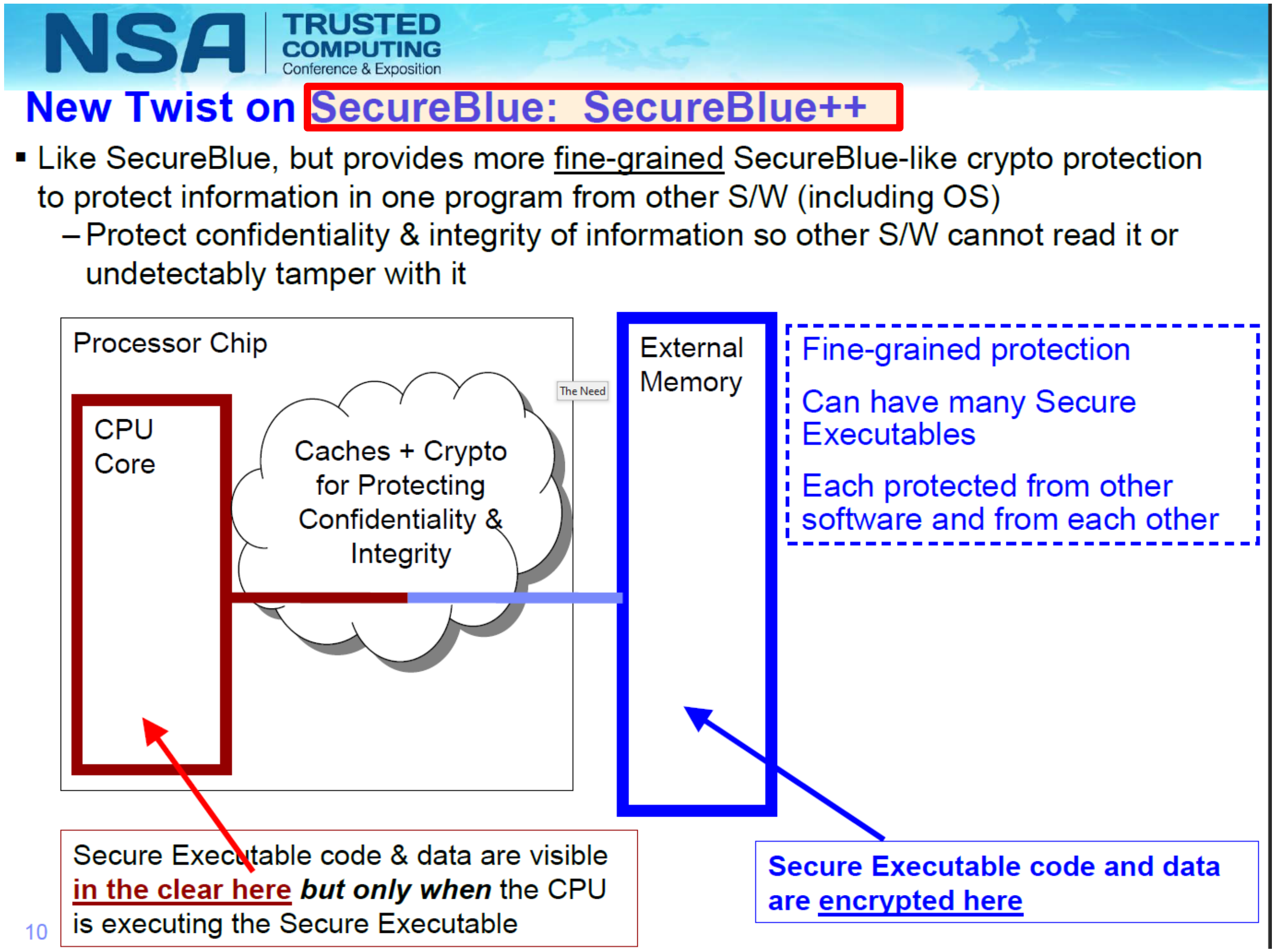
* Shahram Rasoolzadeh, Hardware-Oriented SPN Block Ciphers, PhD Thesis, RUB, 2020

Proposals vs. Metrics

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

Low-latency: The Need

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

Fine-grained protection

Can have many Secure Executables

Each protected from other software and from each other

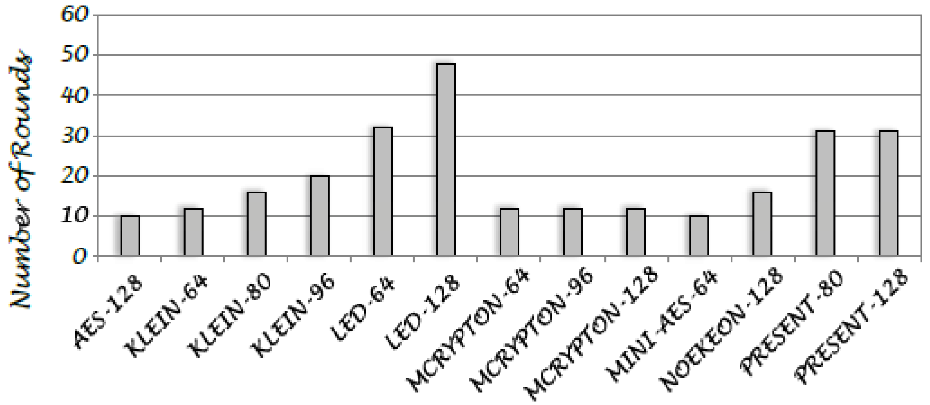
Secure Executable code & data are visible in the clear here ***but only when*** the CPU is executing the Secure Executable

Secure Executable code and data are encrypted here

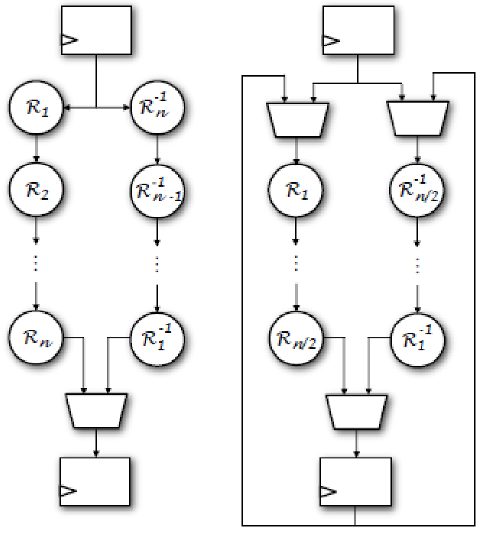
10

Also,
- Intel SGX
- AMD SEV

Low-latency in Focus

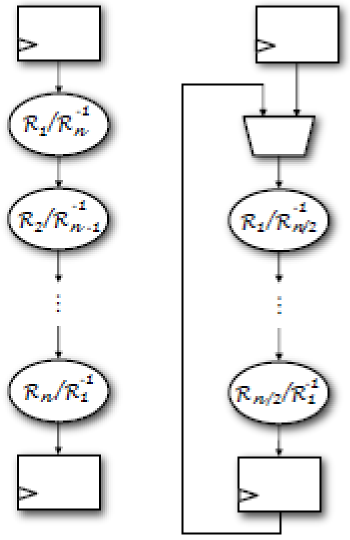


Unrolled



Enc/Dec

1-cycle 2 cycles



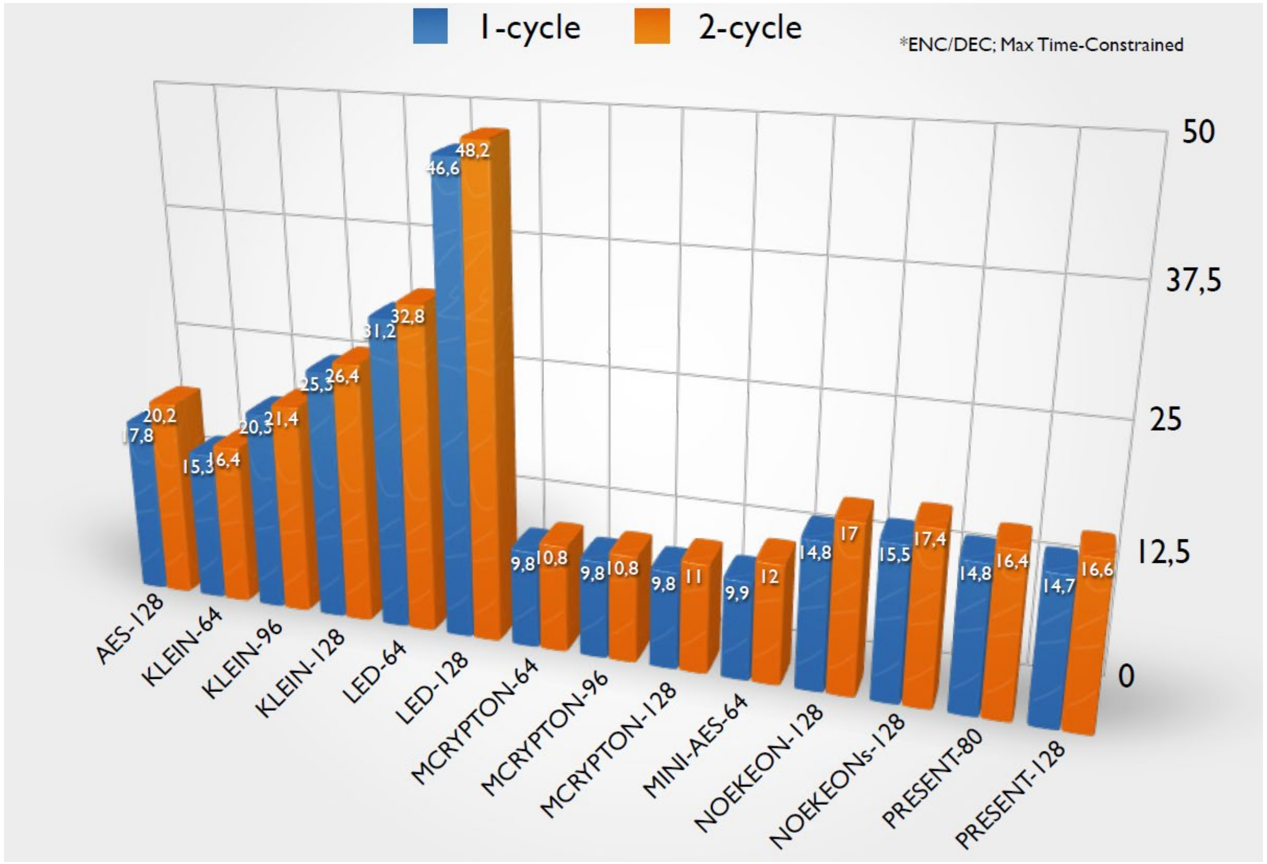
Enc/Dec
Shared Datapath

1-cycle 2 cycles

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

Low-latency in Focus

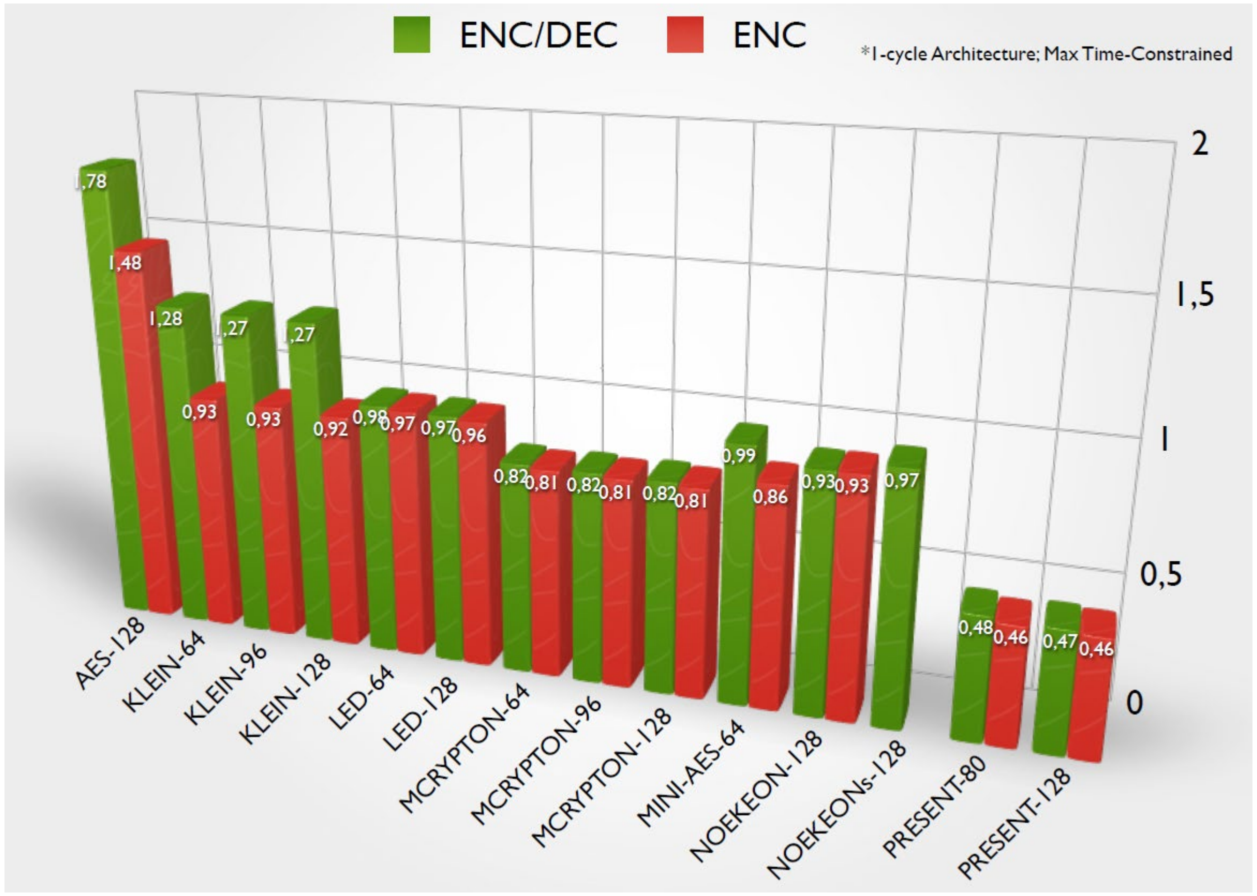
Latency: Time to encrypt one block of data (ns)



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

Low-latency in Focus

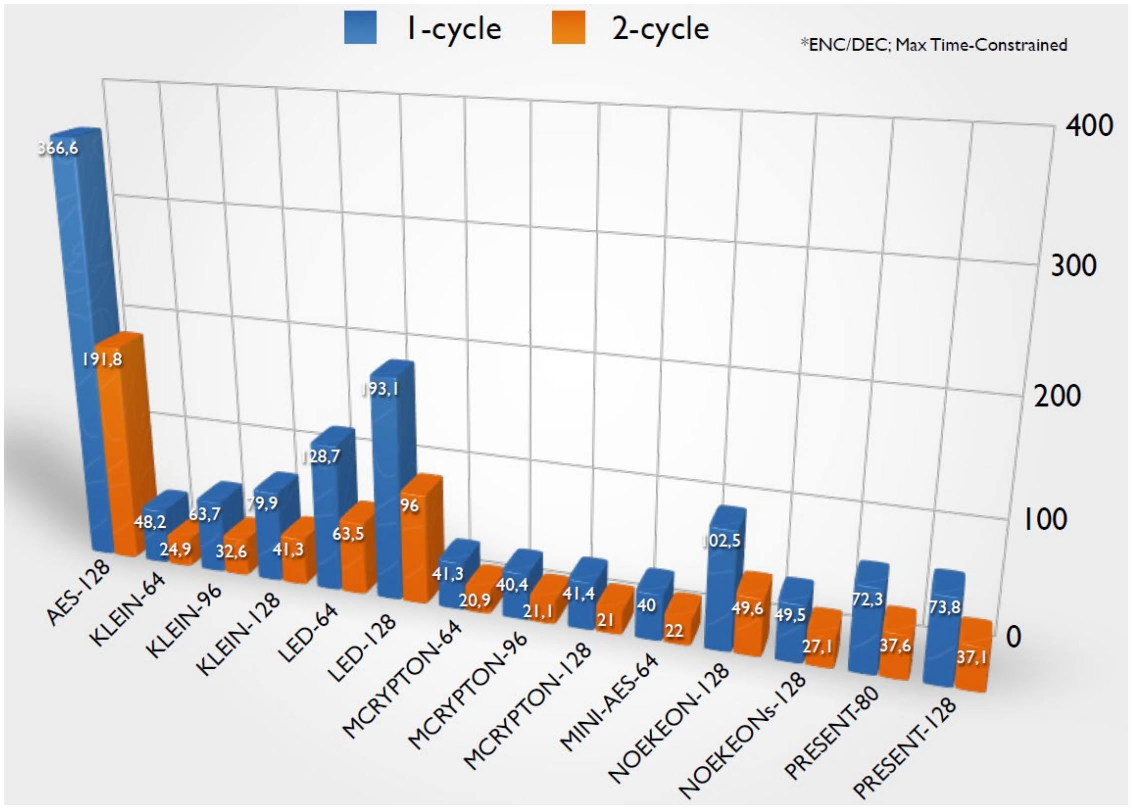
Latency: Time to encrypt one block of data (ns) → per round



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

Low-latency in Focus

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

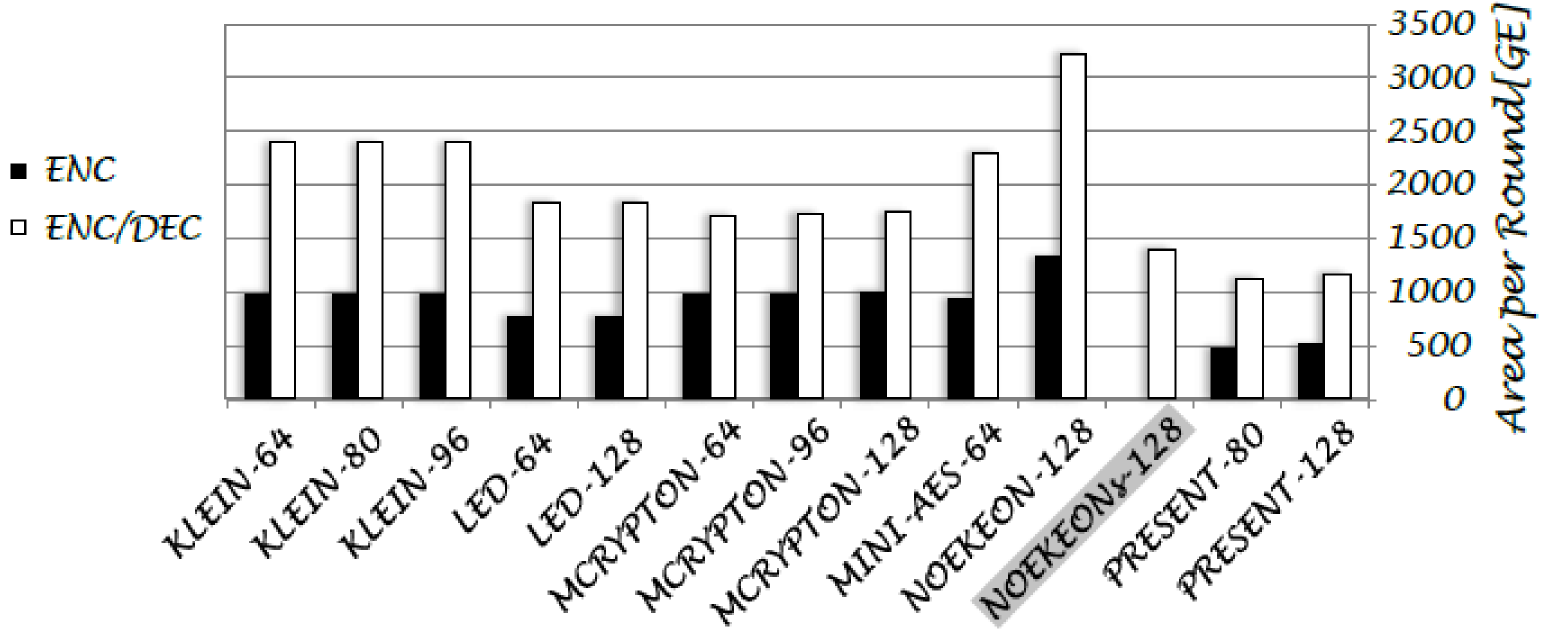


Less area possible for encryption of one block of data?

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

Low-latency in Focus

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



Less area possible for encryption of one block of data?

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

Low-latency in Focus: What is Important? (1/2)

- 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

Low-latency in Focus: What is Important? (2/2)

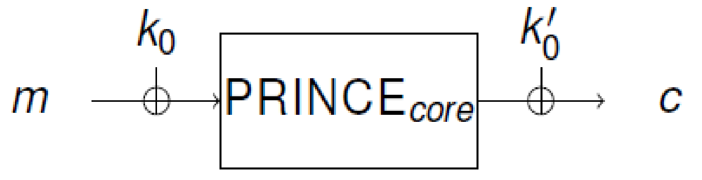
- Slightly heavy round with less/balanced number of rounds
- Simpler key schedule
  - Should be independent of number of rounds
  - Constant addition instead of key schedule should be preferred, if possible
- Minimum overhead for encryption and decryption
  - Use involution

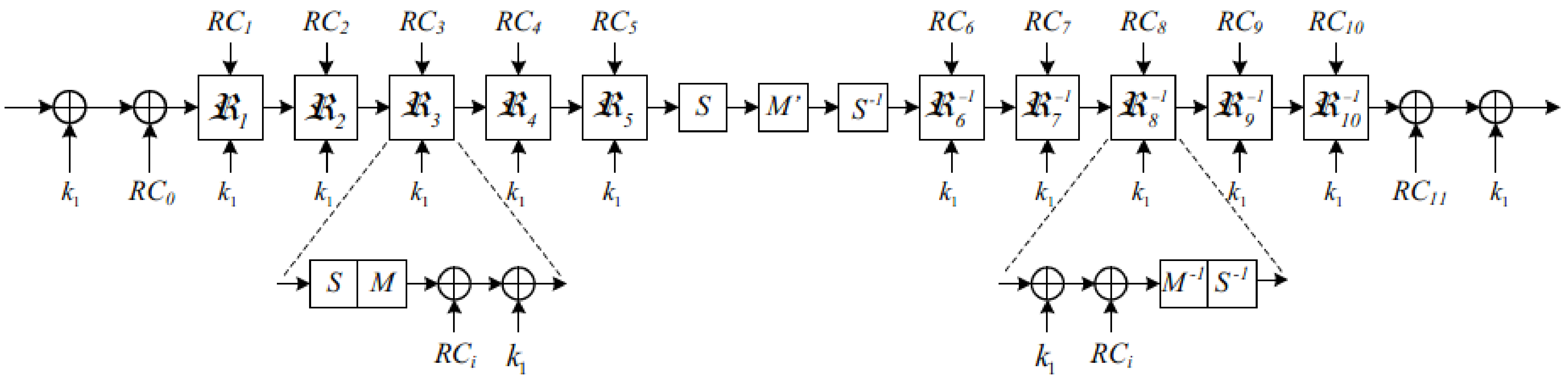
Proposals vs. Metrics

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

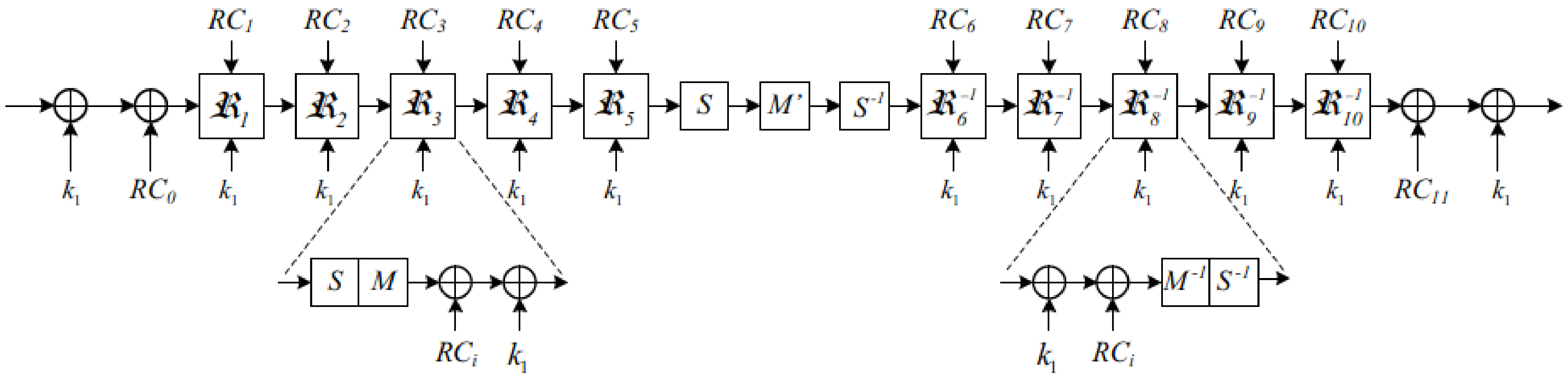
Low-latency: PRINCE Cipher

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





Low-latency: PRINCE Core



RC_1 RC_2 RC_3 RC_4 RC_5 RC_6 RC_7 RC_8 RC_9 RC_{10}
$\mathcal{R}_1$ $\mathcal{R}_2$ $\mathcal{R}_3$ $\mathcal{R}_4$ $\mathcal{R}_5$
S M' S^{-1}
$\mathcal{R}_6^{-1}$ $\mathcal{R}_7^{-1}$ $\mathcal{R}_8^{-1}$ $\mathcal{R}_9^{-1}$ $\mathcal{R}_{10}^{-1}$
k_1 RC_0 k_1 k_1 k_1 k_1 k_1 k_1 k_1 k_1 k_1 k_1 RC_{11} k_1
S M RC_i k_1
k_1 RC_i M^{-1} S^{-1}

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

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

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

Low-latency: PRINCE Core Steps

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

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

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

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

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

M_i is the 4×4 identity matrix

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

MC-layer multiplies the state with M'

involution

Low-latency: PRINCE Core Steps

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

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

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

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

$\rightarrow \alpha = \mathtt{c0ac29b7c97c50dd},$

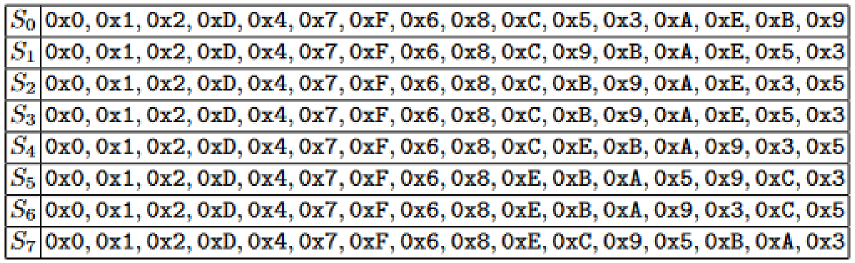
$$(k_0 || k_1) \mapsto (k_0 || P(k_0) || k_1)$$

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

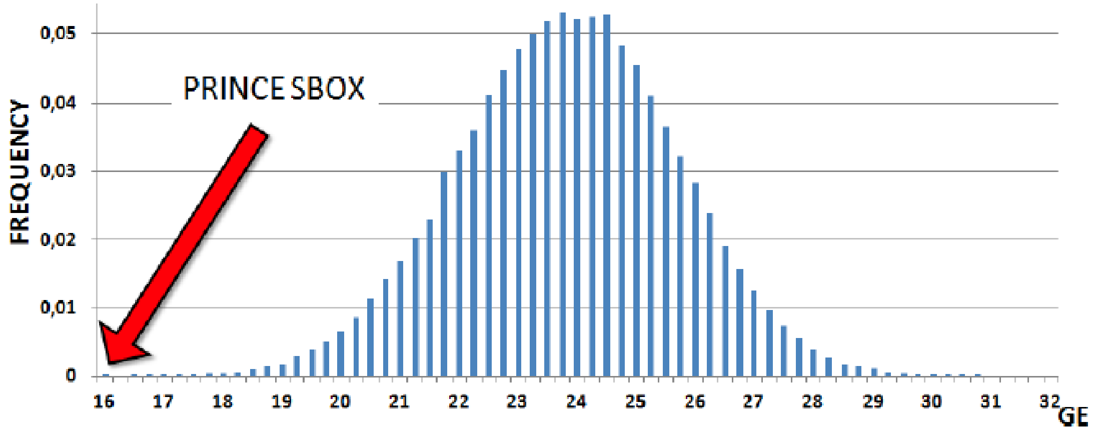
Low-latency: PRINCE Sbox

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

- Smallest Sbox selected



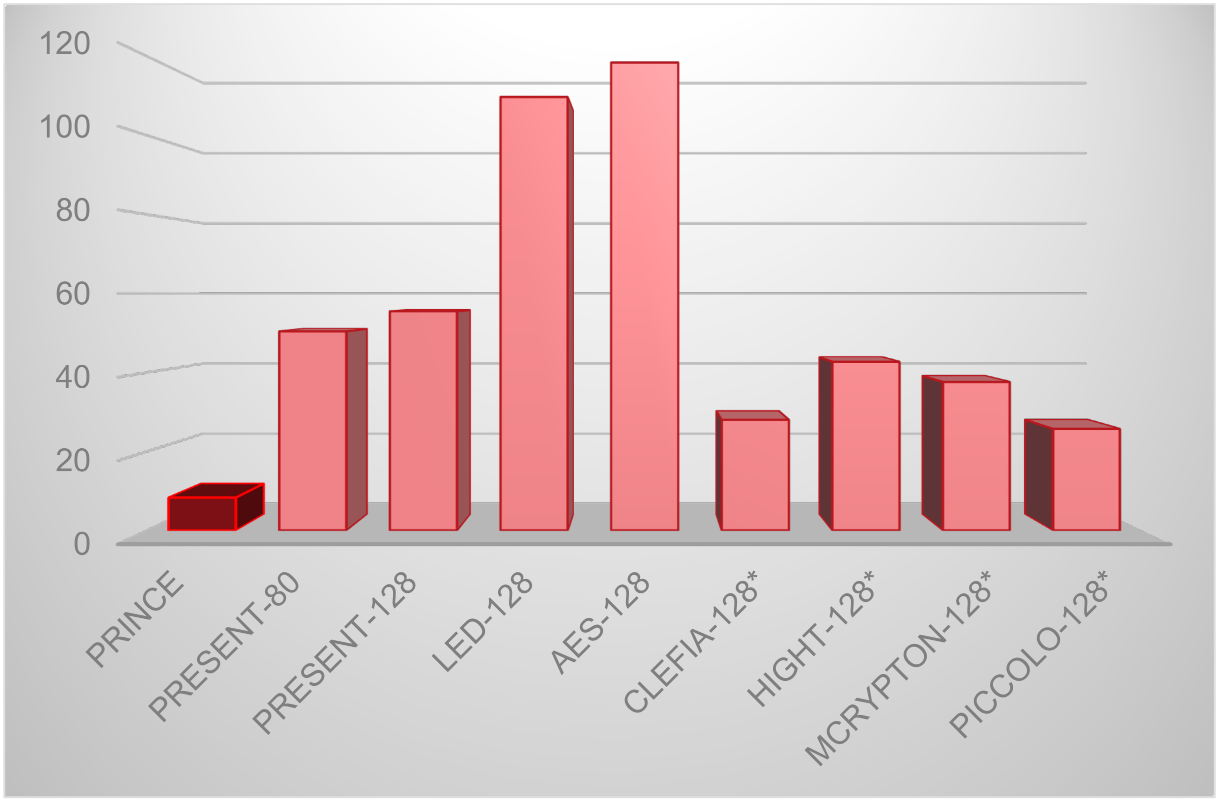
S_0	0x0, 0x1, 0x2, 0xD, 0x4, 0x7, 0xF, 0x6, 0x8, 0xC, 0x5, 0x3, 0xA, 0xE, 0xB, 0x9
S_1	0x0, 0x1, 0x2, 0xD, 0x4, 0x7, 0xF, 0x6, 0x8, 0xC, 0x9, 0xB, 0xA, 0xE, 0x5, 0x3
S_2	0x0, 0x1, 0x2, 0xD, 0x4, 0x7, 0xF, 0x6, 0x8, 0xC, 0xB, 0x9, 0xA, 0xE, 0x3, 0x5
S_3	0x0, 0x1, 0x2, 0xD, 0x4, 0x7, 0xF, 0x6, 0x8, 0xC, 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)

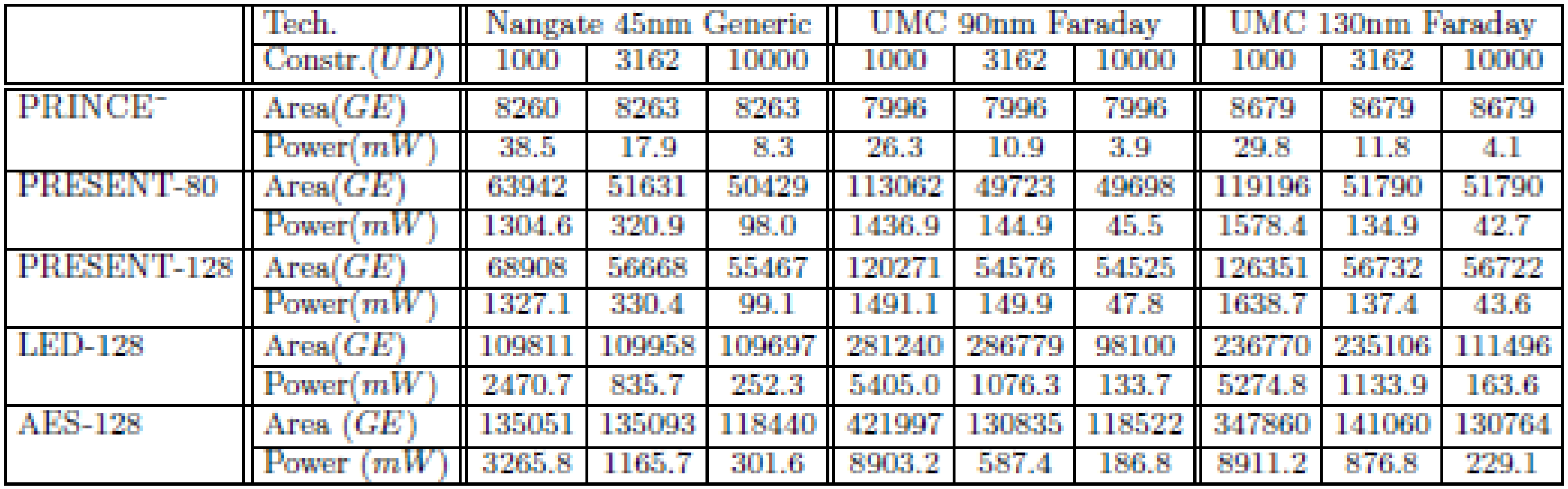
Low-latency: PRINCE Cipher

Results



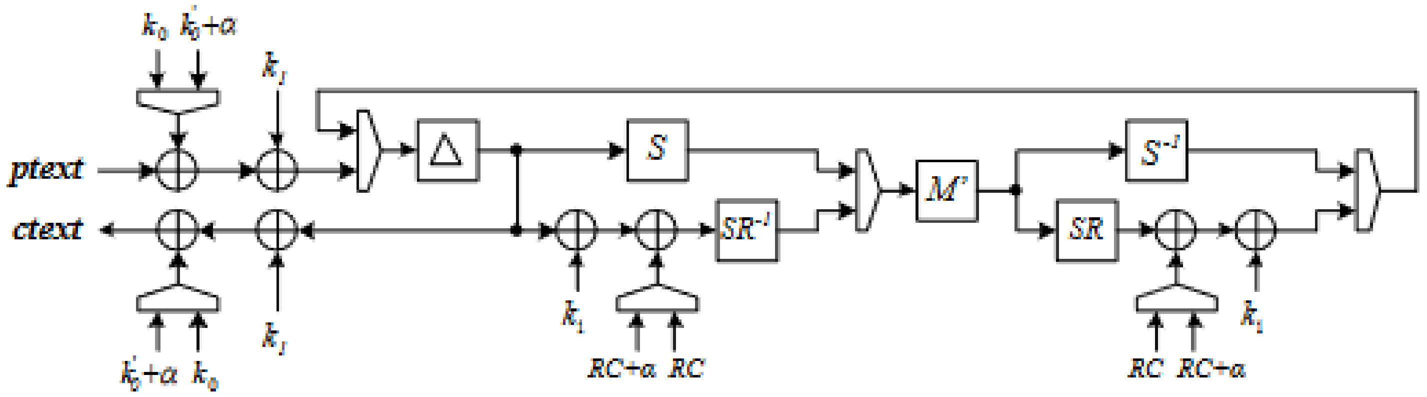
Low-latency: PRINCE Cipher

Results



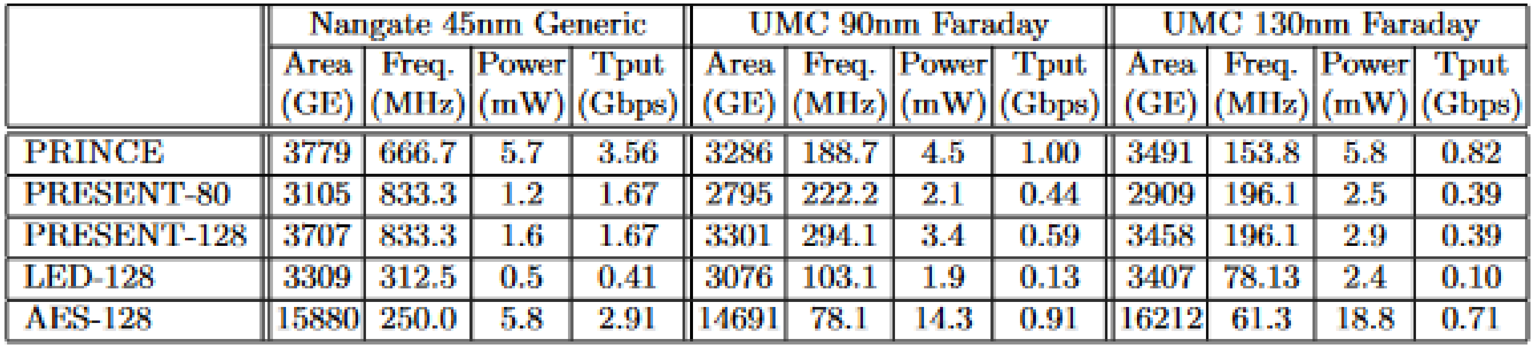
	Tech.	Nangate 45nm Generic			UMC 90nm Faraday			UMC 130nm Faraday		
	Constr.(UD)	1000	3162	10000	1000	3162	10000	1000	3162	10000
PRINCE	Area(GE)	8260	8263	8263	7996	7996	7996	8679	8679	8679
	Power(mW)	38.5	17.9	8.3	26.3	10.9	3.9	29.8	11.8	4.1
PRESENT-80	Area(GE)	63942	51631	50429	113062	49723	49698	119196	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)



Low-latency: PRINCE Cipher (Round-based)

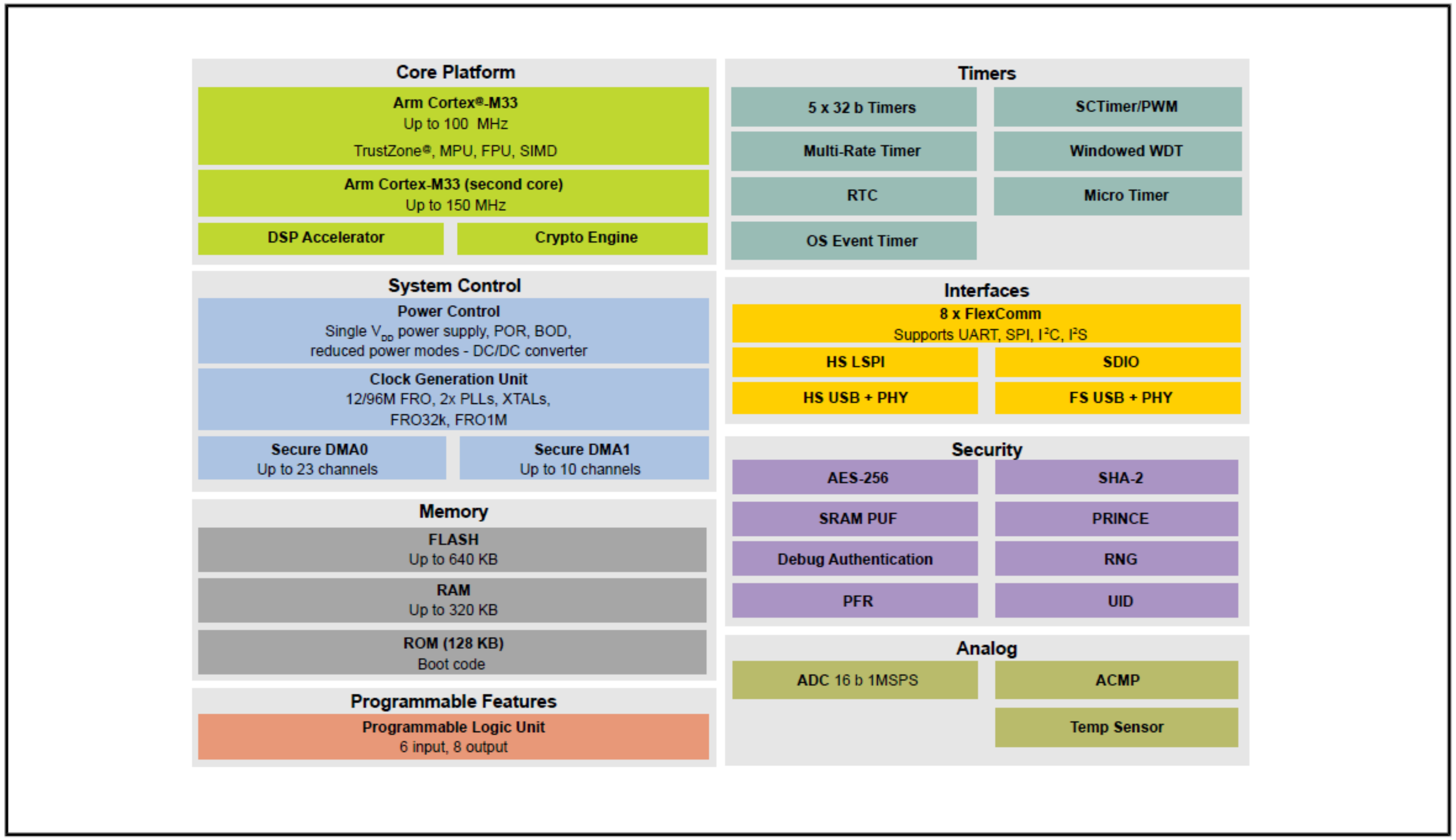
Results



	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

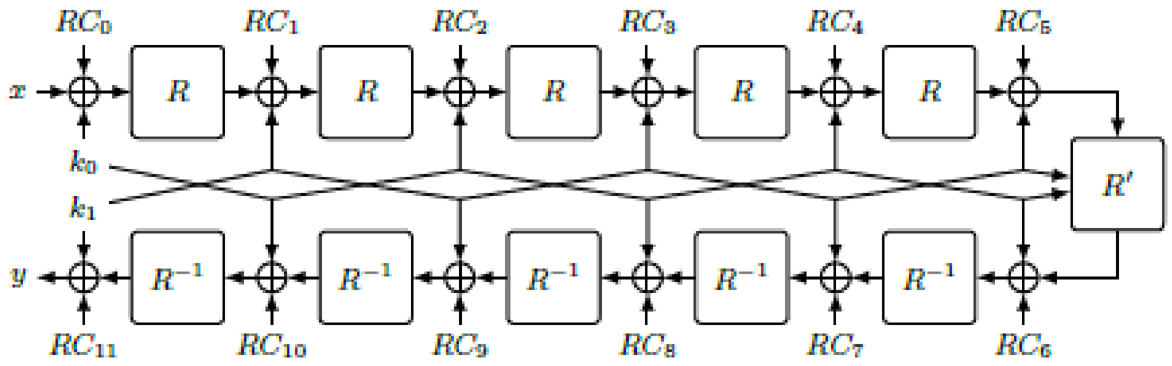
PRINCE: Deployment (NXP)

LPC55S6x MCU Block Diagram



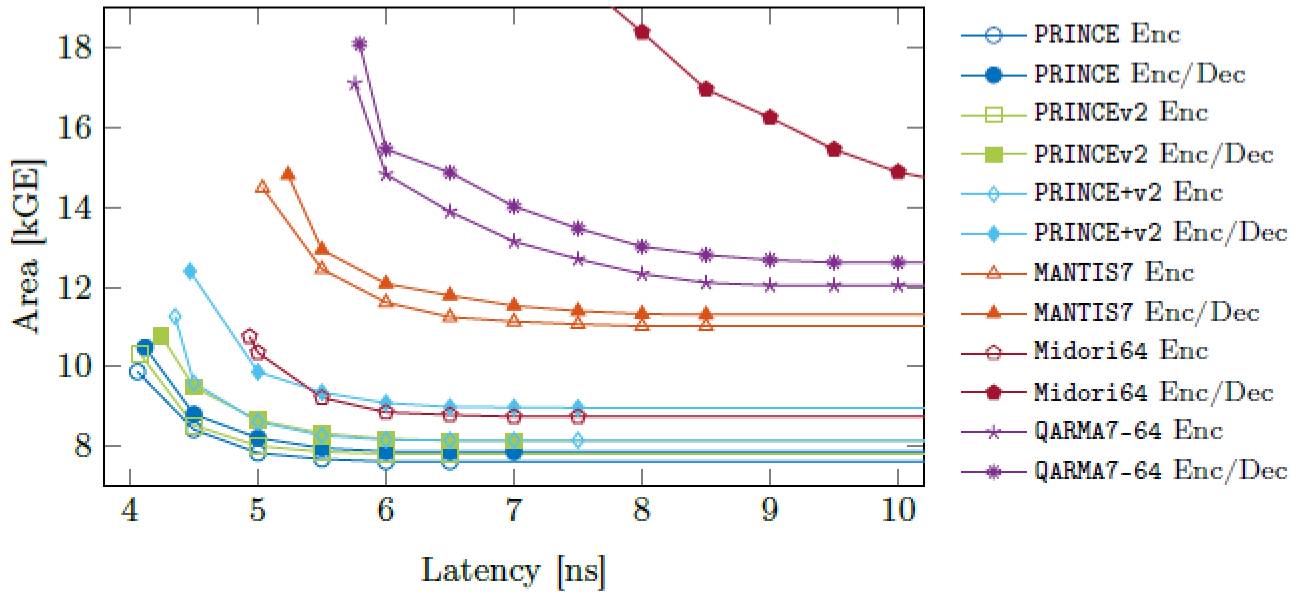
PRINCEv2 Cipher

- 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

Comparison of Low-Latency Proposals



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

* Bozilov et al., PRINCEv2: More Security for (Almost) No Overhead, SAC, 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	PRINCE ?	PRIDE LBlock TWINE ? KLEIN SPECK

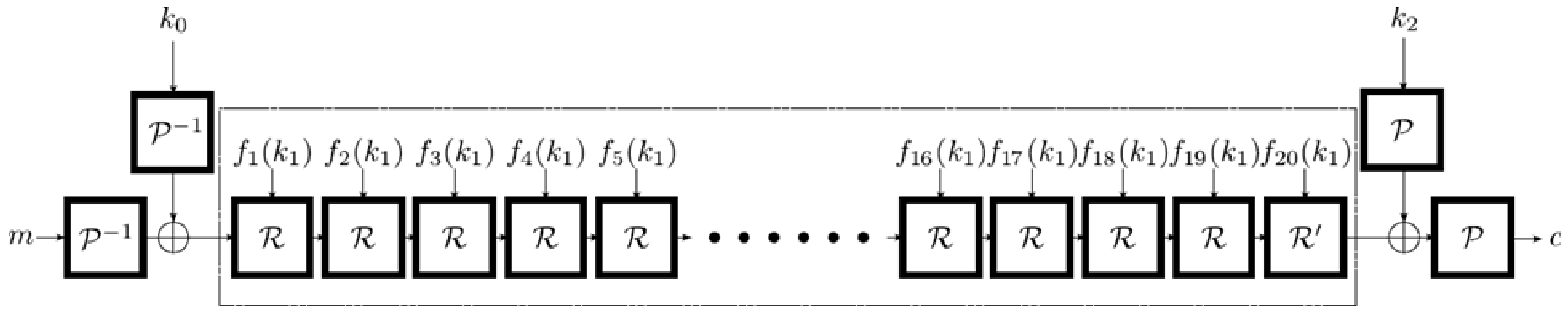
Efficiency in Software: PRIDE Cipher

- A block cipher optimized for software implementations on widely-used embedded microprocessors
  (*e.g.*, **Atmel ATmega8A**)
  - Benchmark
    - SPECK-64/128 cipher of NSA* (also uses ATMega8A)

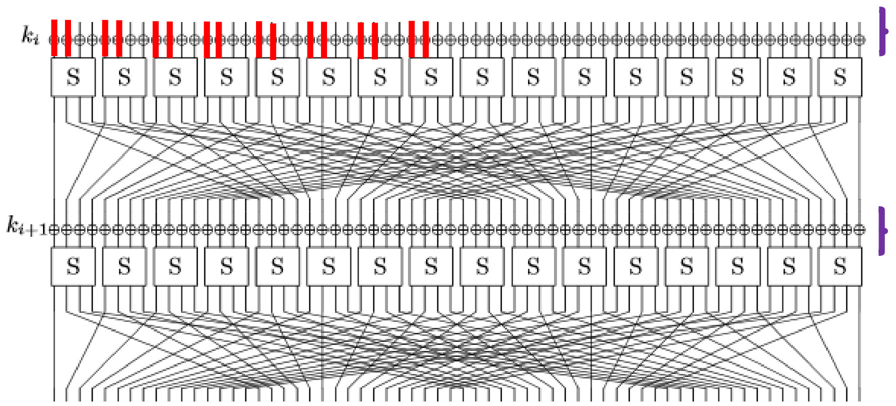
* Beaulieu et al, The Simon and Speck Families of Lightweight Block Ciphers, IACR ePrint Archive 2013/404, 2013

Efficiency in Software: PRIDE Cipher

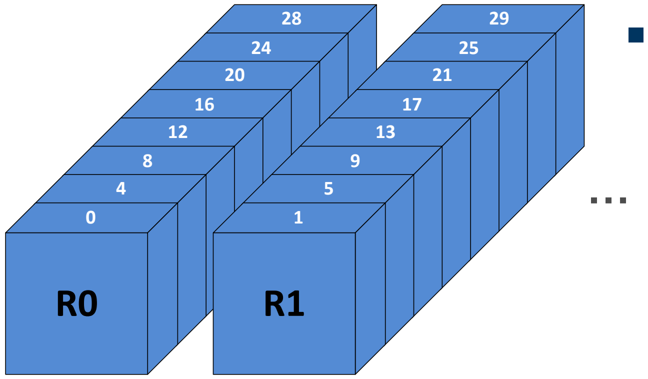
- Bitslicing idea used in design
  - Brings additional permutation without any further effort
- Substitution-permutation network
- 64-bit block, 128-bit key, 64-bit whitening keys
- 20 rounds
  - Last round different – no diffusion as it is not necessary


k_0
k_2
m
$\mathcal{P}^{-1}$
$\mathcal{P}^{-1}$
$f_1(k_1)$ $f_2(k_1)$ $f_3(k_1)$ $f_4(k_1)$ $f_5(k_1)$
$f_{16}(k_1)$ $f_{17}(k_1)$ $f_{18}(k_1)$ $f_{19}(k_1)$ $f_{20}(k_1)$
$\mathcal{R}$
$\mathcal{R}'$
$\mathcal{P}$
$\mathcal{P}$
c

Efficiency in Software: PRIDE Cipher

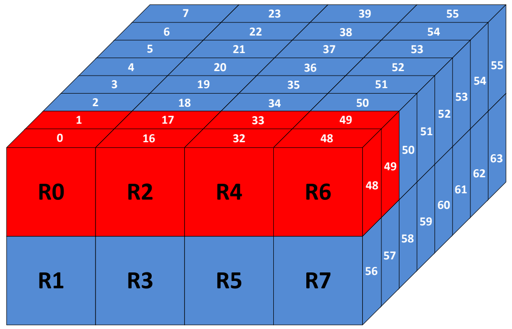


Additional permutation



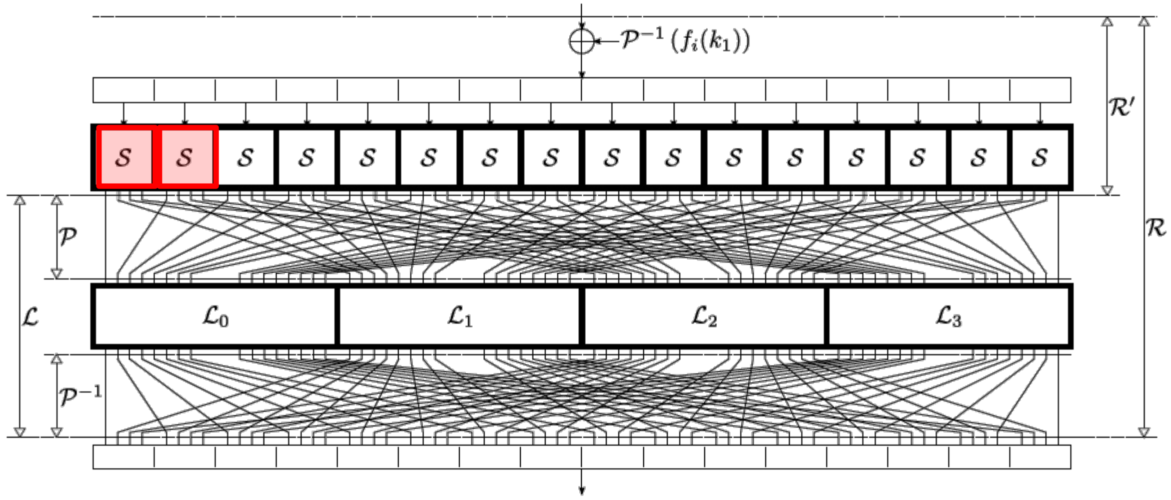
- Sbox: *ANF*
  - $a' = f_a(a, b, c, d)$
    $b' = f_b(a, b, c, d)$
    $c' = f_c(a, b, c, d)$
    $d' = f_d(a, b, c, d)$

Efficiency in Software: PRIDE Cipher

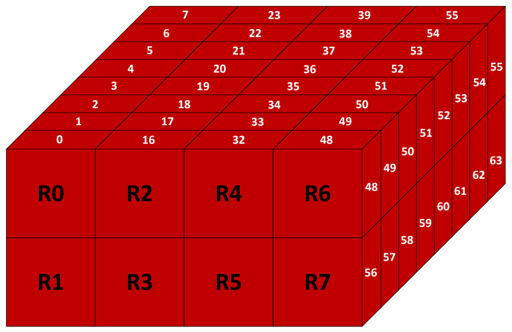


Sbox

Permutation



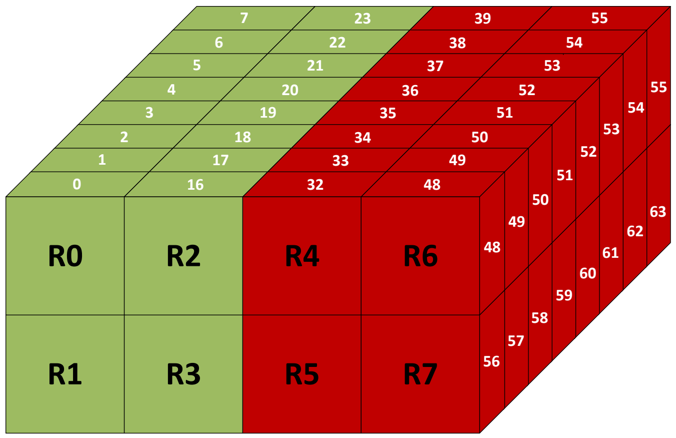
Efficiency in Software: PRIDE Cipher



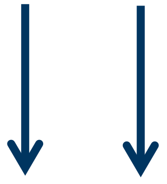
4-bit involution Sbox (formulation)

$$\underbrace{\begin{aligned} R0' &= R4 \textbf{ XOR } (R0 \textbf{ AND } R2)\\ R2' &= R6 \textbf{ XOR } (R2 \textbf{ AND } R4)\\ R4' &= R0 \textbf{ XOR } (R0' \textbf{ AND } R2')\\ R6' &= R2 \textbf{ XOR } (R2' \textbf{ AND } R4') \end{aligned}}_{\textbf{10 instructions}} \qquad \underbrace{\begin{aligned} R1' &= R5 \textbf{ XOR } (R1 \textbf{ AND } R2)\\ R3' &= R7 \textbf{ XOR } (R3 \textbf{ AND } R5)\\ R5' &= R1 \textbf{ XOR } (R1' \textbf{ AND } R3')\\ R7' &= R3 \textbf{ XOR } (R3' \textbf{ AND } R5') \end{aligned}}_{\textbf{10 instructions}}$$

Efficiency in Software: PRIDE Cipher

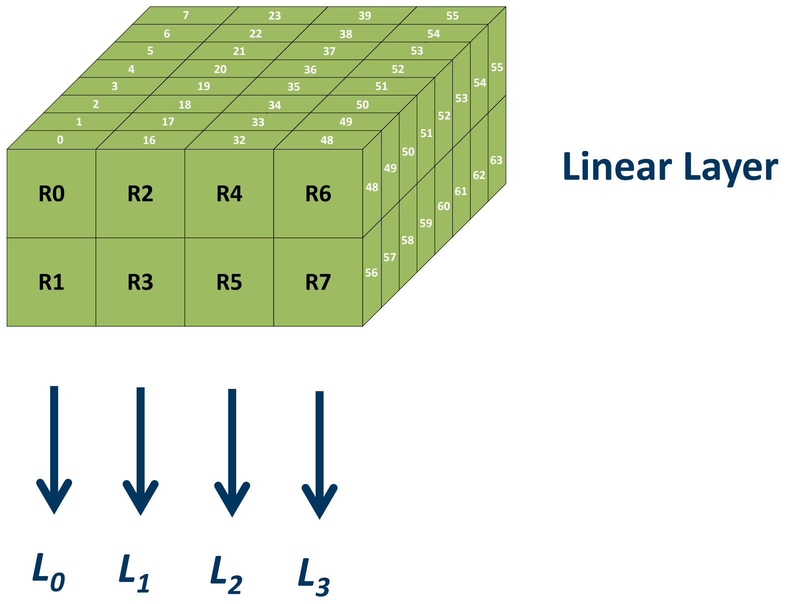


Linear Layer



L_0 L_1

Efficiency in Software: PRIDE Cipher



Matrix multiplication with 16 x 16 matrices!

Should not be very costly: Look for efficiently implementable L_i!

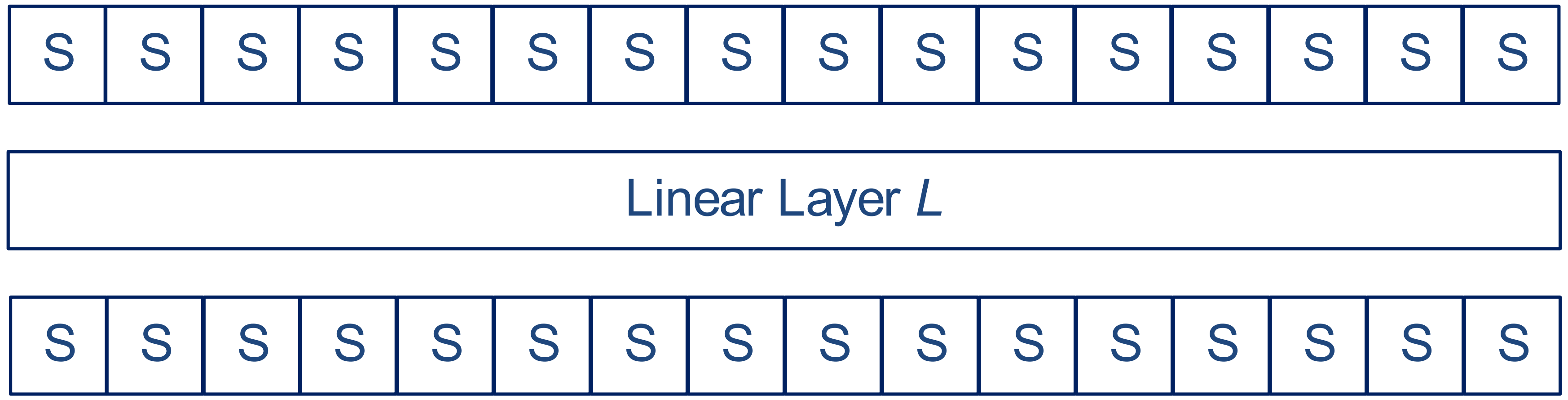
Efficiency in Software: PRIDE Cipher

- Linear layer: Expensive

- One PRESENT round's linear layer cost:
  - Just **wiring** (no cost) in hardware
  - **144 clock cycles** in software (Atmel ATmega8A)

- Try a different approach for better implementation

Efficiency in Software: PRIDE Cipher


S S S S S S S S S S S S S S S S
Linear Layer *L*
S S S S S S S S S S S S S S S S

Block interleaving construction

$$L = P \circ \begin{pmatrix} L_1 & & \\ & \ddots & \\ & & L_k \end{pmatrix} \circ P^{-1}$$

Efficiency in Software: PRIDE Cipher

How to Choose L_i

- Looking for “the cheapest implementation of a given linear layer *L*”?

NO!

Instead...

- “Which good linear layer can be implemented with ‘***N***‘ (minimum) instructions”?
  - In turn it gives us also...
    - # of *clock cycles* for speed
    - #of *bytes* for code size

Efficiency in Software: PRIDE Cipher

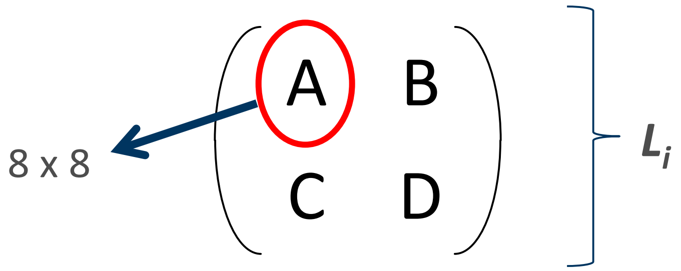
- Search for ‘efficient‘ permutation matrices
  - Look for 4 efficiently-implementable 16x16 binary permutation matrices
  - Similar to Sbox search of Ullrich et al.*
    - Search performed on hardware platform instead of software platform
    - Faster search, larger search space

* Ullrich et al., Finding Optimal Bitsliced Implementations of 4 x 4-bit S-boxes, SKEW 2011

Efficiency in Software: PRIDE Cipher

- Search in a subset of possible 16 x 16 matrices on FPGA platform (reconfigurable hardware)

  - 16 x 16 still quite large...



Efficiency in Software: PRIDE Cipher

- Limit number of instructions
  - CLC, EOR, MOV, MOVW, CLR, SWAP, ASR, ROR, ROL, LSR, LSL
- Limit number of used registers
  - 2 state, 4 temporary registers
- Try all possible combinations of instructions and registers
- Save the matrices generating appropriate code
  - Out of these, look for the ones with least instructions
- Ended up with ***36 instructions*** for the whole linear layer!
  - L_0=7, L_1=11, L_2=7, L_3=11, L_0^{-1}=7, L_1^{-1}=13, L_2^{-1}=7, L_3^{-1}=13

Efficiency in Software: PRIDE Cipher

UNIVERSITY OF PASSAU
Faculty of Computer Science and Mathematics

Results

One Round Cost	*Key Update*	*Key Addition*	*Sbox Layer*	*Linear Layer*	***Total***
Time (Clock Cycles)	4	8	20	36	**68**
Code Size (Bytes)	8	16	40	72	**136**

Efficiency in Software: PRIDE Cipher

UNIVERSITY OF PASSAU
Faculty of Computer Science and Mathematics

Results

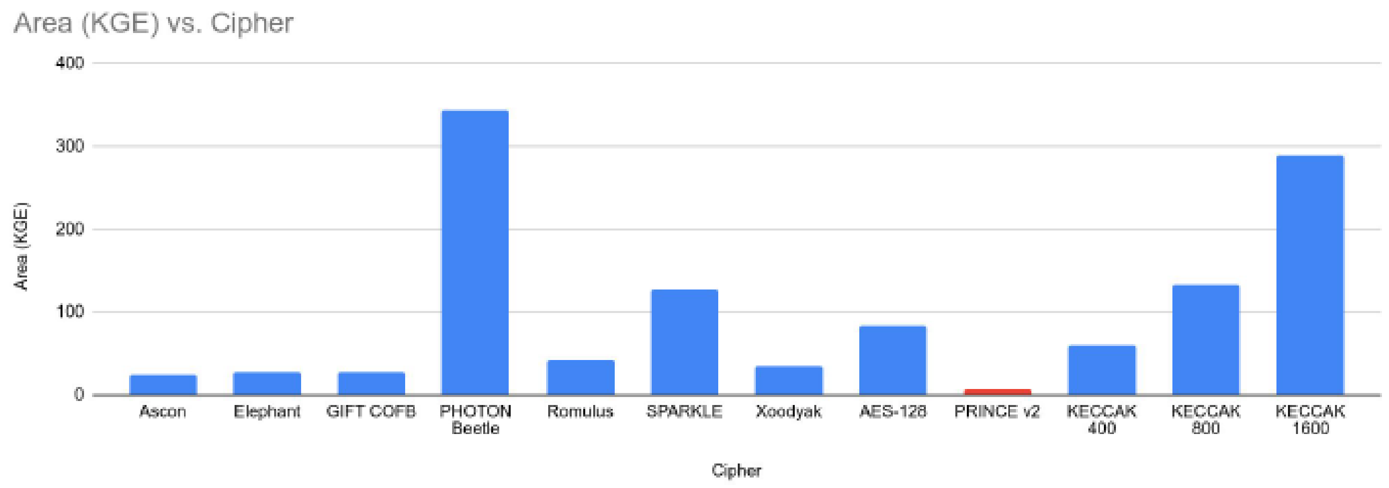
	AES-128	SERPENT-128	PRESENT-128	CLEFIA-128	SEA-96	NOEKEON-128	PRINCE-128	ITUBee-80	SIMON-64/128*	SPECK-64/96*	SPECK-64/128*	**PRIDE**
Time (Clock Cycles)	3159	49314	10792	28648	17745	23517	3614	2607	2000	1152	1200	**1514**
Code Size (Bytes)	1570	7220	660	3046	386	364	1108	716	282	182	186	**266**

* Data & key read-write omitted

- Performance on Atmel AVR 8- bit microcontroller (encryption)
  - *PRIDE decryption:* 1570 clock cycles and 282 bytes
- Results close to SPECK
  - Good results for a "traditional" design

ASCON?

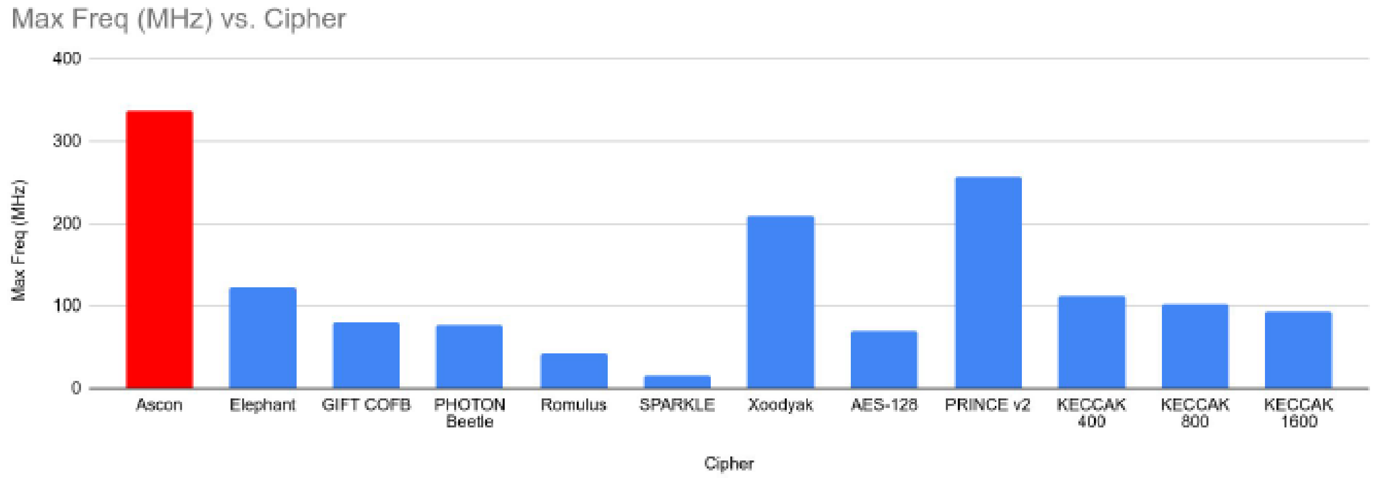
Compactness												
	Ascon	Elephant	GIFT COFB	PHOTON Beetle	Romulus	SPARKLE	Xoodyak	AES-128	**PRINCE v2**	KECCAK 400	KECCAK 800	KECCAK 1600
Area (KGE)	24.437	28.350	28.709	343.55	43.392	128.51	34.148	83.581	**7.9813**	60.128	132.74	289.92



* Yalcin et al., Need for Low-latency Ciphers: A Comparative Study of NIST LWC Finalists, NIST Lightweight Crypto WS, 2022

ASCON?

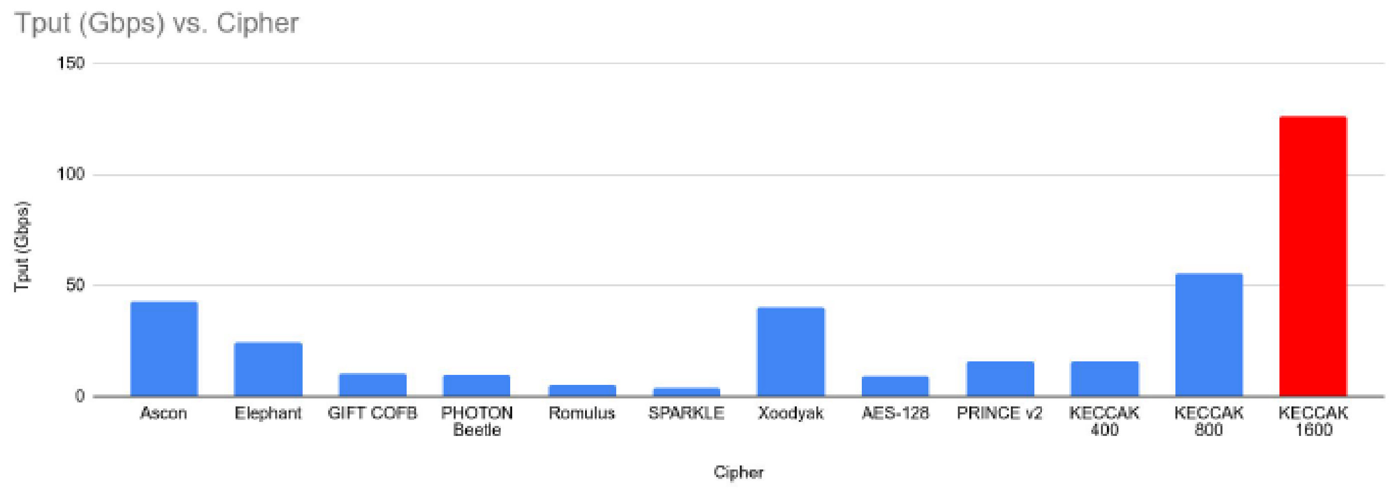
High Frequency												
	Ascon	Elephant	GIFT COFB	PHOTON Beetle	Romulus	SPARKLE	Xoodyak	AES-128	PRINCE v2	KECCAK 400	KECCAK 800	KECCAK 1600
Max Freq (MHz)	**336.7**	122.55	79.81	77.700	43.535	16.736	209.21	69.735	257.07	112.36	102.88	93.897



* Yalcin et al., Need for Low-latency Ciphers: A Comparative Study of NIST LWC Finalists, NIST Lightweight Crypto WS, 2022

ASCON?

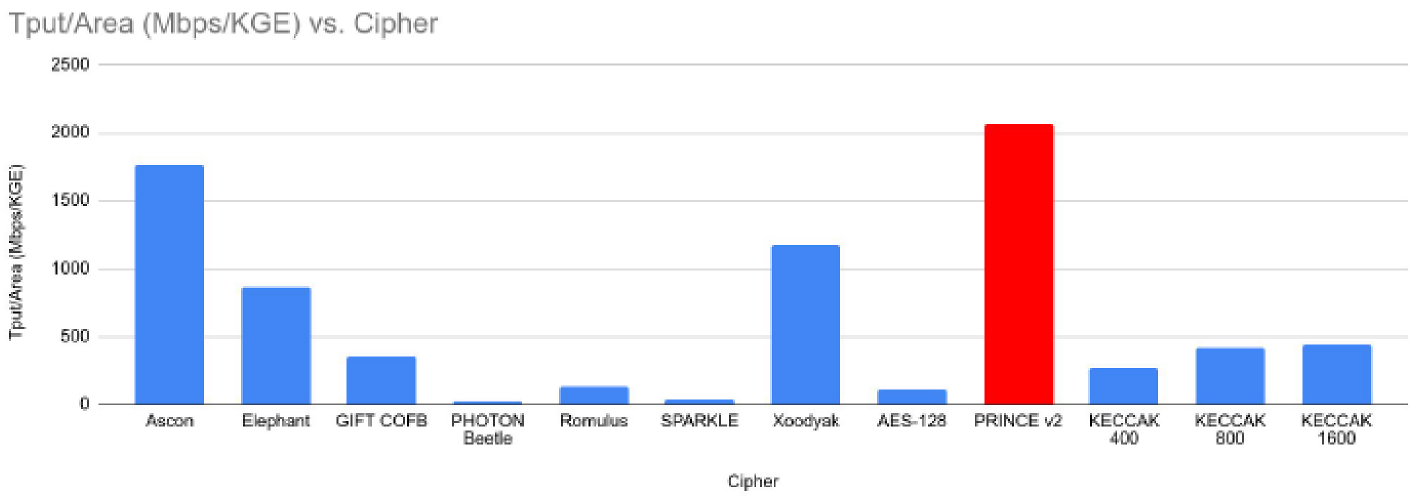
High Throughput											
	Ascon	Elephant	GIFT COFB	PHOTON Beetle	Romulus	SPARKLE	Xoodyak	AES-128	PRINCE v2	KECCAK 400	KECCAK 800	**KECCAK 1600**
Tput (Gbps)	43.098	24.510	10.215	9.945	5.572	4.2845	40.167	8.9261	16.452	16.180	55.967	**126.20**



* Yalcin et al., Need for Low-latency Ciphers: A Comparative Study of NIST LWC Finalists, NIST Lightweight Crypto WS, 2022

ASCON?

Throughput per Area												
	Ascon	Elephant	GIFT COFB	PHOTON Beetle	Romulus	SPARKLE	Xoodyak	AES-128	**PRINCE v2**	KECCAK 400	KECCAK 800	KECCAK 1600
Tput/Area (Mbps/KGE)	1763	864.54	355.82	28.949	128.42	33.339	1176	106.80	**2061**	269.09	421.64	435.28



* Yalcin et al., Need for Low-latency Ciphers: A Comparative Study of NIST LWC Finalists, NIST Lightweight Crypto WS, 2022

LWC on FPGAs

- ASIC vs. FPGA for LWC implementations
  - Metrics may not always match
    - Gates or Gate Equivalents vs. DSP and RAM Blocks
  - Example study: PRESENT implementation [*]
    - RAM-based
    - Uses existing RAM blocks for storing internal states
    - Hence reduces slice count
    - Two versions
      1. Sbox also on RAM (further #slice reduction)
      2. Sbox on slices

[*] Kavun and Yalcin, RAM-Based Ultra-Lightweight FPGA Implementation of PRESENT, ReConFig'11, 2011.

LWC on FPGAs

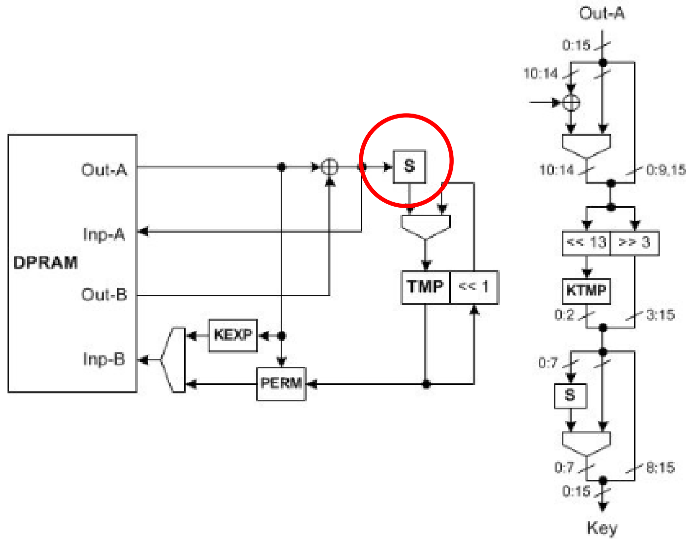


Figure 10. PRESENT block diagram (key expansion module shown on the right) for on-slice S-box version.

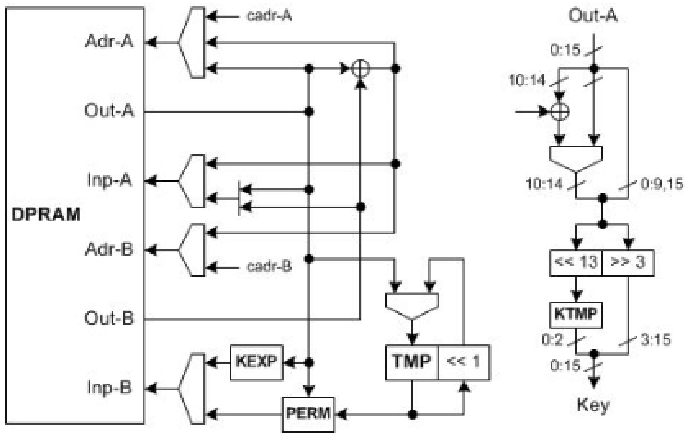
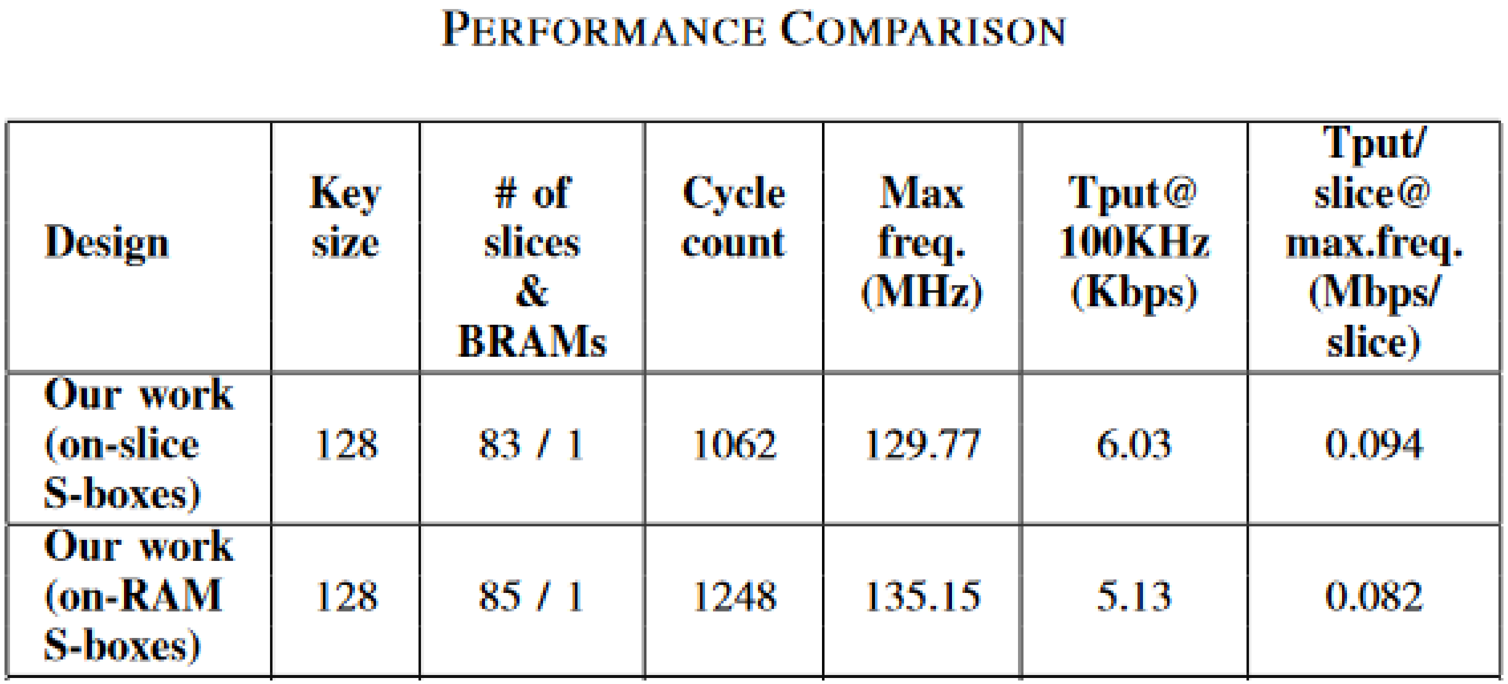


Figure 11. PRESENT block diagram (key expansion module shown on the right) for on-RAM S-box version.

[*] Kavun and Yalcin, RAM-Based Ultra-Lightweight FPGA Implementation of PRESENT, ReConFig'11, 2011.

LWC on FPGAs



PERFORMANCE COMPARISON

Design	Key size	# of slices & BRAMs	Cycle count	Max freq. (MHz)	Tput@ 100KHz (Kbps)	Tput/ slice@ max.freq. (Mbps/ slice)
Our work (on-slice S-boxes)	128	83 / 1	1062	129.77	6.03	0.094
Our work (on-RAM S-boxes)	128	85 / 1	1248	135.15	5.13	0.082

[*] Kavun and Yalcin, RAM-Based Ultra-Lightweight FPGA Implementation of PRESENT, ReConFig'11, 2011.

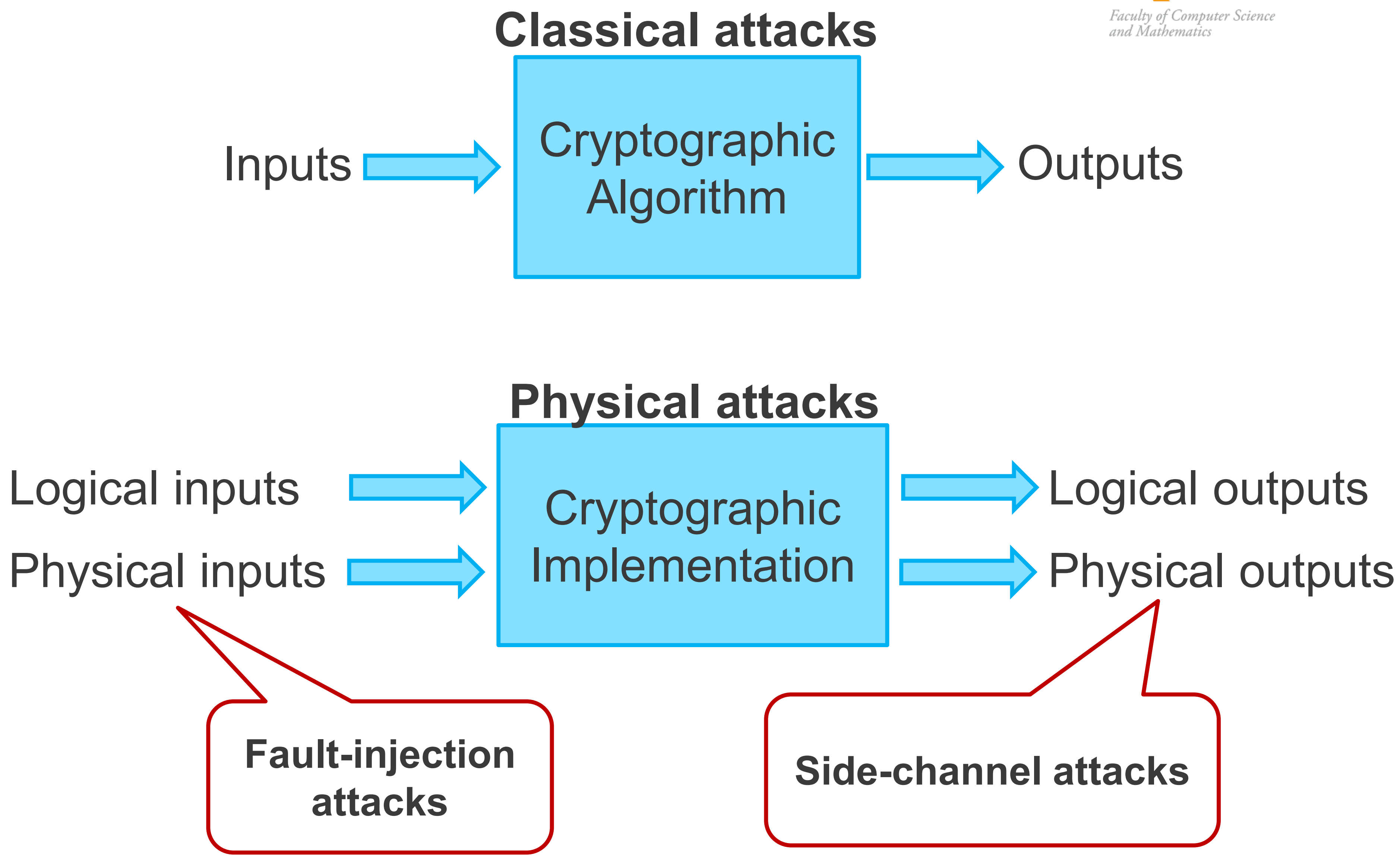
LWC on FPGAs

- Remarks

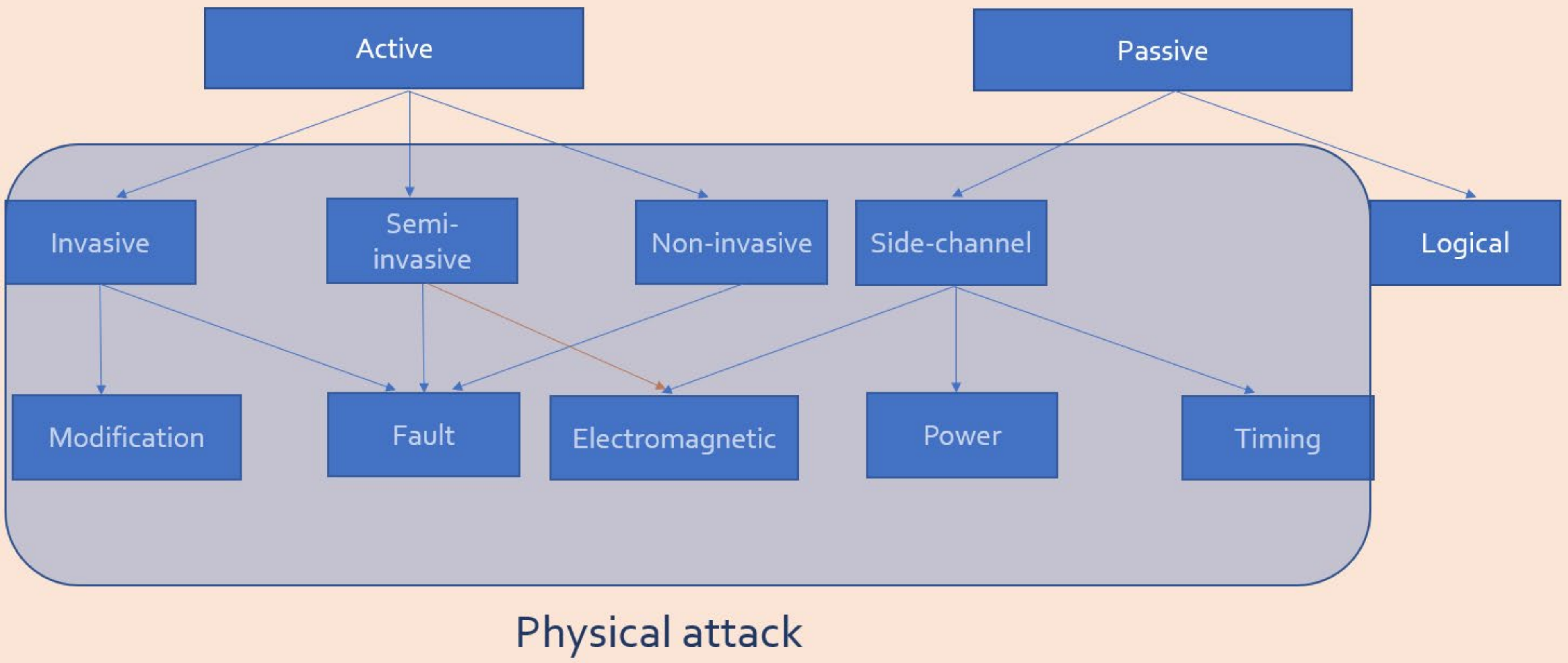
  - Gates or Gate Equivalents (GE) vs. DSP and RAM Blocks
    - Less #GE is required for lightweight ciphers in ASIC
    - Sbox consumes lot of GEs (area)
    - Hence, Sboxes in lightweight cipher designs optimized heavily
    - However…

    - Example:
      - PRESENT cipher Sbox costs 25 GE (90 nm Faraday library)
      - PRINCE cipher Sbox costs 16 GE (90 nm Faraday library)
      - **Same cost on FPGA**
        - (1 BRAM block, if PRINCE implemented similar to [*])

Reality: Physical Attacks



Reality: Physical Attacks



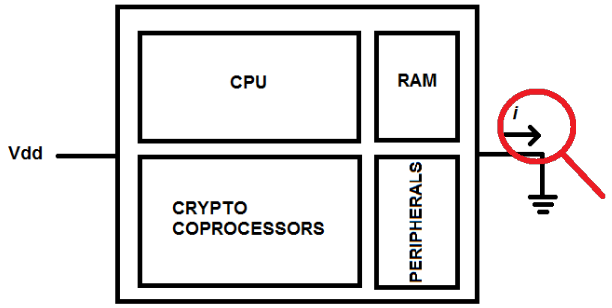
Reality: Physical Attacks

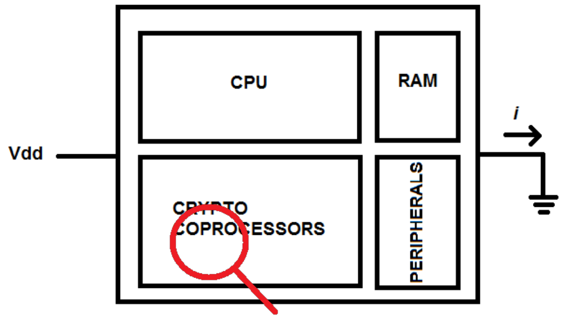
Attack-resistant Lightweight Crypto Implementations

- 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





Reality: Physical Attacks

Countermeasures

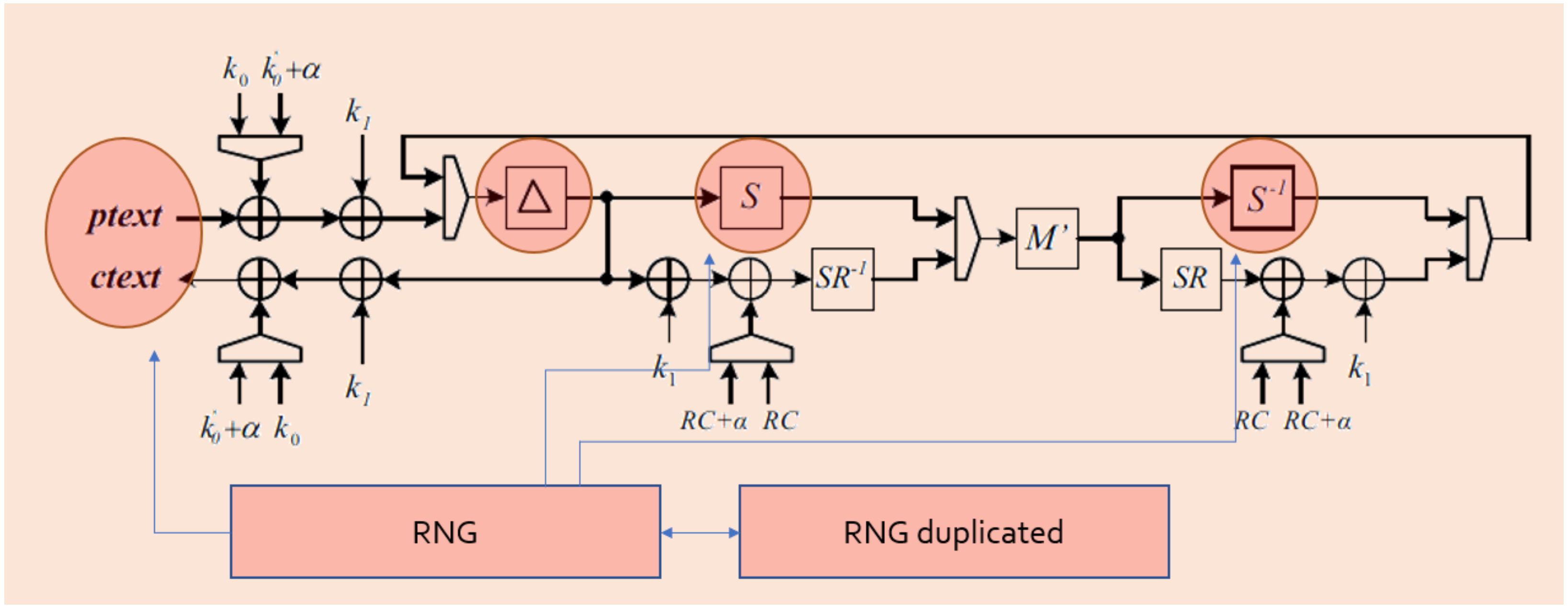
- Masking – Novel techniques
  - Threshold Implementations
  - Domain-oriented masking
  - Comes with cost, sharing of secret triples/quadruples costs (even for first-order security)

- 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)

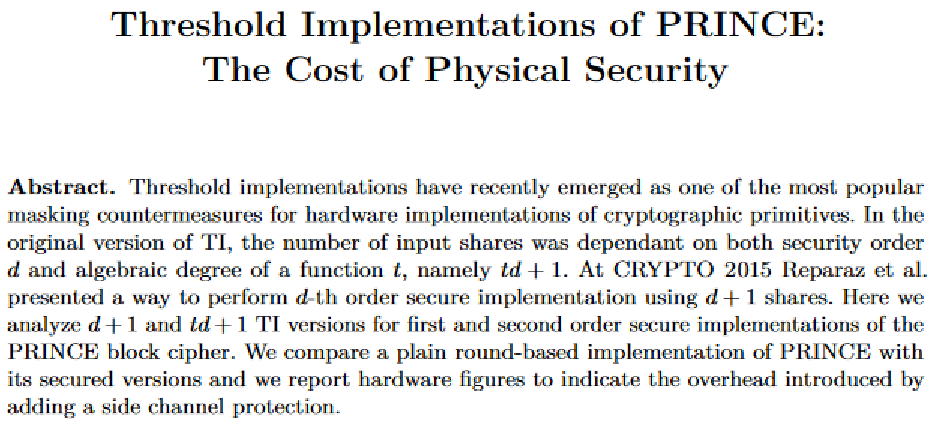
Reality: Physical Attacks

PRINCE: When Protected



SCA-resistant “Threshold Implementation” of PRINCE

Bozilov et al (KU Leuven) at LWC Workshop 2016



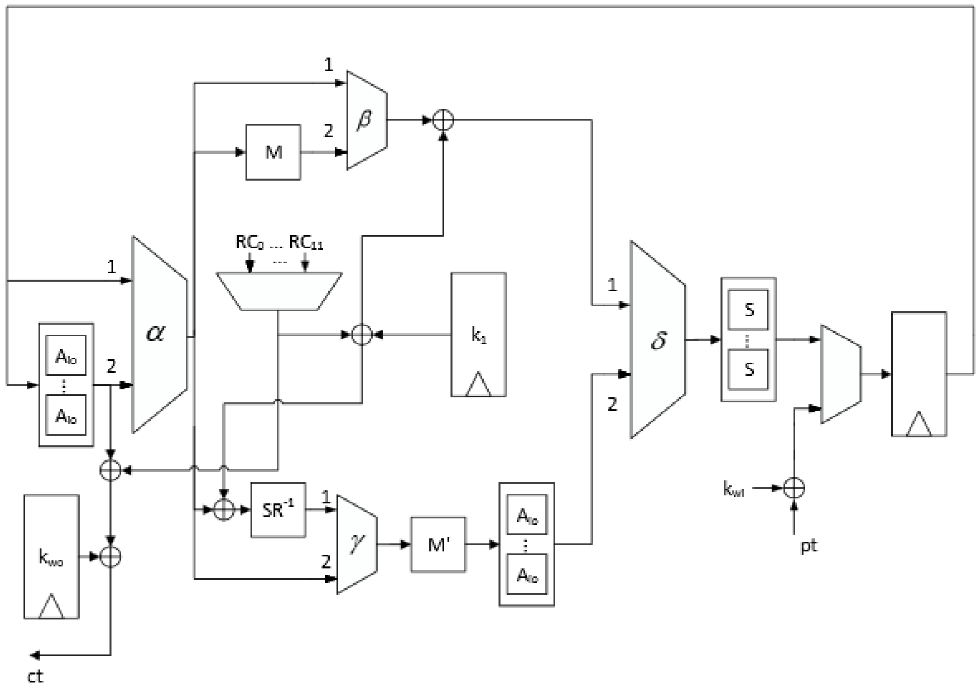
Threshold Implementations of PRINCE:
The Cost of Physical Security

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.

Reality: Physical Attacks

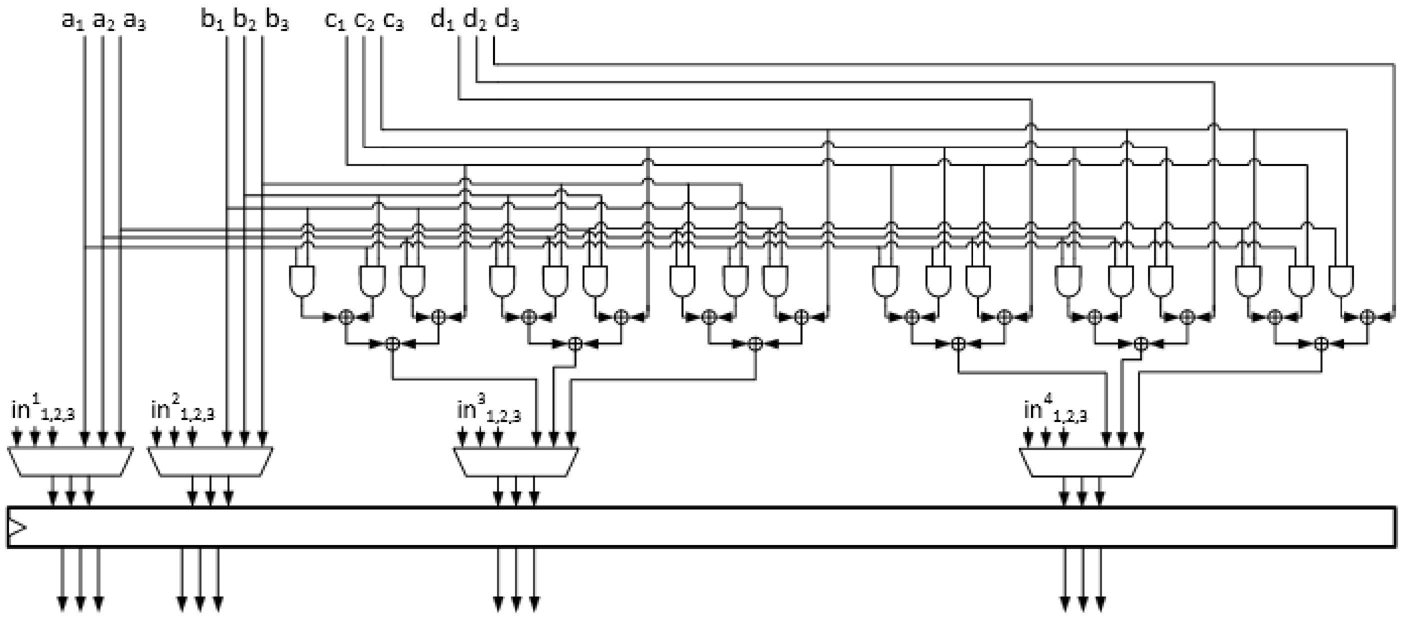
SCA-resistant "Threshold Implementation" of PRINCE

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



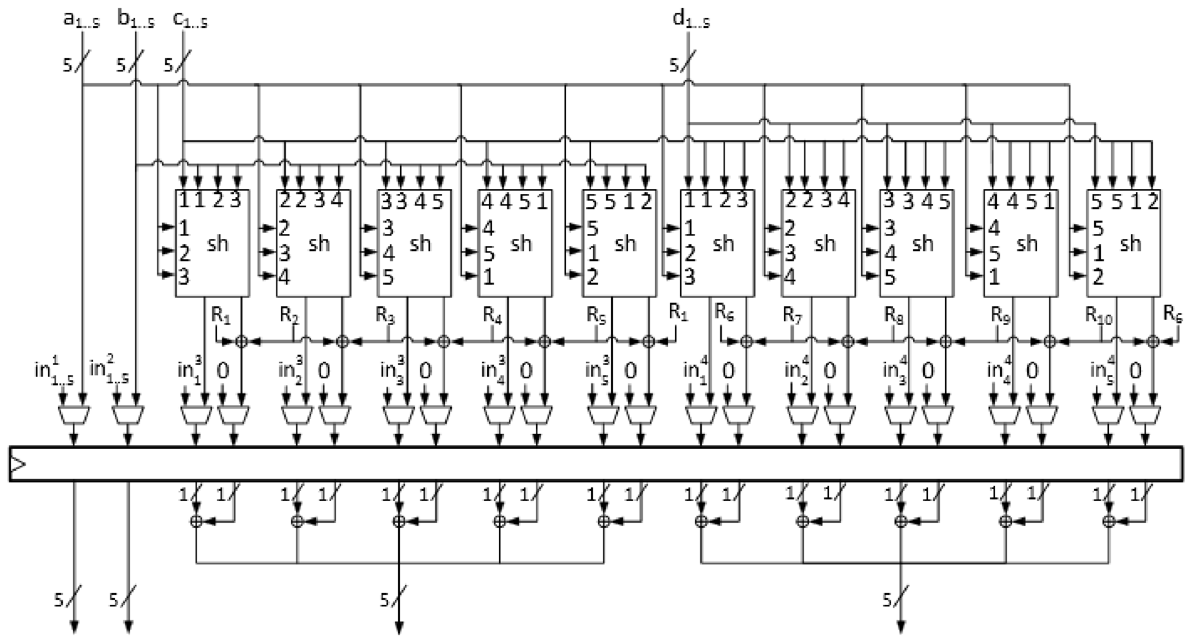
SCA-resistant “Threshold Implementation” of PRINCE

- 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



UNIVERSITY OF PASSAU
Faculty of Computer Science and Mathematics

SCA-resistant "Threshold Implementation" of PRINCE

Results

PRINCE-128 (round-based implementation) unprotected

Technology	Area (GE)
ASIC, 90nm	3589

PRINCE-128 (round-based implementation) 1st-order secure

Technology	Area (GE)
ASIC, 90nm	11958

PRINCE-128 (round-based implementation) 2nd-order secure

Technology	Area (GE)
ASIC, 90nm	21879

Future Steps

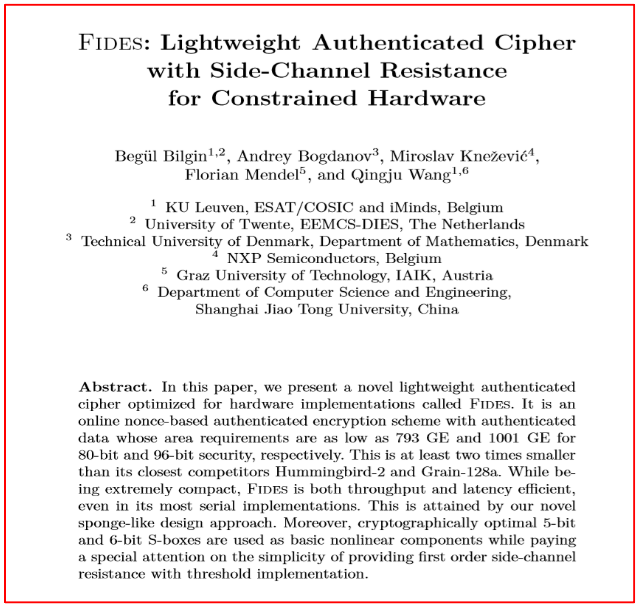
UNIVERSITY OF PASSAU
Faculty of Computer Science and Mathematics

- 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 better resource utilization!

New Directions

SCA resistance

Low-latency

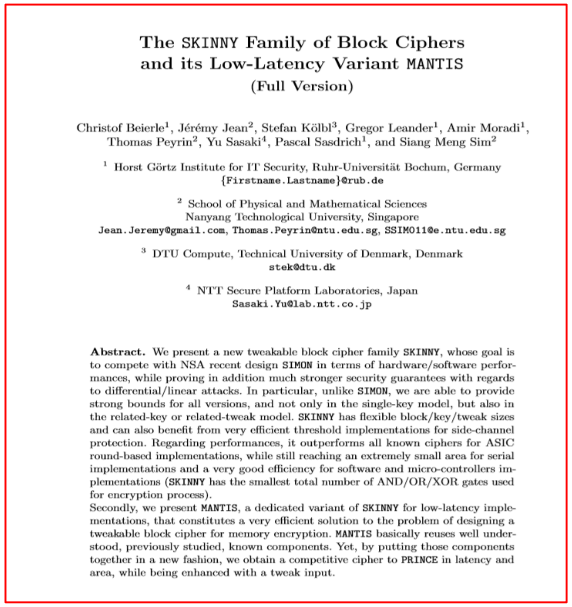


FIDES: Lightweight Authenticated Cipher with Side-Channel Resistance for Constrained Hardware

Begül Bilgin[1,2], Andrey Bogdanov[3], Miroslav Knežević[4], Florian Mendel[5], and Qingju Wang[1,6]

[1] KU Leuven, ESAT/COSIC and iMinds, Belgium
[2] University of Twente, EEMCS-DIES, The Netherlands
[3] Technical University of Denmark, Department of Mathematics, Denmark
[4] NXP Semiconductors, Belgium
[5] Graz University of Technology, IAIK, Austria
[6] Department of Computer Science and Engineering, Shanghai Jiao Tong University, China

Abstract. In this paper, we present a novel lightweight authenticated cipher optimized for hardware implementations called FIDES. It is an online nonce-based authenticated encryption scheme with authenticated data whose area requirements are as low as 793 GE and 1001 GE for 80-bit and 96-bit security, respectively. This is at least two times smaller than its closest competitors Hummingbird-2 and Grain-128a. While being extremely compact, FIDES is both throughput and latency efficient, even in its most serial implementations. This is attained by our novel sponge-like design approach. Moreover, cryptographically optimal 5-bit and 6-bit S-boxes are used as basic nonlinear components while paying a special attention on the simplicity of providing first order side-channel resistance with threshold implementation.



The SKINNY Family of Block Ciphers and its Low-Latency Variant MANTIS (Full Version)

Christof Beierle[1], Jérémy Jean[2], Stefan Kölbl[3], Gregor Leander[1], Amir Moradi[1], Thomas Peyrin[2], Yu Sasaki[4], Pascal Sasdrich[1], and Siang Meng Sim[2]

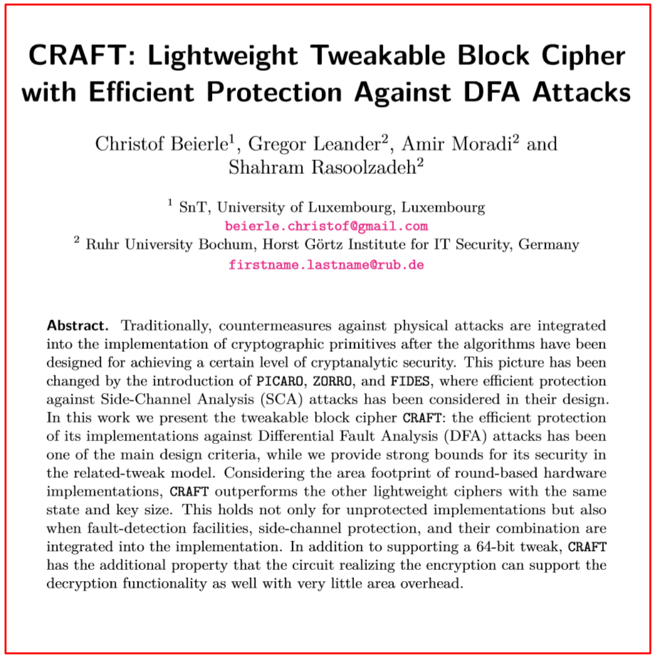
[1] Horst Görtz Institute for IT Security, Ruhr-Universität Bochum, Germany
{Firstname.Lastname}@rub.de

[2] School of Physical and Mathematical Sciences
Nanyang Technological University, Singapore
Jean.Jeremy@gmail.com, Thomas.Peyrin@ntu.edu.sg, SSIM011@e.ntu.edu.sg

[3] DTU Compute, Technical University of Denmark, Denmark
stek@dtu.dk

[4] NTT Secure Platform Laboratories, Japan
Sasaki.Yu@lab.ntt.co.jp

Abstract. We present a new tweakable block cipher family SKINNY, whose goal is to compete with NSA recent design SIMON in terms of hardware/software performances, while proving in addition much stronger security guarantees with regards to differential/linear attacks. In particular, unlike SIMON, we are able to provide strong bounds for all versions, and not only in the single-key model, but also in the related-key or related-tweak model. SKINNY has flexible block/key/tweak sizes and can also benefit from very efficient threshold implementations for side-channel protection. Regarding performances, it outperforms all known ciphers for ASIC round-based implementations, while still reaching an extremely small area for serial implementations and a very good efficiency for software and micro-controllers implementations (SKINNY has the smallest total number of AND/OR/XOR gates used for encryption process).
Secondly, we present MANTIS, a dedicated variant of SKINNY for low-latency implementations, that constitutes a very efficient solution to the problem of designing a tweakable block cipher for memory encryption. MANTIS basically reuses well understood, previously studied, known components. Yet, by putting those components together in a new fashion, we obtain a competitive cipher to PRINCE in latency and area, while being enhanced with a tweak input.



CRAFT: Lightweight Tweakable Block Cipher with Efficient Protection Against DFA Attacks

Christof Beierle[1], Gregor Leander[2], Amir Moradi[2] and Shahram Rasoolzadeh[2]

[1] SnT, University of Luxembourg, Luxembourg
beierle.christof@gmail.com
[2] Ruhr University Bochum, Horst Görtz Institute for IT Security, Germany
firstname.lastname@rub.de

Abstract. Traditionally, countermeasures against physical attacks are integrated into the implementation of cryptographic primitives after the algorithms have been designed for achieving a certain level of cryptanalytic security. This picture has been changed by the introduction of PICARO, ZORRO, and FIDES, where efficient protection against Side-Channel Analysis (SCA) attacks has been considered in their design. In this work we present the tweakable block cipher CRAFT: the efficient protection of its implementations against Differential Fault Analysis (DFA) attacks has been one of the main design criteria, while we provide strong bounds for its security in the related-tweak model. Considering the area footprint of round-based hardware implementations, CRAFT outperforms the other lightweight ciphers with the same state and key size. This holds not only for unprotected implementations but also when fault-detection facilities, side-channel protection, and their combination are integrated into the implementation. In addition to supporting a 64-bit tweak, CRAFT has the additional property that the circuit realizing the encryption can support the decryption functionality as well with very little area overhead.

UNIVERSITY OF PASSAU

Faculty of Computer Science and Mathematics

Thanks for listening!

Any questions?

(elif.kavun@uni-passau.de)