# The ACRYPT project

## Lightweight Cryptography for the Internet of Things

Alex Biryukov, Daniel Dinu, Johann Großschädl
Dmitry Khovratovich, Yann Le Corre, Léo Perrin

SnT, University of Luxembourg

FSE 2015 Rump Session

# Description

- High level view of the algorithms
- Detailed description
- Best attacks
- Hardware implementation footprint (if available)
- 51 primitives!

# Description

- High level view of the algorithms
- Detailed description
- Best attacks
- Hardware implementation footprint (if available)
- 51 primitives!

**Let us know if you have new results!**

The best attack by the designers is a linear attack based on a 2-rounds iterative linear trail covering 9 rounds, which is then extended to cover 12 rounds through key guessing.

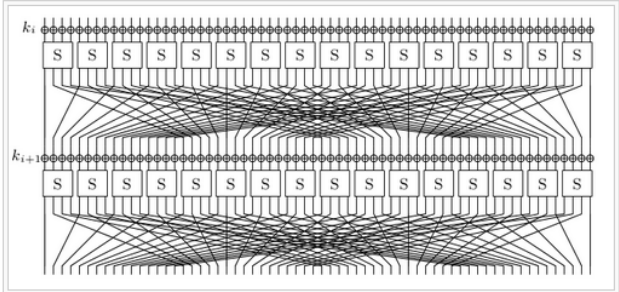
### PRESENT [edit]

- Article: *PRESENT: An Ultra-Lightweight Block Cipher*, CHES 07[40]
- Authors: A. Bogdanov, L.R. Knudsen, G. Leander, C. Paar, A. Poschmann, M.J.B. Robshaw, Y. Seurin, and C. Vikkelsoe
- Target: Hardware

This cipher is a SPN but, interestingly, it was not inspired by the AES. Indeed, while many SPN-based ciphers have permutation layers close in structure to that of the AES (see LED or mCrypton), that of PRESENT is completely different: it is bit oriented and is rather simple. It can be implemented in hardware using simple wiring. However, since bit-oriented permutations are not software-friendly, the target of PRESENT is clearly a hardware implementation. Its S-box was selected for its good cryptographic properties as well as for its small hardware footprint.

PRESENT is a very important design as it has been an inspiration for many others. For instance, its S-box has also been re-used by GOST revisited and LED as well as the lightweight hash function PHOTON. This cipher also inspired the design of two lightweight hash functions: DM-PRESENT and SPONGENT.

While only PRESENT-80 is described in the body of the CHES 07 article[40], PRESENT-128 and its modified key-schedule are described in the appendix. This cipher has been standardized and is part of the ISO-29192[73] with CLEFIA.



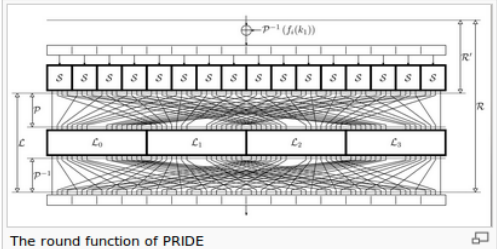
Two rounds of PRESENT encryption

### PRIDE [edit]

- Article: *Block Ciphers -- Focus On the Linear Layer (feat. PRIDE)*, CRYPTO'14[74]
- Authors: Martin R. Albrecht, Benedikt Driessen, Elif Bilge Kavun, Gregor Leander, Christof Paar and Tolga Yalcın
- Target: Software

PRIDE is the output of research focusing on the design of the linear layer in Substitution-Permutation Networks. Its main target is 8-bit micro-controllers. Specifically, the computer assisted search for components of the linear layer was optimized to look for permutations which can be efficiently implemented using the AVR instruction set.

To limit the overhead implied by the implementation of both encryption and decryption, its S-Box is an involution. The key-schedule is very similar to that of PRINCE: the master key is split in two halves, the first being used as whitening key and the second being used to derive subkeys XOR-ed in the internal state at every round. However, unlike in PRINCE, the post-whitening key is the same as the pre-whitening key and the subkeys are not derived by XOR-ing round constants but by adding round constants on some bytes using a regular addition modulo 256.


The round function of PRIDE

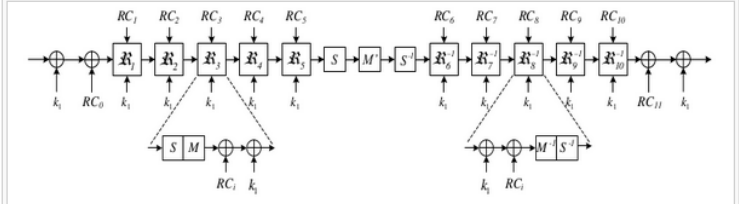
### PRINCE [edit]

- Article: *PRINCE – A Low-latency Block Cipher for Pervasive Computing Applications*, ASIACRYPT 12[45]
- Authors: Julia Borghoff, Anne Canteaut, Tim Guneysu, Elif Bilge Kavun, Miroslav Knezevic , Lars R. Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, Peter Rombouts, Søren S. Thomsen, and Tolga Yalcın
- Target: Hardware (low latency)

The main aim of the design of PRINCE is low latency.

There is no real key schedule: three 64 bits keys are derived from the 128 master key. Two are used as whitening keys and the third is simply xored in the internal state during encryption. To make the rounds behave differently from one another, different constants are xored in the internal state at each round. These constants $RC_i$ (i=0,..,11) are such that $RC_i \oplus RC_{11-i} = \alpha$ where $\alpha$ is a constant derived from $\pi$. This property, combined with the fact that the first 5 rounds are the inverse of the last 5 mean that the decryption algorithm for key k is identical to an encryption with key $k \oplus \alpha$. This property is refered to as "$\alpha$-reflexivity".

The authors challenge the symmetric cryptography community to attack (rounds-reduced versions of) this cipher and offer different rewards for "practical" attacks.


The PRINCE-core algorithm

## Feistel Networks [edit]

### Two Branched [edit]

In this category, we put all the Feistel networks operating on blocks of size 2n for which the Feistel function maps n bits to n bits.

#### DESLX [edit]

- Article: *New Lightweight DES Variants*, FSE 07[10]
- Authors: Gregor Leander, Christof Paar, Axel Poschmann, and Kai Schramm
- Target: Hardware

# Benchmark description

- 1 paper[1]: *Triathlon of Lightweight Block Ciphers for the IoT*

[1] eprint 2015/209 (`https://https://eprint.iacr.org/2015/209`)

# Benchmark description

- 1 paper[1]: *Triathlon of Lightweight Block Ciphers for the IoT*
- 1 benchmarking framework (`FELICS`: Fair Evaluation of LIghtweight Cryptograhic Systems)

[1] eprint 2015/209 (`https://https://eprint.iacr.org/2015/209`)

# Benchmark description

- 1 paper[1]: *Triathlon of Lightweight Block Ciphers for the IoT*
- 1 benchmarking framework (`FELICS`: Fair Evaluation of LIghtweight Cryptograhic Systems)
- 3 different platforms (8-bit AVR, 16-bit MSP, 32-bit ARM)

[1] eprint 2015/209 (`https://https://eprint.iacr.org/2015/209`)

# Benchmark description

- 1 paper[1]: *Triathlon of Lightweight Block Ciphers for the IoT*
- 1 benchmarking framework (FELICS: Fair Evaluation of LIghtweight Cryptograhic Systems)
- 3 different platforms (8-bit AVR, 16-bit MSP, 32-bit ARM)
- 3 different usage scenarios

[1] eprint 2015/209 (https://https://eprint.iacr.org/2015/209)

# Benchmark description

- 1 paper[1]: *Triathlon of Lightweight Block Ciphers for the IoT*
- 1 benchmarking framework (`FELICS`: `F`air `E`valuation of `LI`ghtweight `C`ryptograhic `S`ystems)
- 3 different platforms (8-bit AVR, 16-bit MSP, 32-bit ARM)
- 3 different usage scenarios
- 3 different metrics: RAM, clock cycles, code size

[1] eprint 2015/209 (`https://https://eprint.iacr.org/2015/209`)

# Benchmark description

- 1 paper[1]: *Triathlon of Lightweight Block Ciphers for the IoT*
- 1 benchmarking framework (`FELICS`: `F`air `E`valuation of `LI`ghtweight `C`ryptograhic `S`ystems)
- 3 different platforms (8-bit AVR, 16-bit MSP, 32-bit ARM)
- 3 different usage scenarios
- 3 different metrics: RAM, clock cycles, code size
- 13 different block ciphers

[1] eprint 2015/209 (`https://https://eprint.iacr.org/2015/209`)

# Benchmark description

- 1 paper[1]: *Triathlon of Lightweight Block Ciphers for the IoT*
- 1 benchmarking framework (`FELICS`: `F`air `E`valuation of `LI`ghtweight `C`ryptograhic `S`ystems)
- 3 different platforms (8-bit AVR, 16-bit MSP, 32-bit ARM)
- 3 different usage scenarios
- 3 different metrics: RAM, clock cycles, code size
- 13 different block ciphers
- 90 different implementations

[1] eprint 2015/209 (`https://https://eprint.iacr.org/2015/209`)

**Results for scenario 1 - I: Encryption + Decryption (including key schedule). Encrypt 128 bytes of data using CBC mode. For each cipher, an optimal implementation on each architecture is selected.**

| | AVR | | | MSP | | | ARM | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Cipher | Code [B] | RAM [B] | Time [cyc.] | Code [B] | RAM [B] | Time [cyc.] | Code [B] | RAM [B] | Time [cyc.] | FOM |
| Speck | 1710 | 305 | 239491 | 1342 | 300 | 93239 | 792 | 356 | 19529 | 3.8 |
| Robin | 4944 | 271 | 146149 | 3170 | 238 | 76878 | 3684 | 320 | 92132 | 6.8 |
| Fantomas | 5892 | 267 | 111677 | 4164 | 234 | 57430 | 4620 | 324 | 70197 | 7.1 |
| Simon | 2494 | 380 | 390078 | 8158 | 392 | 214745 | 892 | 424 | 25863 | 7.4 |
| RC5 | 2634 | 382 | 515823 | 1952 | 378 | 482894 | 1144 | 432 | 32903 | 8.1 |
| LBlock | 3104 | 336 | 207590 | 2024 | 328 | 313349 | 2136 | 430 | 162645 | 8.7 |
| HIGHT | 2626 | 347 | 168528 | 2368 | 342 | 423221 | 2196 | 416 | 173762 | 9.5 |
| Piccolo | 2672 | 324 | 407890 | 1824 | 318 | 349423 | 1604 | 430 | 291401 | 11.4 |
| PRINCE | 5668 | 246 | 280340 | 4174 | 240 | 405552 | 4660 | 416 | 226401 | 12.4 |
| TWINE | 3628 | 407 | 384952 | 3452 | 352 | 565495 | 2464 | 442 | 257039 | 13.2 |
| AES | 26794 | 551 | 109422 | 20726 | 574 | 54075 | 15272 | 576 | 40868 | 20.3 |
| LED | 4556 | 279 | 2634419 | 7004 | 252 | 2505640 | 3732 | 358 | 692265 | 40.6 |
| PRESENT | 11226 | 596 | 3838506 | 4564 | 500 | 10983553 | 7372 | 814 | 607047 | 98.6 |

# Contributing

- Ciphers are implemented in C
- `FELICS` is open and flexible: simplifies measures and ensures fair comparison
- Everything is published on our website (`link`)

# Contributing

- Ciphers are implemented in C
- FELICS is open and flexible: simplifies measures and ensures fair comparison
- Everything is published on our website (link)

**Contributions are welcome!**

# Conclusion

⇓⇓ Click on this link ⇓⇓

https://www.cryptolux.org/index.php/Lightweight_Cryptography

⇑⇑ Click on this link ⇑⇑