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

18 August 2015

Review of Lightweight Crypto

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

Review of Lightweight Crypto

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

Let us know if you have new results!

PRESENT

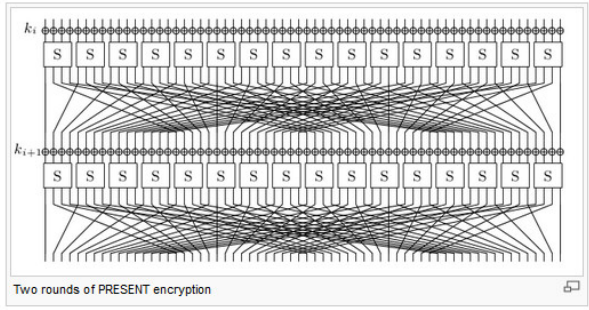
[edit]

- Article: *PRESENT: An Ultra-Lightweight Block Cipher*, CHES 07[43]
- 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[43], PRESENT-128 and its modified key-schedule are described in the appendix. This cipher has been standardized and is part of the ISO-29192[79] with CLEFIA.



Two rounds of PRESENT encryption

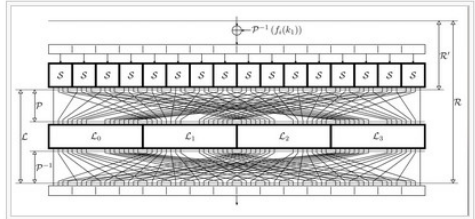
PRIDE

[edit]

- Article: *Block Ciphers -- Focus On the Linear Layer (feat. PRIDE)*, CRYPTO'14[80]
- 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

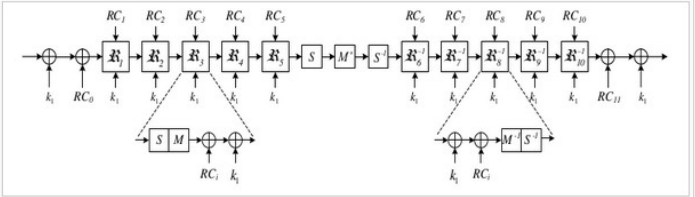
[edit]

- Article: *PRINCE – A Low-latency Block Cipher for Pervasive Computing Applications*, ASIACRYPT 12[48]
- 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 α is a constant derived from π. 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 "α-reflexivity".

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



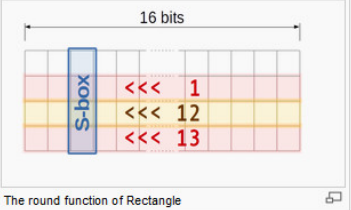
The PRINCE-core algorithm

Rectangle

[edit]

- Article: *RECTANGLE: A Bit-slice Ultra-Lightweight Block Cipher Suitable for Multiple Platforms*, eprint.iacr.org[55]
- Authors: Wentao Zhang, Zhenzhen Bao, Dongdai Lin, Vincent Rijmen, Bohan Yang, Ingrid Verbauwhede
- Target: Hardware and software

Rectangle is a substitution permutation network. Its state is represented as a 4×16 matrix. The non-linear layer consists in the parallel application of a 4-bit S-Box on the columns of the state and the linear layer consists simply in applying a fixed rotation by a different amount on each row. The key-schedule operates similarly by storing the key in a matrix which is updated in a similar fashion except that the S-Box is only applied on the first column.



The round function of Rectangle

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[12]
- Authors: Gregor Leander, Christof Paar, Axel Poschmann, and Kai Schramm
- Target: Hardware

Benchmarking Framework

FELICS – **F**air **E**valuation of **Li**ghtweight **C**ryptographic **S**ystems

- open-source software benchmarking framework
- 3 different platforms (8-bit AVR, 16-bit MSP, 32-bit ARM)
- 3 different metrics: execution time, RAM, code size
- different usage scenarios
- 100+ different implementations of block and stream ciphers!

Benchmarking Framework

FELICS – **F**air **E**valuation of **Li**ghtweight **C**ryptographic **S**ystems

- open-source software benchmarking framework
- 3 different platforms (8-bit AVR, 16-bit MSP, 32-bit ARM)
- 3 different metrics: execution time, RAM, code size
- different usage scenarios
- 100+ different implementations of block and stream ciphers!

Contributions are welcome!

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	1644	305	59612	1342	300	93239	792	356	19529	4.6
Chaskey	6356	261	102197	7014	246	37382	1776	276	5558	6.1
Simon	2304	380	82085	9398	394	162012	896	428	24019	8.1
AES	*4356*	*434*	*59085*	*3444*	*412*	*84070*	*3928*	*500*	*70905*	10.5
Fantomas	5892	267	111677	4164	234	57430	4620	324	70197	10.6
RC5	4574	378	252147	1952	378	482894	1144	432	32903	11.2
Robin	4944	271	146149	3170	238	76878	3684	320	92132	11.5
LBlock	3104	336	207590	2024	328	313349	2208	598	140595	16.1
HIGHT	2624	347	166480	2370	340	363829	2196	416	173762	18.1
PRINCE	5358	374	243396	4174	240	405552	4304	548	202445	22.6
PRESENT	*2840*	*458*	*245853*	*2230*	*454*	*201885*	*2528*	*526*	*270603*	23.5
TWINE	4236	646	297265	3796	564	393320	2464	506	255574	25.7
Piccolo	2672	324	407890	1824	318	349423	1604	430	291401	26.0
LED	5156	574	2221555	7004	252	2505640	3640	678	585216	76.3

Implementation Competition

Win Luxembourgish Chocolate/Beer!

Triathlon Competition

How do I win?

What to submit? Implementations (assembly/C) of published lightweight block ciphers

What targets? AVR, MSP, ARM

Scores Get points based on the implementation performance figures

Who gets a prize? First 3 players/teams *and* first 3 implementations

Triathlon Competition

How do I win?

What to submit? Implementations (assembly/C) of published lightweight block ciphers

What targets? AVR, MSP, ARM

Scores Get points based on the implementation performance figures

Who gets a prize? First 3 players/teams *and* first 3 implementations

First Deadline: **September 6, 2015** (before CHES 2015)

Website: https://www.cryptolux.org/index.php/FELICS_Triathlon

Conclusion

↓↓ Click on this link ↓↓

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

↑↑ Click on this link ↑↑