CUBE Cipher: A Family of Quasi-Involutive Block Ciphers Easy to Mask

Thierry Berger[1], Julien Francq[2] and **Marine Minier**[3]

[1]XLIM - Université de Limoges, France

[2]Airbus Defence & Space - CyberSecurity, France

[3]Université de Lyon, INRIA, INSA Lyon, CITI Lab, France
marine.minier@insa-lyon.fr

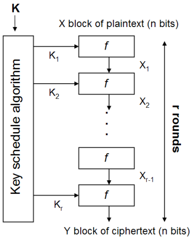
October 2015

Motivation

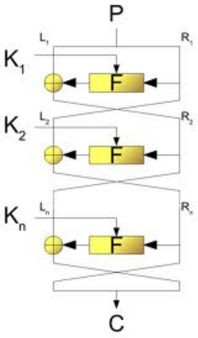
- Many lightweight block ciphers:
  - PRESENT, LED or PRINCE that are SPN
  - TWINE, LBlock, SIMON or *Piccolo* that are Feistel based constructions
- Recently, one more constraint: easy to mask by design
  - PICARO, Zorro or Fantomas and Robin
- Aim here: bring grist to the mill in this research direction
  - Using a generic CUBE representation
  - Quasi-involutive to limit the hardware footprint
  - SPN based framework

Block Ciphers

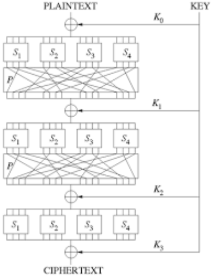
- Block cipher overview



- Feistel cipher

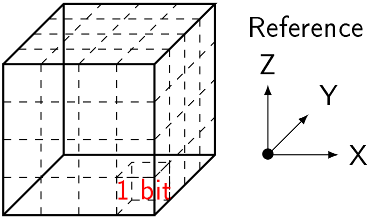


- SPN cipher



CUBE Representation

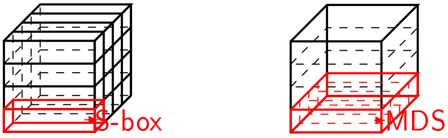
- Plaintext = a CUBE of size $n \times n \times n$
- The CUBE is fulfilled: least significant bit at position $(0, 0, 0)$ according (X, Y, Z)



Overview of CUBE family (1/2)

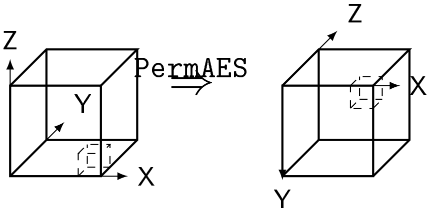
- CUBE family: 2 designs, 3 instantiations ($n = 4, 5$ or 6), r rounds followed by a final `KeyAdd` with K_r at the end

- i-th round function:
  - `KeyAdd`: A subkey addition (XOR) with K_i
  - `SbLayer`: A layer of involutive S-boxes. Apply $n \times n$ a single involutive n-bit S-box easy to mask
  - `MDSLayer`: On each plane $(0, Y, Z)$, $(1, Y, Z)$, $(2, Y, Z)$ and $(3, Y, Z)$, apply a quasi-involutive linear Feistel-MDS transformation on n words of size n bits

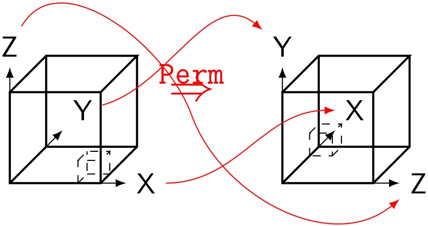


Overview of CUBE family (2/2)

- Permutation Layer: 2 different permutations
  - For CUBEAES, PermAES rotates by 90^o the reference (X, Y, Z)



  - For CUBE, Perm rotates the axes (X, Y, Z) as (Z, X, Y)



Key Schedule

- 2 possible key sizes for K: n^3 bits or $2 \times n^3$ bits
- $|K| = n^3$, subkeys computation
  - $K_0 = K$ and $K_{i+1} = K_i A \oplus (i+1)$ for $i = 0, \cdots, (r-1)$
  - A = invertible matrix of linear diffusion using a Feistel structure
  - The counter $(i+1)$ is added to the least significant bits
- $|K| = 2 \times n^3$, subkeys generation
  - $K = K_1 || K_0$ and $K_1 = K_1 \oplus 1$
  - $K_{i+2} = K_{i+1} A \oplus K_i \oplus (i+2)$ for $i = 0, \cdots, (r-2)$
  - A = same thing
- word size of A
  - For $n = 4$, matrix of size 8×8 that acts on bytes
  - For $n = 5$, matrix of size 5×5 that acts on 25-bit words
  - For $n = 6$, matrix of size 12×12 that acts on 18-bit words

Instantiation with $n = 4$: round function (1/2)

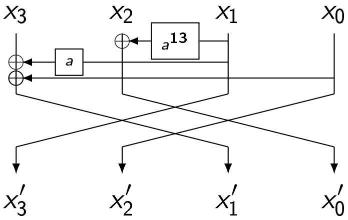
- CUBEAES and CUBE with $n = 4$:
  - 64-bit plaintext/ciphertext
  - Key length: 128 bits
  - 15 rounds. Final subkey addition with K_{15}
- `KeyAdd`: subkey addition with K_i of 64 bits
- `SbLayer`: The Noekeon S-box at nibble applied 16 times in parallel

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

Instantiation with $n = 4$: round function (2/2)

- `MDSLayer`: The 4×4 MDS matrix M acts on $\mathbb{F}_{16} = \mathbb{F}_2[X]/(X^4 + X + 1)$. $M = D^4$, D acts on nibbles

$$D = \begin{pmatrix} 0 & a^{13} & 1 & 0 \\ 1 & a & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$



- Multiplications by binary matrices of a and a^{13}:

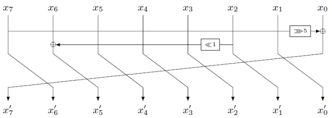
$$M_a = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix} \text{ and } M_{a^{13}} = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

- `Permutation`: CUBEAES = `PermAES`. CUBE = `Perm`

Instantiation with $n = 4$: key schedule

- derives 16 subkeys $K_0, \cdots K_{15}$ of 64 bits from the master key K of length 128 bits
- The 8×8 matrix A acts on bytes level: $A = B^3$

$$B = \begin{pmatrix} 0 & I & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & \lll 1 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I \\ I & 0 & 0 & 0 & 0 & 0 & 0 & \ggg 5 \end{pmatrix}$$



Design Rational: Cube structure

- Cube Structure: Lightweight, classical state = 64 bits
  - Several S-boxes apply in parallel
  - Linear layer mix the outputs. Most efficient way = MDS matrix on subblocks
  - Easy to construct MDS matrices but costly for implementation
  - BUT if nibbles, the MDS matrix limited to 4 subblocks
  - Cube structure simplifies mixing !

  - Not new: KECCAK, PRESENT
  - Keep the PRESENT cube structure BUT include a diffusion layer to discard statistical properties and "linear hulls" of PRESENT

$\Rightarrow$ Our choice: elementary operations on smaller words improve the latency

Design Rational: MDS Diffusion

- MDS Diffusion in Cube Structure
  - `Permutation` well chosen ⇒ MDS diffusion activates n n-bit words in a plane and all the planes activated just after ⇒ our design choice for CUBEAES
  - For CUBE keep the permutation of PRESENT and discard "bad PRESENT properties"
- Recursive and Quasi-Involutive MDS: 2 requirements
  - Quasi-involutive
  - Use elementary operations
- MDS diffusion performed with iterative approach and "generalized Feistel schemes"
- as done in PHOTON except that D is not a companion matrix to reduce the fan-in

Design Rational: S-box and Key Schedule

- Involutive S-box Suitable for Masking
  - involutive for a quasi-involutive cipher
  - For $n = 4$, involutive Noekeon S-box
    - Optimal differential and linear probabilities
    - Algebraic degree equal to 3
    - Simple implementation: 7 XOR, 2 AND and 2 NOR
  - $\Rightarrow$ Easy to mask: 4 non-linear operations. Quadratic in the number of shares for the 4 non linear operations and linear in the number of shares for the 7 linear operations

- Key Schedule
  - Good mixing to maximize the master key entropy in each subkey
  - Low hardware implementation cost, linear and involutive
  - K could be recovered from any pair of subkeys

$\Rightarrow$ matrix A, invertible binary matrix, follows these rules

Security Analysis: Diff./Lin.

- Focused on $n = 4$ with key of 128 bits
- Differential / Linear Cryptanalysis: $DP = 2^{-2}$ and $LP = 2^{-1}$.
  Minimal number of active S-boxes for CUBEAES and CUBE

	Round	1	2	3	4	5	6	7	8	9	10
CUBEAES	AS_D	1	5	9	25	26	30	34	50	51	55
	AS_L	1	5	9	25	26	30	34	50	51	55
CUBE	AS_D	1	5	9	13	20	21	25	29	33	40
	AS_L	1	5	9	12	19	20	24	28	31	38

- CUBEAES = maximum as the AES
- CUBE: branch and bound method ⇒ no elementary differential/linear paths

⇒ best differential/linear cryptanalysis: 6 rounds of CUBEAES, 8 rounds of CUBE

Security Analysis: Structural Attacks

- Impossible Differential
  - For CUBEAES, best ID attack on 7 rounds using a 4 rounds ID
  - For CUBE, best ID attack on 8 rounds using a 5 rounds ID
- Integral Attack
  - For CUBEAES, attack on 6 rounds with a complexity 2^{75}
  - For CUBE, attack on 7 rounds with same complexity
- Related Key and Chosen Key Attacks For CUBE and CUBEAES, best related key attack gains 2 rounds at the beginning. Branch and bound algorithm $\Rightarrow$ no simple way to cancel differences
- Resistance to Side Channel Analysis S-box chosen to offer resistance to side channel analysis at a reasonable cost

Security Analysis: Conclusion

Conjecture

no attack against 8 rounds of CUBEAES and against 9 rounds of CUBE in the single key settings

Security Analysis: Conclusion

Conjecture

no attack against 8 rounds of CUBEAES and against 9 rounds of CUBE in the single key settings

Conjecture

no attack against 11 rounds of CUBEAES and against 12 rounds of CUBE in the related, known and chosen key settings

Implementation Results

- Theoretical Implementation Results: round-wise implementation of CUBE cipher 2656 GEs
- Implementation Results and Comparisons: Implementation in VHDL clock frequency 100kHz, 2536 GEs, simulated power of 0.663 μW

	Key Size	Block Size	Lat. (cycles)	Area (GEs)	Logic Process
mCrypton	128	64	13	4108	0.13μm (theo.)
HIGHT	128	64	34	3048	0.25μm
TWINE-128	128	64	36	2285	90nm
Piccolo-128	128	64	27	1938	0.13μm (theo.)
PRESENT-128	128	64	32	1886	0.18μm (only enc.)
CUBE cipher	128	64	25	2536	65 nm LP

- Power Comparison: CUBE cipher is fast. Advantageous in terms of latency and energy
- CUBE cipher compares reasonably well
- The price for a secure Key Schedule and to avoid undesirable properties of PRESENT is limited

Conclusion

- 2 involutive families of ligthweight block ciphers easy to mask
- with reasonable hardware cost
- Involutive means a near-free implementation of the decryption process
- the MDS layer added to CUBE prevents the bad behaviors of the PRESENT

Thank You for Your Attention!

Any questions ?

Bibliography

- B. Collard and F.-X. Standaert. A Statistical Saturation Attack against the Block Cipher PRESENT. In Topics in Cryptology - CT-RSA 2009, LNCS 5473, pages 195-210, 2009.
- B. Collard and F.-X. Standaert. Multi-trail statistical saturation attacks. In Applied Cryptography and Network Security - ACNS 2010, LNCS 6123, pages 123-138, 2010.
- J. Guo, T. Peyrin, A. Poschmann, and M. J. B. Robshaw. The LED Block Cipher. In Cryptographic Hardware and Embedded Systems - CHES 2011, LNCS 6917, pages 326-341. Springer, 2011.
- Y. Ishai, A. Sahai, and D. Wagner. Private circuits: Securing hardware against probing attacks. In Advances in Cryptology - CRYPTO 2003, LNCS 2729, pages 463-481, 2003.

Other Instantiations: $n = 5$

- Number of rounds $r = 17$
- 5-bit to 5-bit involutive S-box ($DP = 2^{-2.41}$, $LP = 2^{-2}$, algebraic degree of 4, a non linearity equal to 3)

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
$S(x)$	1F	1D	1A	1B	12	1E	13	E	F	18	16	C	B	10	7	8	D	19	4	6	15	14	A	1C	9	11	2	3	17	1	5	0

- Matrices M and D of `MDSLayer`, $M = D^5$:

$$D = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & a^{30} & a & a & a^{30} \end{pmatrix}$$

- Matrices $A = B^5$ and B of the key schedule acts on 5 blocks of 25 bits:

$$B = \begin{pmatrix} 0 & I & 0 & 0 & 0 \\ 0 & \lll 9 & I & 0 & 0 \\ 0 & 0 & 0 & I & 0 \\ \gg 1 & 0 & 0 & 0 & I \\ I & 0 & 0 & 0 & 0 \end{pmatrix}$$

Other Instantiations: $n = 6$

- Number of rounds $r = 19$
- 6-bit to 6-bit involutive S-box ($DP = 2^{-3.41}$, $LP = 2^{-2.41}$, algebraic degree of 5, non linearity equal to 5)

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S(x)$	17	13	35	A	C	26	B	23	1C	31	3	6	4	3D	3E	20

x	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
$S(x)$	16	18	14	1	12	29	10	0	11	2F	25	39	8	33	36	2E

x	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F
$S(x)$	F	3A	37	7	2B	1A	5	38	3B	15	2C	24	2A	3C	1F	19

x	30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
$S(x)$	32	9	30	1D	3F	2	1E	22	27	1B	21	28	2D	D	E	34

- Matrices $M = D^6$ of the `MDSLayer`:

$$D = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & a^{61} & a^{49} & a & a^{49} \end{pmatrix}$$

- Matrices $A = B^9$ and B in the key schedule acts on 12 blocks of 18 bits

Classical Method for Efficient Boolean Masking

Algorithme 1 : Non linear operation performed on two masked secrets x and y

Data : Shares $(x_i)_i$ and $(y_i)_i$ satisfying $\oplus_i x_i = x$ and $\oplus_i y_i = y$.
Result : Shares $(w_i)_i$ satisfying $\oplus_i w_i = x \cdot y$.

for *i from 0 to d* **do**
    for *j from $i+1$ to d* **do**
        $r_{i,j} \in_R \mathbb{F}$;
        $r_{j,i} \leftarrow (r_{i,j} \oplus x_i \cdot y_j) \oplus x_j \cdot y_i$;

for *i from 0 to d* **do**
    $w_i \leftarrow x_i \cdot y_i$;
    for *j from 0 to d, $j \neq i$* **do**
        $w_i \leftarrow w_i \cdot r_{i,j}$;