Low Area Hardware Implementations of CLOC, SILC and AES-OTR

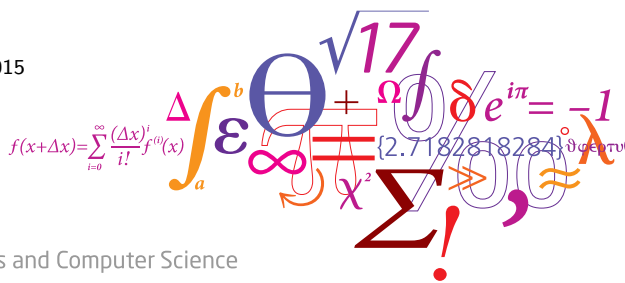
Subhadeep Banik[1], Andrey Bogdanov[1], Kazuhiko Minematsu[2]

[1] DTU Compute, Technical University of Denmark, Lyngby

[2] NEC Corporation, Japan

Directions in Authenticated Ciphers 2015

Singapore



DTU Compute

Department of Applied Mathematics and Computer Science

Outline

- Preliminaries
- CLOC/SILC
- AES-OTR
- Conclusion

Lightweight Implementation of AES

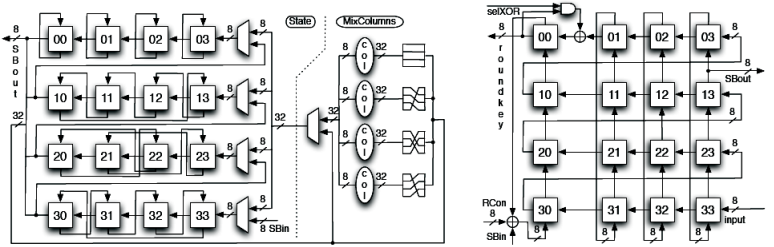


Figure : Lightweight AES

Moradi et al. EUROCRYPT 2011

- 8-bit Serial Design.
- Area: 2430 Gate Equivalents (UMC 180nm process).

Lightweight Implementation of AES

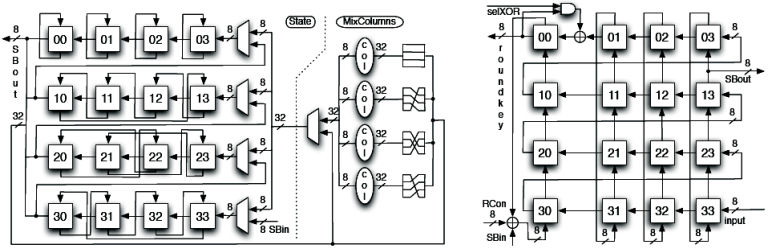


Figure : Lightweight AES

Moradi et al. EUROCRYPT 2011

- Encryption cycle of 226 rounds.
- Area Optimization: 21 cycle LFSR + Canright S-Box.

Lightweight Implementation of AES

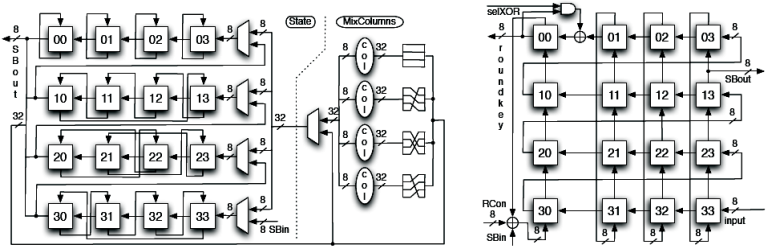


Figure : Lightweight AES

Moradi et al. EUROCRYPT 2011

- Row major interpretation of byte ordering.
- Encryption only core.

Lightweight Implementation of CLOC, SILC and AES-OTR

CLOC/SILC

- Designed to take advantage of 8-bit serial implementation.
- Additional functions are byte-oriented.
- Low area implementation possible.

AES-OTR

- Encryption only + 1 Block Cipher call per plaintext block.
- Additional function is a field multiplication over $GF(2^{128})$.
- Requires slightly higher area.

Low area Implementation I

Aggressive and Conservative Designs

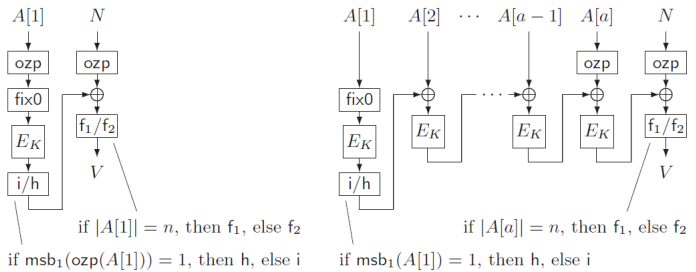
- Aggressive — Commonly used assumptions to reduce area.
  - Associated data is the empty.
  - Integral messages only (limited to 256 blocks).
  - Re-import certain intermediate outputs.
- Conservative — Data size may be fractional.
  - Associated data limited to one block.
  - Messages need not be integral (limited to 256 blocks).
  - No re-import of intermediate outputs.
  - All intermediate results stored in separate registers.

Overview of Results

#	Mode	Design	Area(μm^2)	Area (in GE)
1	CLOC	Aggressive	13672.8	3107
		Conservative	18966.5	4311
2	SILC	Aggressive	13678.3	3109
		Conservative	18100.0	4114
3	AES-OTR	Aggressive	20762.2	4719
		Conservative	29795.4	6772

Table : Areas for the CLOC, SILC, AES-OTR modes

CLOC

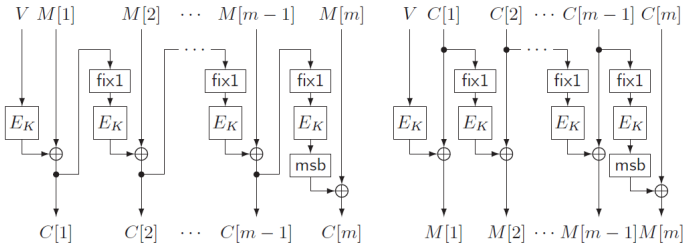


(a) $V \leftarrow$ HASH(N, A) for $0 \leq |A| \leq n$ (left) and $|A| \geq n + 1$ (right)

Compact Low Overhead CFB

- Iwata et. al. in FSE 2014.
- Three submodules: HASH, ENC, PRF.

CLOC

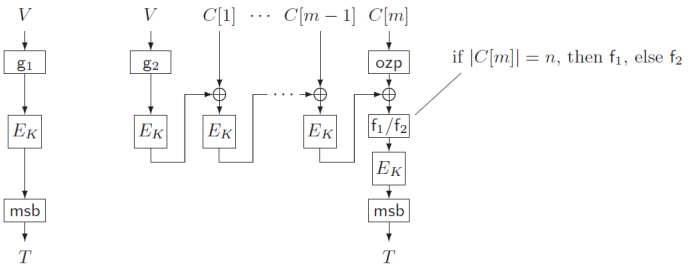


(b) $C \leftarrow \mathsf{ENC}_K(V, M)$ for $|M| \geq 1$ (left), and $\mathsf{DEC}_K(V, C)$ for $|C| \geq 1$ (right)

Compact Low Overhead CFB

- Iwata et. al. in FSE 2014.
- Three submodules: HASH, ENC, PRF.

CLOC



(c) $V \leftarrow \mathrm{PRF}(V, C)$ for $|C| = 0$ (left) and $|C| \geq 1$ (right)

Compact Low Overhead CFB

- Iwata et. al. in FSE 2014.
- Three submodules: HASH, ENC, PRF.

Tweaking Functions in CLOC

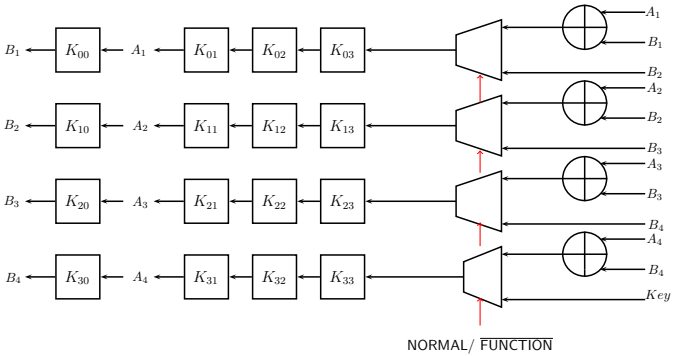
Tweaking Functions over $\{0,1\}^{128} \to \{0,1\}^{128}$

- fix1: Set MSB to 1.
- fix0: Set MSB to 0.
- i: Identity Function.
- f_2: Shift register like column oriented function

$$\mathsf{f}_2(X_1, X_2, X_3, X_4) = (X_2, X_3, X_4, X_1 \oplus X_2)$$

- h: Given as $\mathsf{f}_2{}^4$, f_1: Given as $\mathsf{f}_2{}^8$, g_1: Given as $\mathsf{f}_2{}^2$.

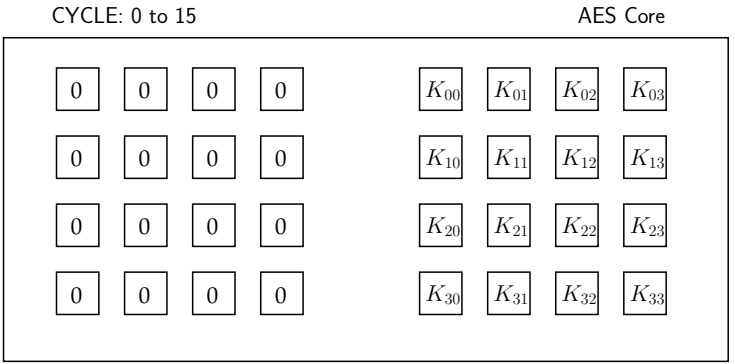
Tweaking Functions in CLOC



Implementing f_2 on Key register

- AES core hase byte wise arrangement of Key register.
- Two modes : NORMAL and FUNCTION
- Any function can be implemented by running the circuit in FUNCTION mode appropriate number of times.

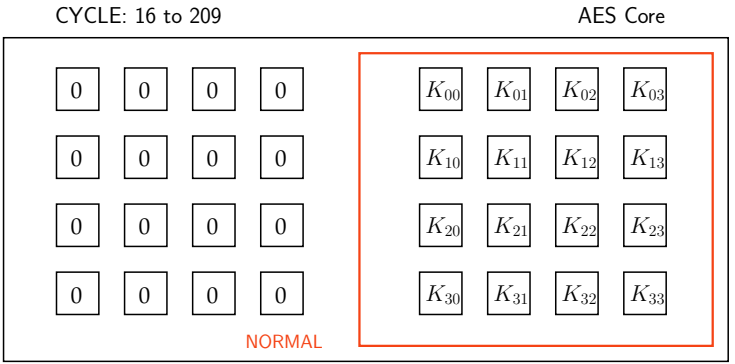
Example 1: Calculating V



$V = \mathsf{f}_2(N \oplus \mathsf{h}(E_K(0)))$

- Drive **0** and K into state and key register.

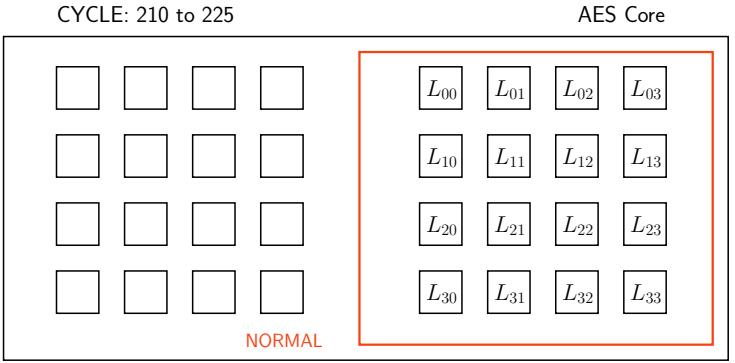
Example 1: Calculating V



$V = \mathrm{f}_2(N \oplus \mathrm{h}(E_K(0)))$

- Perform Encryption.

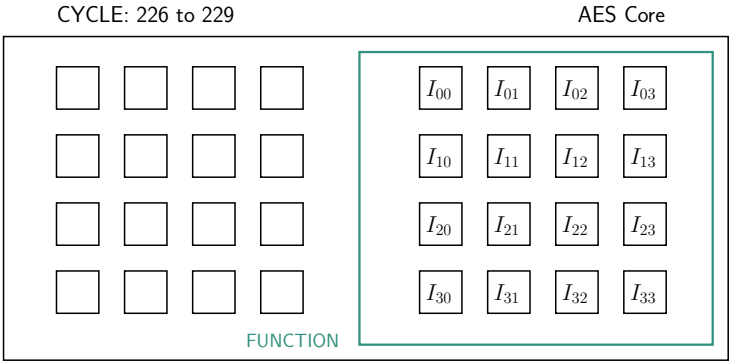
Example 1: Calculating V



$V = \mathsf{f}_2(N \oplus \mathsf{h}(E_K(0)))$

- Drive $L = E_K(\mathbf{0})$ from state to key register.

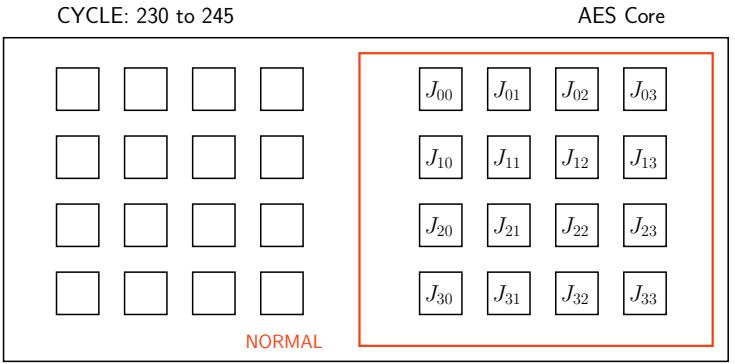
Example 1: Calculating V



$V = \mathsf{f}_2(N \oplus \mathsf{h}(E_K(0)))$

- Run key register in FUNCTION mode for 4 cycles to get $I = \mathsf{h}(E_K(\mathbf{0})) = \mathsf{h}(L)$.

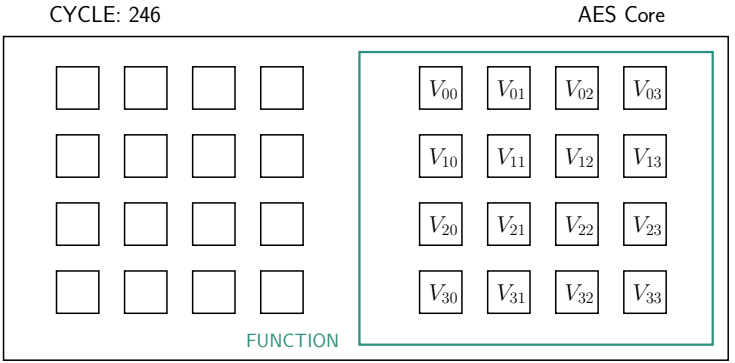
Example 1: Calculating V



$V = \mathsf{f}_2(N \oplus \mathsf{h}(E_K(0)))$ (Denote $J = N \oplus \mathsf{h}(E_K(0))$)

- Take M out of key reg, add with Nonce byte by byte and drive back into key reg.

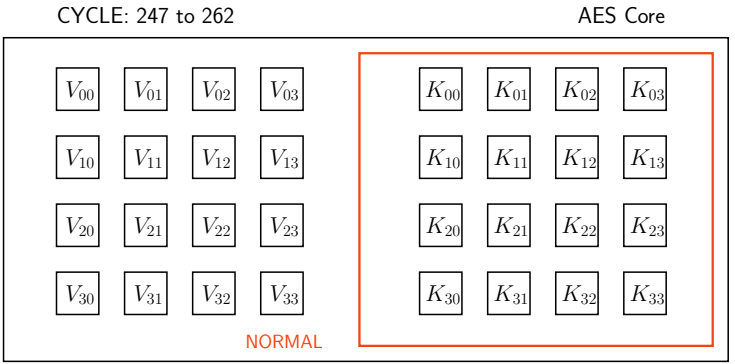
Example 1: Calculating V



$V = \mathsf{f}_2(N \oplus \mathsf{h}(E_K(0)))$ (Denote $J = N \oplus \mathsf{h}(E_K(0))$)

- Run key reg in FUNCTION mode for 1 cycle to compute $V = \mathsf{f}_2(N \oplus \mathsf{h}(E_K(0)))$

Example 1: Calculating V



$V = \mathrm{f}_2(N \oplus \mathrm{h}(E_K(0)))$ (Denote $J = N \oplus \mathrm{h}(E_K(0))$)

- Push V from key to state reg, and K back to key reg for next encryption round.

Area Shares (STM 90nm process)

#	Component	Area (in GE)	Percentage
1	AES Core	2730	86.2
2	Four 8 bit Muxes	80	2.6
3	Control System	200	8.0
4	Length Counter	97	3.2
	TOTAL	3107	100

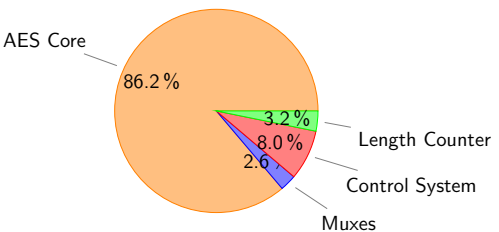
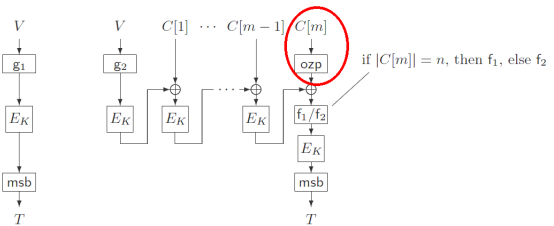


Figure : Percentage area shares

CLOC: The Conservative Design

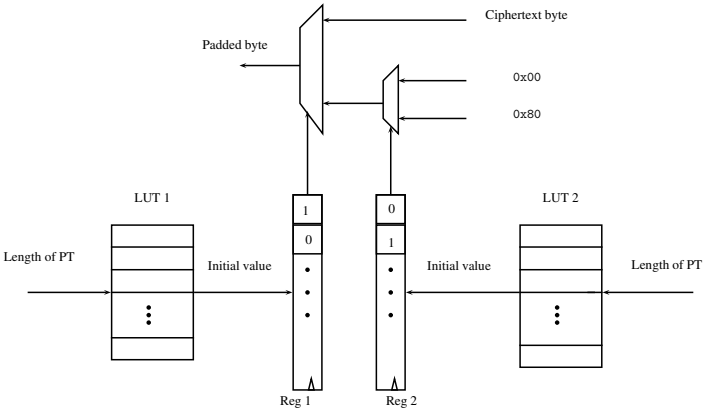
- V input to both ENC, PRF modules: one extra register required.
- Logic for last block padding for fractional ciphertext.



Example

• Last block CT	0x ab2345
• After padding	0x ab234580 00000000 00000000 00000000
• Row major arrangement	0x abxxxxxx 23xxxxxx 45xxxxxx xxxxxxxx
• We need	0x ab000000 23000000 45000000 80000000

CLOC: The Conservative Design



Example

• Last block CT	0x ab2345
• After padding	0x ab234580 00000000 00000000 00000000
• Row major arrangement	0x abxxxxxx 23xxxxxx 45xxxxxx xxxxxxxx
• We need	0x ab000000 23000000 45000000 80000000

Area Shares (STM 90nm process): Conservative Design

#	Component	Area (in GE)	Percantage
1	AES Core	2730	63.2
2	Four 8-bit Muxes	100	2.3
3	Control System	250	5.7
4	Length Counter	100	2.3
5	One additional Register	750	17.2
6	Padding Logic	381	9.2
	TOTAL	4311	100

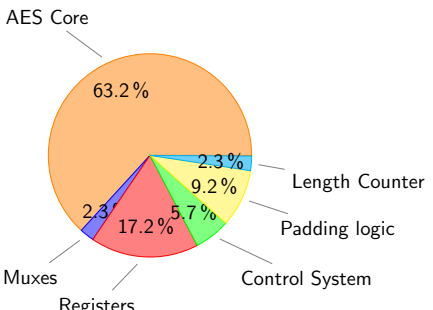
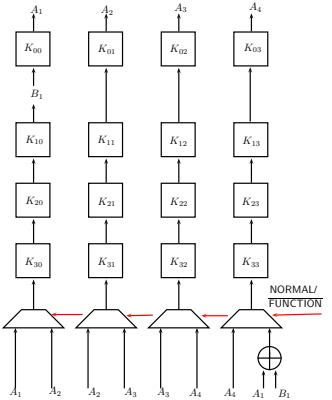


Figure : Percentage area shares

SILC

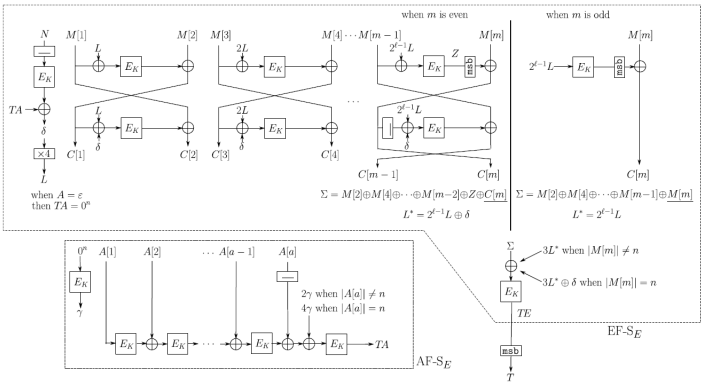


SILC

- Design identical to CLOC.
- Only difference is in the tweaking function:

$$\mathrm{g}(X_1, X_2, X_3, \ldots, X_{16}) = (X_2, X_3, X_4, \ldots, X_{16}, X_1 \oplus X_2)$$

AES-OTR



AES-OTR

- Proposed by Minematsu.
- Tweaking function: Multiplication in $GF(2^{128}) \Rightarrow$ Larger Area.

AES-OTR

AES-OTR

- One register for running $L, 2L, 2^2L \cdots$
- Another for Calculating $\Sigma = M_2 \oplus M_4 \oplus \cdots$
- Multiply by 3 circuit can be avoided $\Rightarrow$ save 300 GE !!!
  - Before Tag calculation update $\Sigma = \Sigma \oplus L^*$ (byte by byte, 8-bit xor)
  - Calculate $2L^*$ in the L register.
  - Add the 2 registers byte by byte $\Sigma \oplus L^* \oplus 2L^* = \Sigma \oplus 3L^*$

Area Shares (STM 90nm process): AES-OTR (Aggressive)

#	Component	Area (in GE)	Percentage
1	AES Core	2520	54.0
2	Eight 8-bit Muxes	200	4.3
3	Control System	400	7.5
4	Length Counter	100	2.1
5	Two additional Registers	1500	32.1
	TOTAL	4720	100

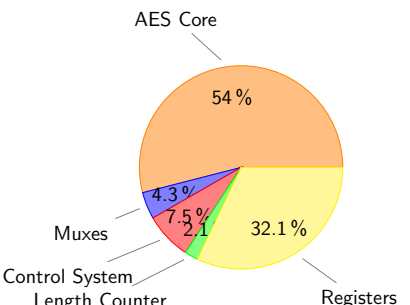


Figure : Percentage area shares

Area Shares (STM 90nm process): AES-OTR (Conservative)

#	Component	Area (in GE)	Percentage
1	AES Core	2522	37.2
2	Twelve 8-bit Muxes	300	4.4
3	Control System	450	6.7
4	Length Counter	100	1.5
5	Four additional Registers	3000	44.3
6	Padding Logic	400	5.9
	TOTAL	6772	100

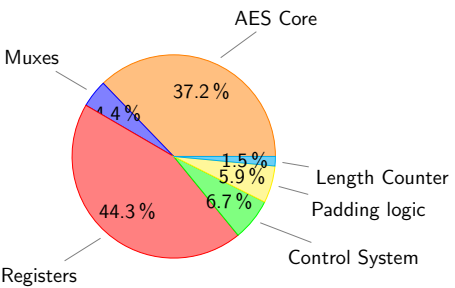


Figure : Percentage area shares

Overview of Results

#	Mode	Design	Area(μm^2)	Area (in GE)
1	CLOC	Aggressive	13672.8	3107
		Conservative	18966.5	4311
2	SILC	Aggressive	13678.3	3109
		Conservative	18100.0	4114
3	AES-OTR	Aggressive	20762.2	4719
		Conservative	29795.4	6772

Table : Areas for the CLOC, SILC, AES-OTR modes

Closing Thoughts

- Some practical diffficulties arise when using 8-bit serial AES circuit.
- None that can't be solved though.
- Lightweight implementation of modes possible !!!


Technical University of Denmark
DTU
THANK YOU