Some integral properties of Rijndael

Marine Minier
CITI Laboratory
INSA de Lyon

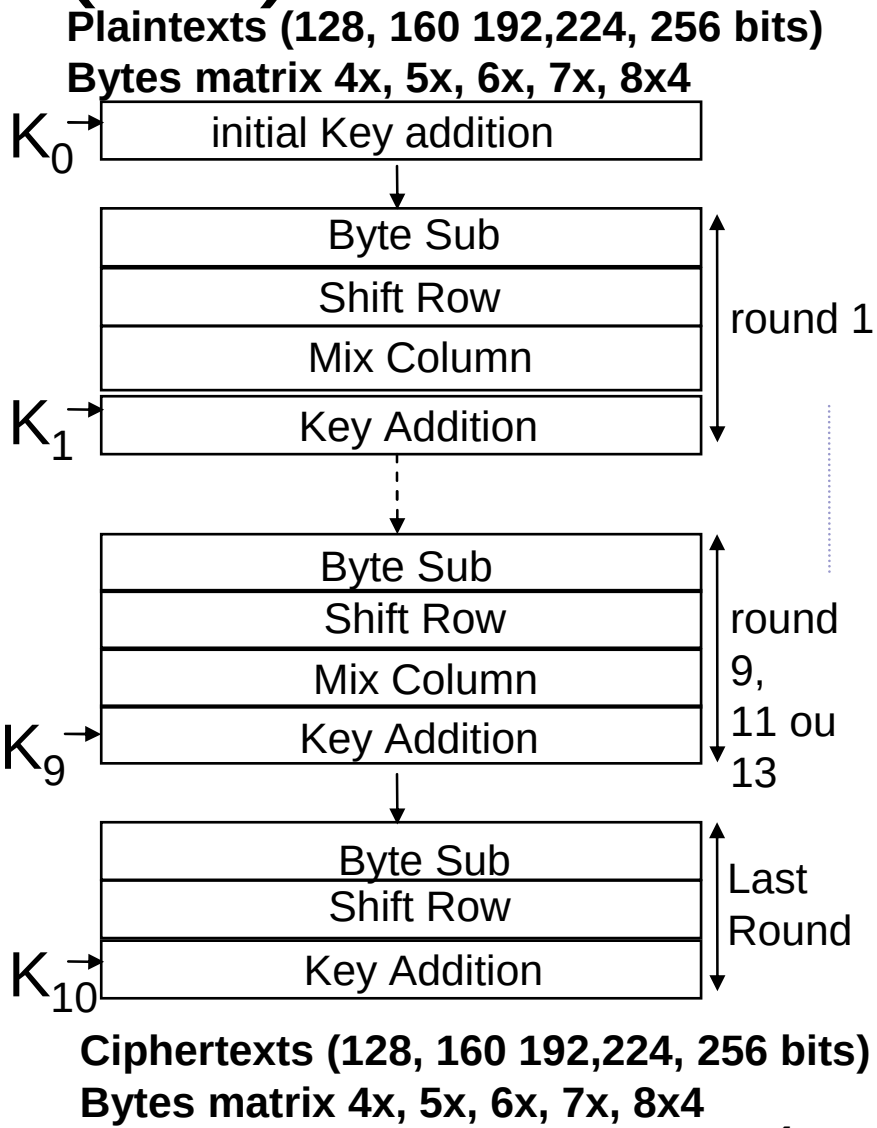
Guideline

- Description of the AES and of its little brothers
- Integral properties of the AES
- Integral properties of the different Rijndael versions
- Deduced distinguishers
  - With unknown keys
  - With known keys
- LANE
- Conclusion

The AES and its brothers

AES and Rijndael (1/3)

- Rijndael, created by J. Daemen and V. Rijmen, AES new standard
  - Iterative block ciphers with a parallel structure.
  - **blocks sizes:**
    128, 160, 192, 224 or 256 bits.
  - **Key sizes**:
    128, 192 or 256 bits.
  - The number of rounds vary between 10 and 14 according to the blocks sizes and the key sizes.



The AES (2/3): Round function (1/2)

✶ **Byte Substitution**

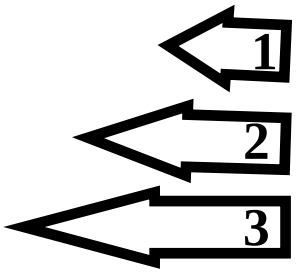
a_{00}	a_{01}	a_{02}	a_{03}
a_{10}	a_{11}	a_{12}	a_{13}
a_{20}	a_{21}	a_{22}	a_{23}
a_{30}	a_{31}	a_{32}	a_{33}

(8x8 S-box S) ↓

$S(a_{00})$	$S(a_{01})$	$S(a_{01})$	$S(a_{00})$
$S(a_{13})$	$S(a_{12})$	$S(a_{11})$	$S(a_{10})$
$S(a_{23})$	$S(a_{22})$	$S(a_{21})$	$S(a_{20})$
$S(a_{33})$	$S(a_{32})$	$S(a_{31})$	$S(a_{30})$

✸ **Shift Row**

a_{00}	a_{01}	a_{02}	a_{03}
a_{10}	a_{11}	a_{12}	a_{13}
a_{20}	a_{21}	a_{22}	a_{23}
a_{30}	a_{31}	a_{32}	a_{33}

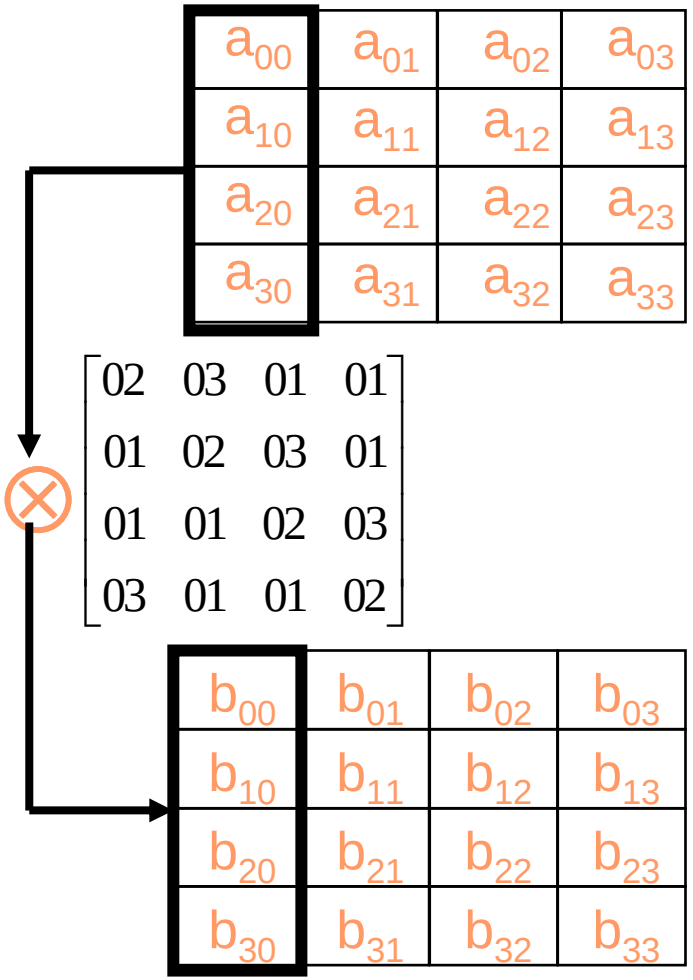


↓

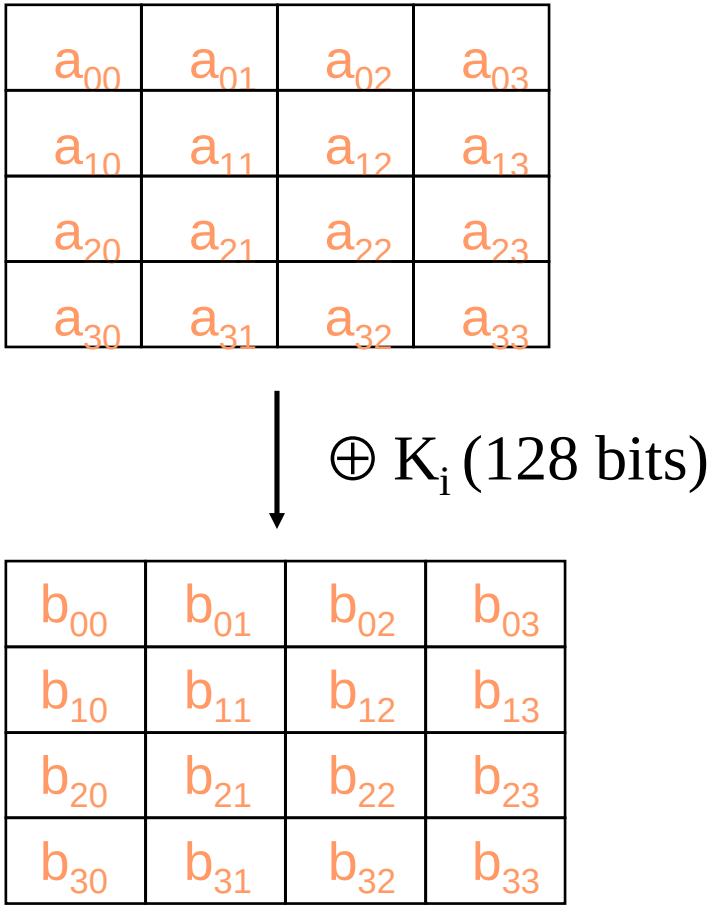
a_{00}	a_{01}	a_{02}	a_{03}
a_{11}	a_{12}	a_{13}	a_{10}
a_{22}	a_{23}	a_{20}	a_{21}
a_{32}	a_{30}	a_{33}	a_{31}

The AES (3/3): Round function (2/2)

Mix Column

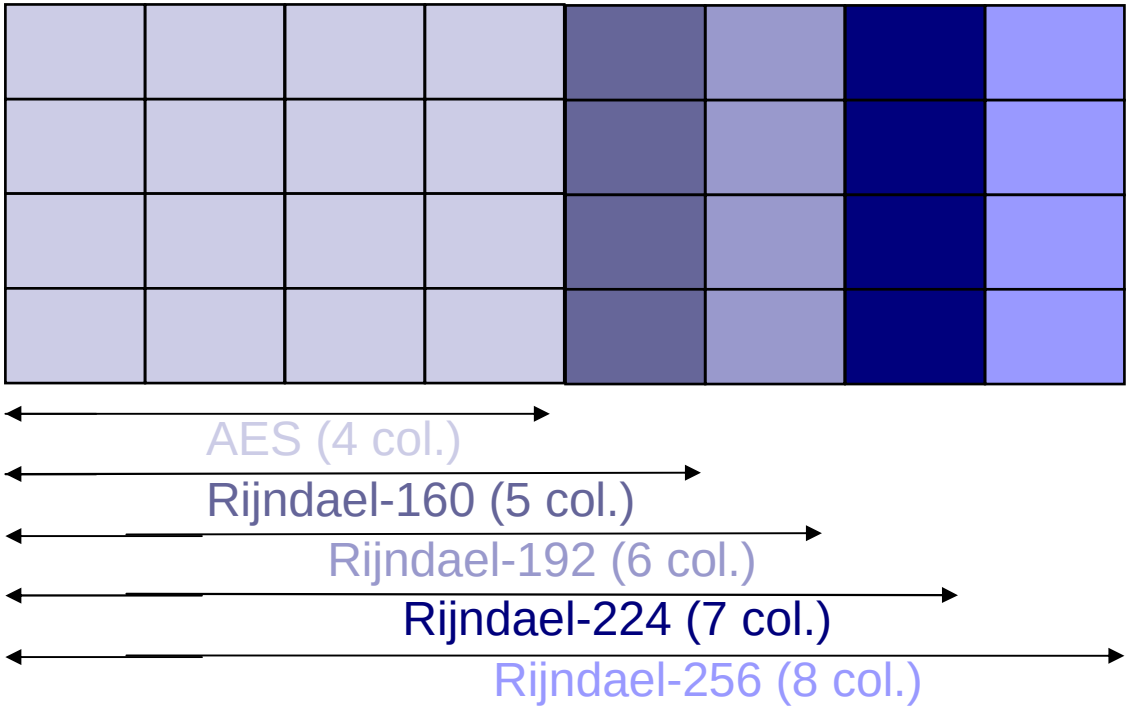


Key Addition



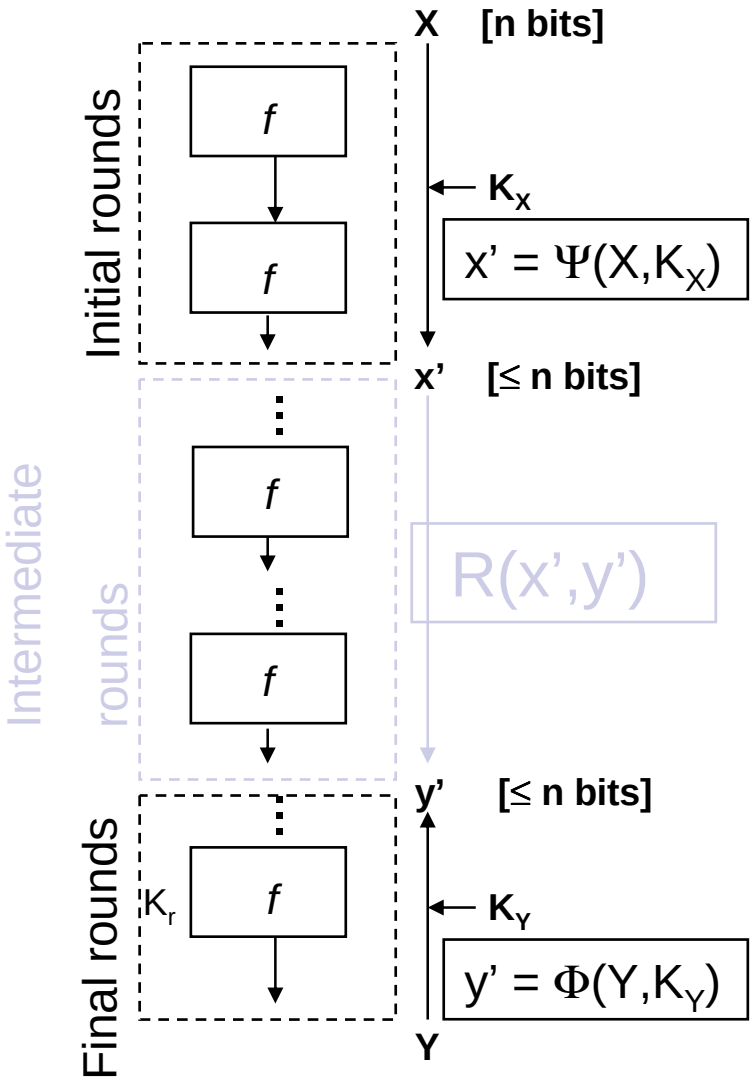
Rijndael: main differences

- Change:
  - nb of rounds
  - ShiftRows



	AES	Rijndael-160	Rijndael-192	Rijndael-224	Rijndael-256
ShiftRows	(1,2,3)	(1,2,3)	(1,2,3)	(1,2,4)	(1,3,4)
Nb rounds (Nk=128)	10	11	12	13	14
Nb rounds (Nk=192)	12	12	12	13	14
Nb rounds (Nk=256)	14	14	14	14	14

General principle of cryptanalysis



- **Distinguisher A:**
  To find a relation **R(x',y')** on intermediate states which has a probability *p* of happening as far as possible from the uniform probability *p*:*

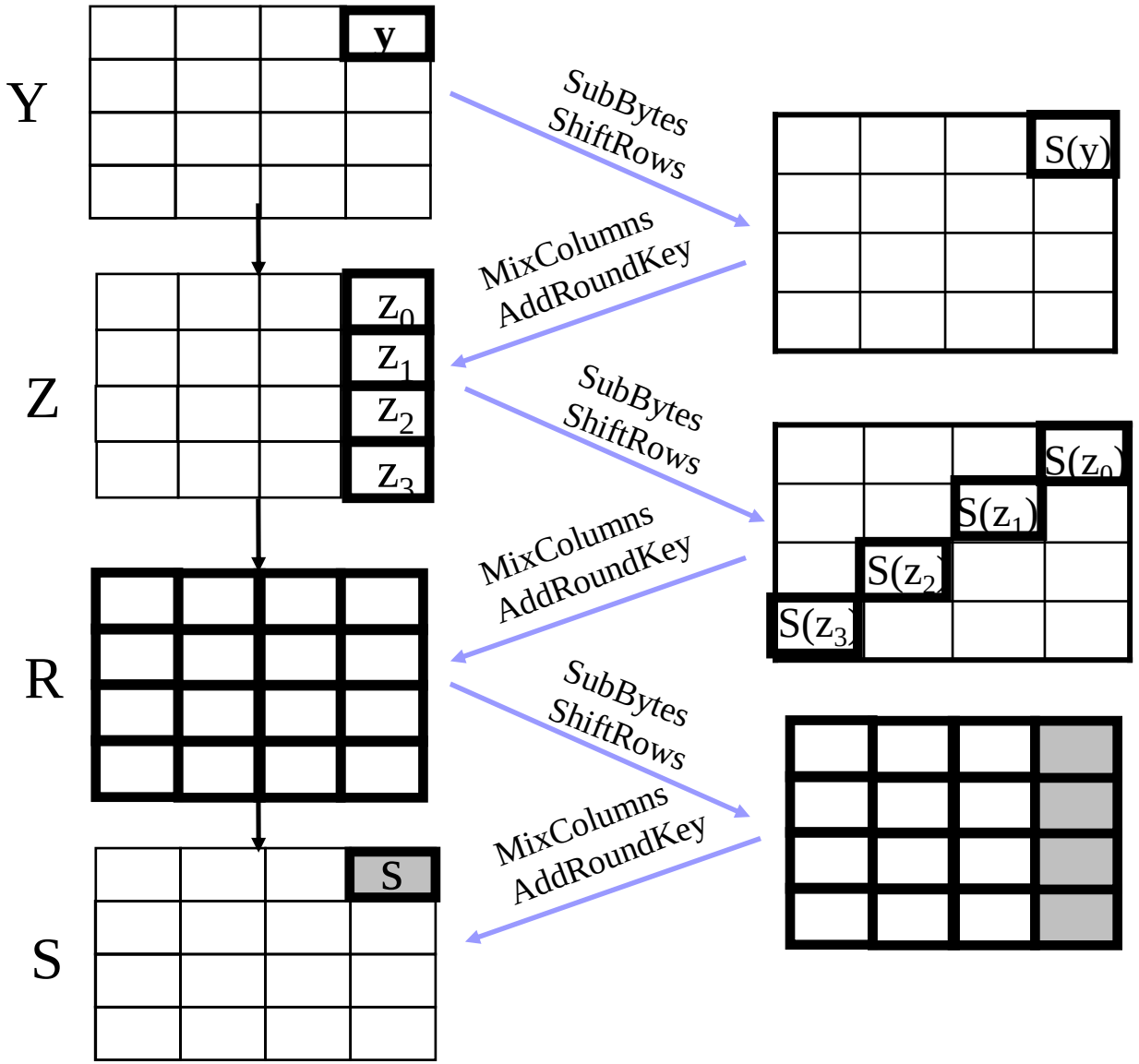
  $$Pr[A]=Adv(A)=|p-p^*|$$

- **Test over the keys** sur (K_X, K_Y)

Integral properties

Integral property of the AES (1/2)

- **byte y = 0…255**
- other bytes = **constants**



$$\bigoplus_{y=0}^{255} s(y) = 0$$

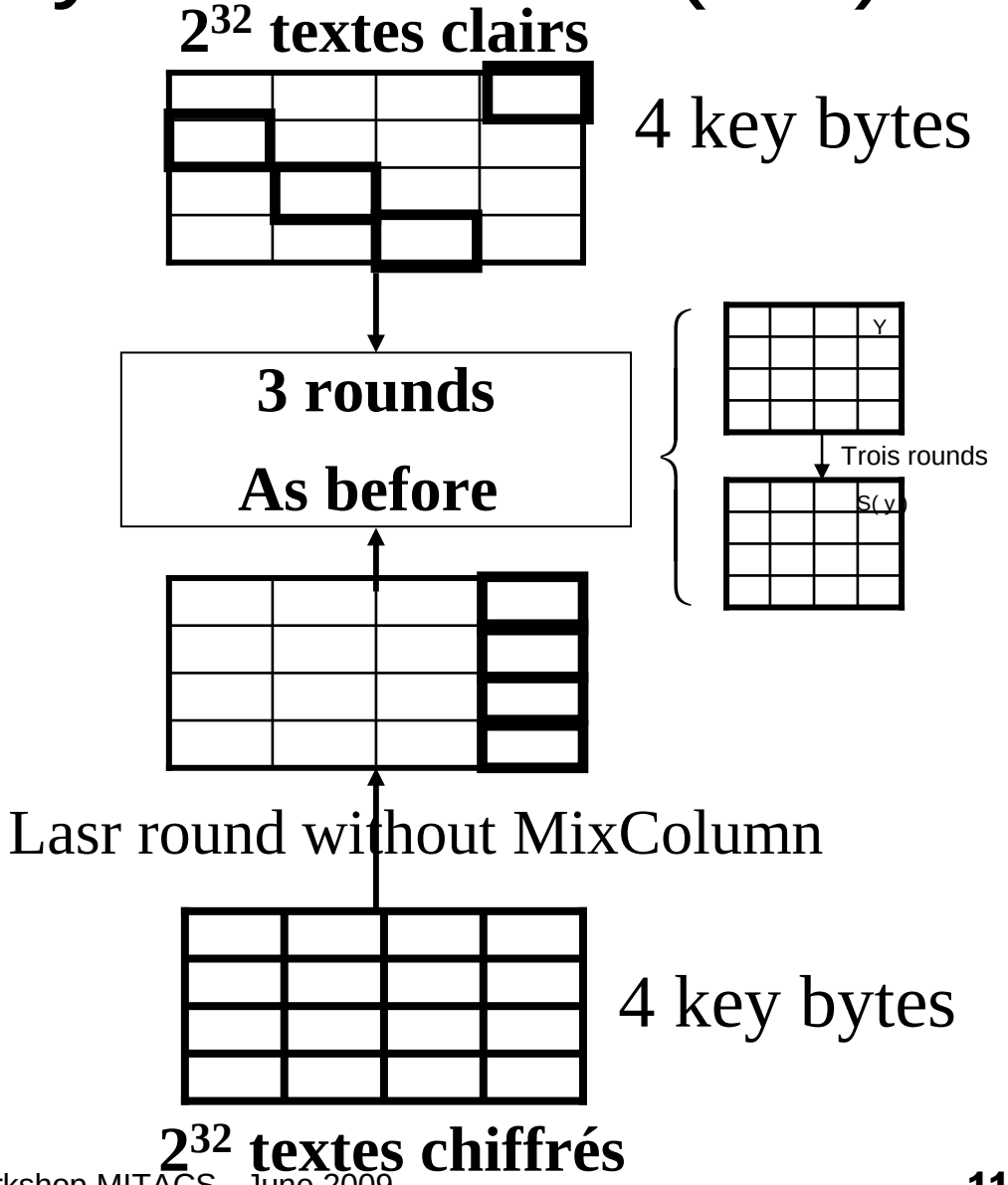
Integral property of the AES (2/2)

- **On 6 rounds:**
- **For each 9 bytes of keys:**
- **Test if:**

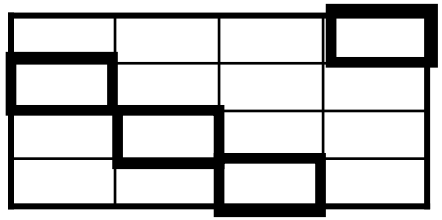
$$\bigoplus_{y=0}^{255} s(y) =? \ 0$$

Good keys pass the test.

- **Take care of false alarms.**



Complexity of integral attacks

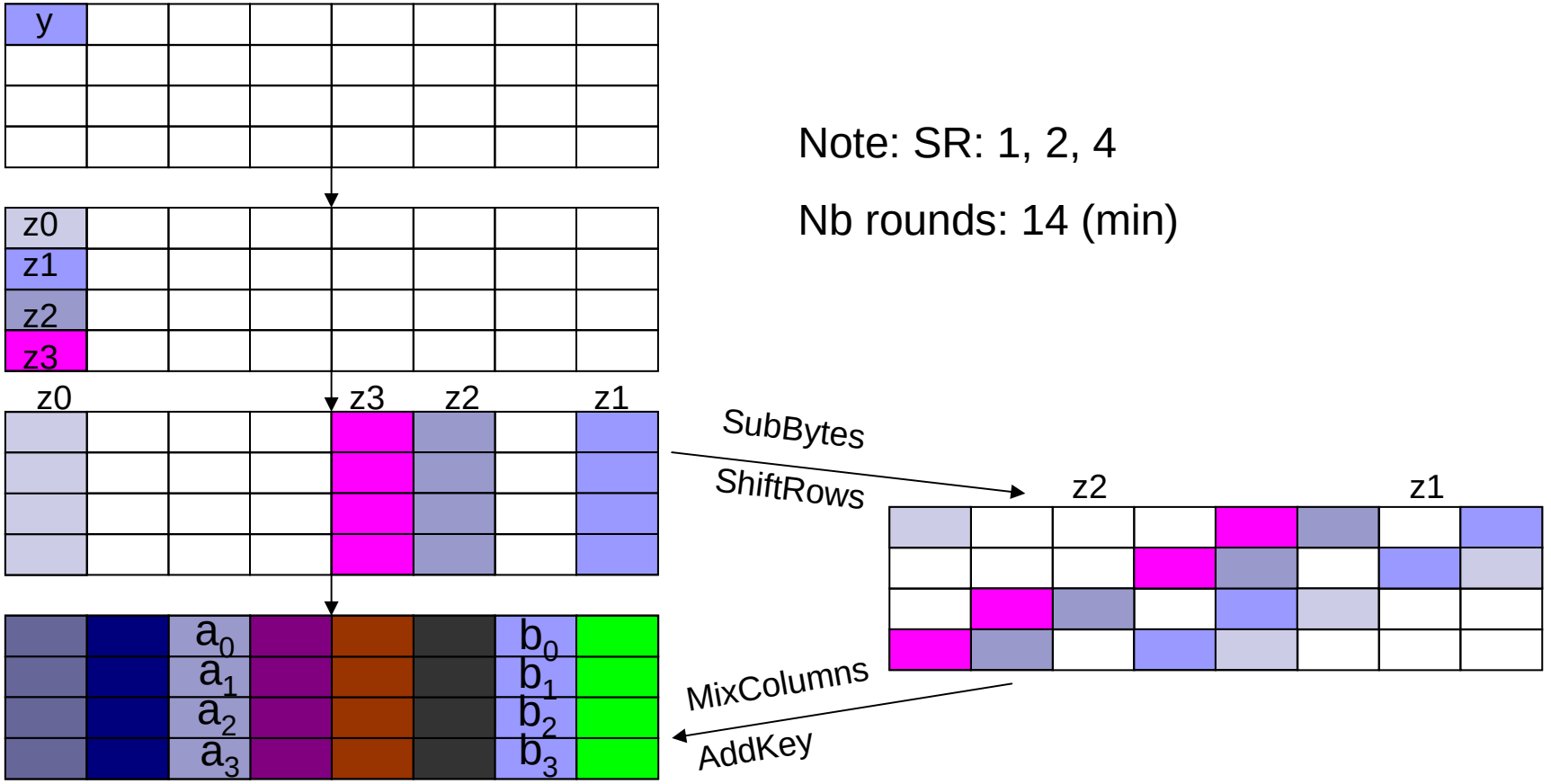


- Improvement by Ferguson:
  - Sum over the 2^{32} values
  - => Complexity for 6 rounds
    - Nb plaintexts = **$6*2^{32}$**
    - Complexity = 2^{46} using partial sum techniques
  - For **7 rounds:**
    - **Nb plaintexts = $2^{128} - 2^{119}$ (with herd technique)**
    - **Complexity = 2^{120} cipher operations**

For Rijndael

- The same kind of properties
- But, due to the slower diffusion, => more rounds and better extensions

Rijndael-256: first remark



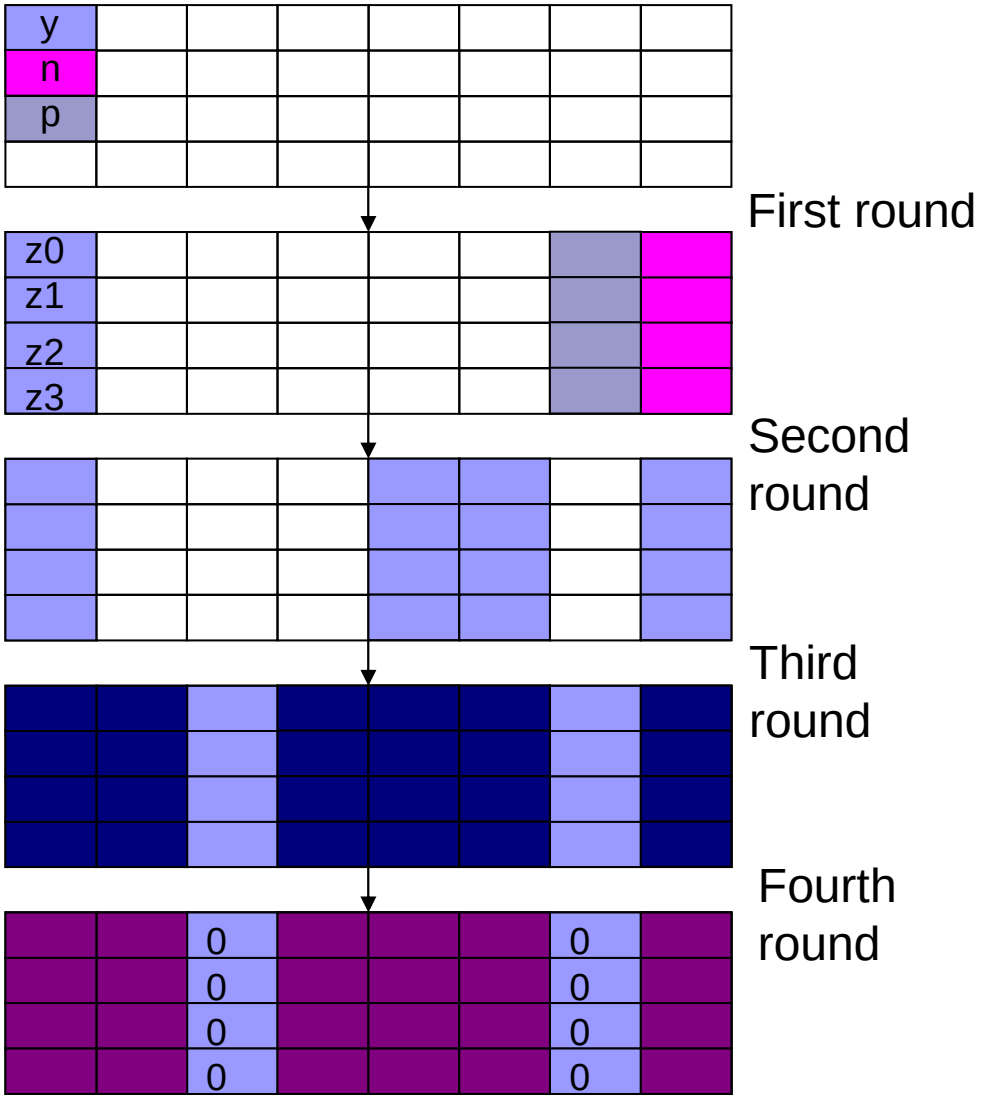
Note: SR: 1, 2, 4

Nb rounds: 14 (min)

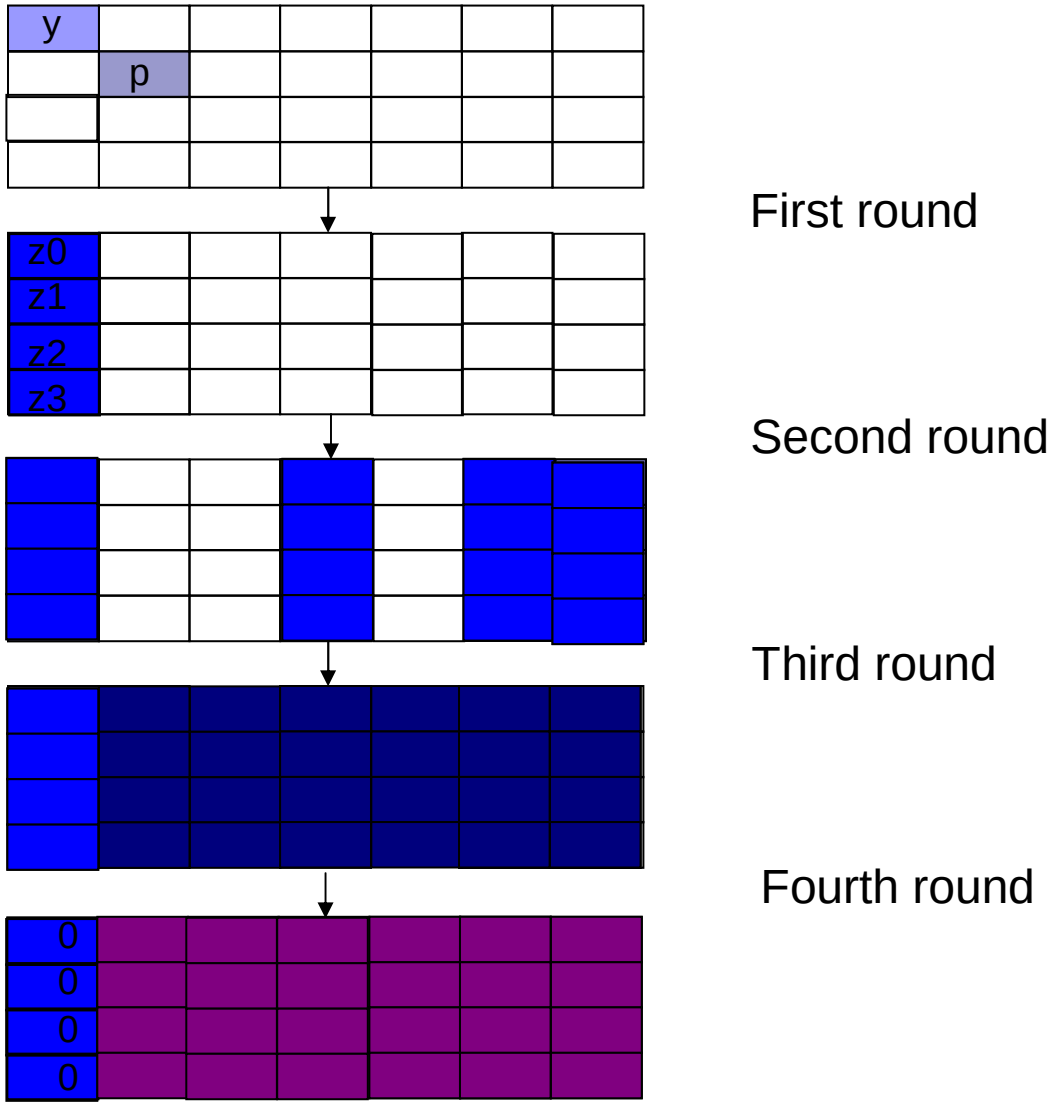
Rijndael 256 Integral property

Distinguisher on 4 rounds:

- Saturation on 3 bytes
- => Complexity: 2^{24} ciphers



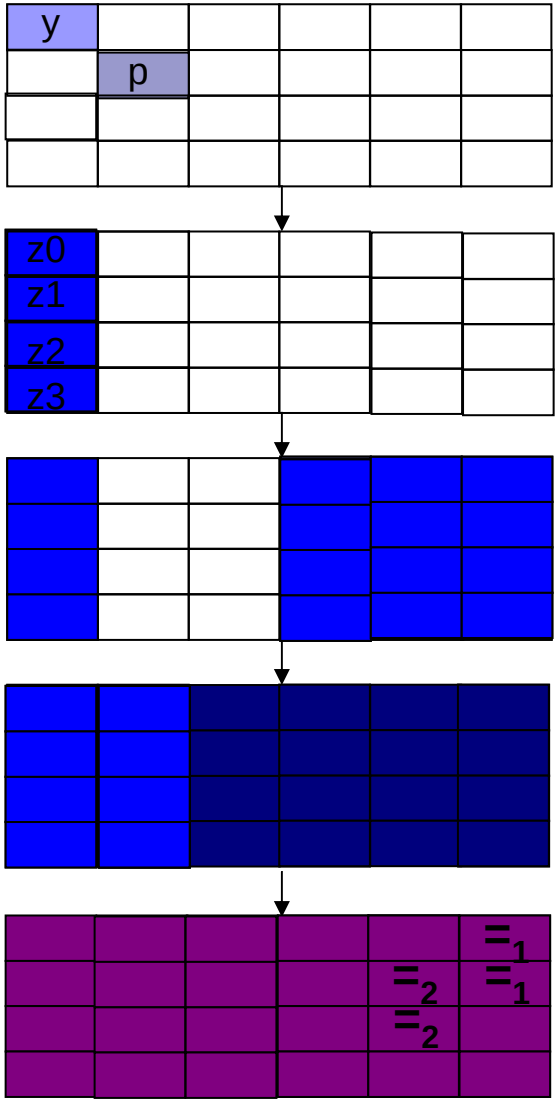
Rijndael 224 Integral property



Distinguisher on 4 rounds:

- Saturation on 2 bytes
- => Complexity: 2^{16} ciphers

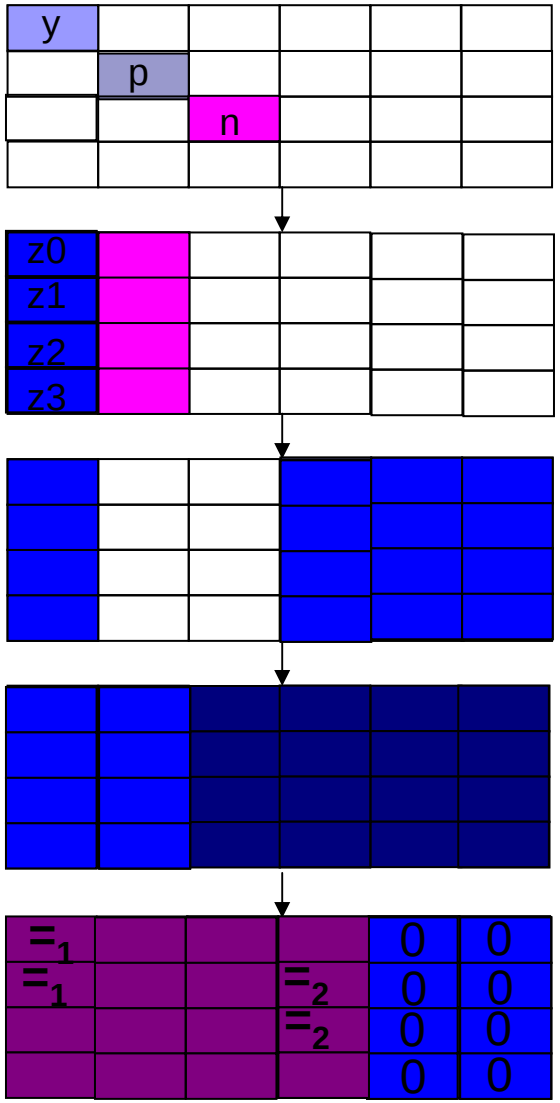
Rijndael 192
Integral property (1)



Distinguisher on 4 rounds:

- Saturation of 2 bytes
- => Complexity: 2^{16} ciphers

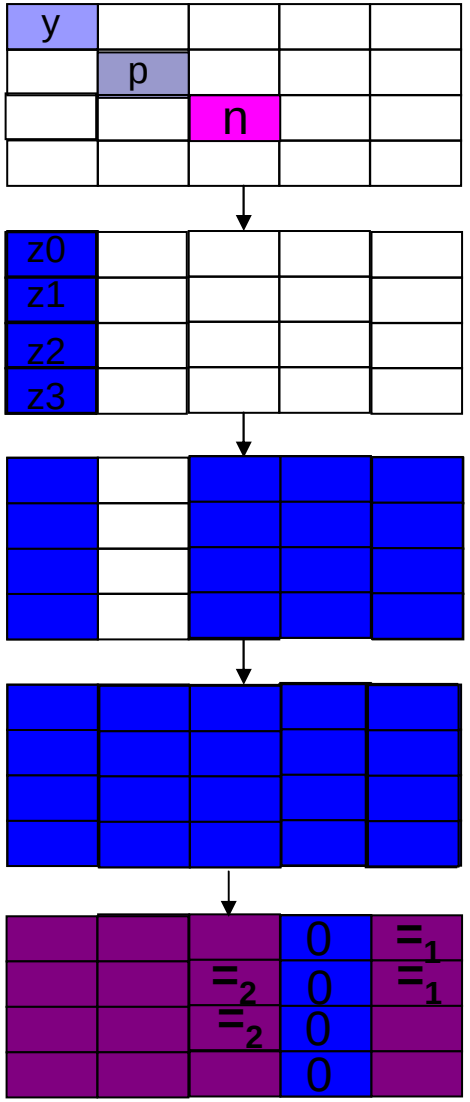
Rijndael 192
Integral property



Distinguisher on 4 rounds:

- Saturation on 3 bytes
- => Complexity: 2^{24} ciphers

Rijndael 160
Integral property



Distinguisher on 4 rounds:

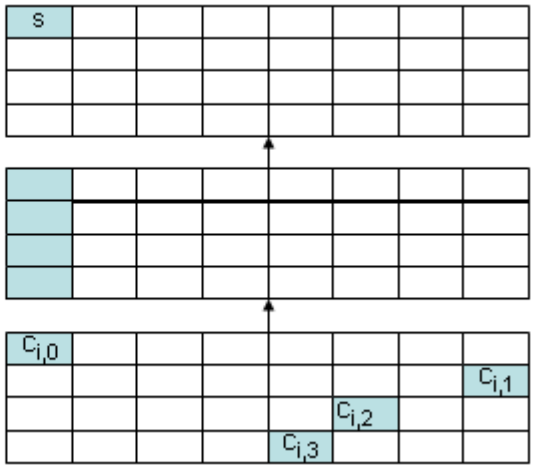
- Saturation de 3 bytes
- => Complexity: 2^{24} ciphers

Unknown keys Distinguishers

Extension of 2 rounds at the end

- [Ferguson and al. -00]: partial sums
- s directly deduced from $c_{i,j}$

$$\bigoplus_i S^{-1}\left[S_0\left[c_{i,0} \oplus k_0\right] \oplus S_1\left[c_{i,1} \oplus k_1\right] \oplus S_2\left[c_{i,2} \oplus k_2\right] \oplus S_3\left[c_{i,3} \oplus k_3\right] \oplus k_4\right]$$



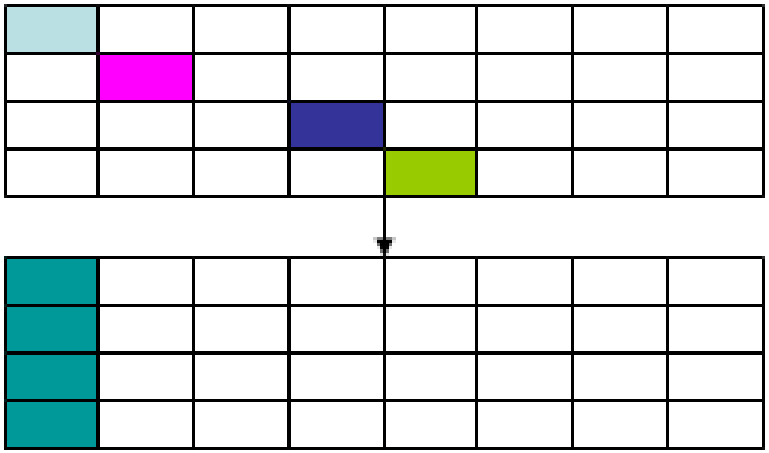
- For each ciphertext c, we associate the partial sum:

$$x_k := \sum_{j=0}^{k} S_j\left[c_j \oplus k_j\right] \text{ for } k \text{ from 0 to 3}$$

- Use $(c_0, c_1, c_2, c_3) \longrightarrow (x_k, c_{k+1}, \cdots, c_3)$ to sequentially determine k_k
  => Share in 4 steps the key serach

Extension at the beginning: 2 methods

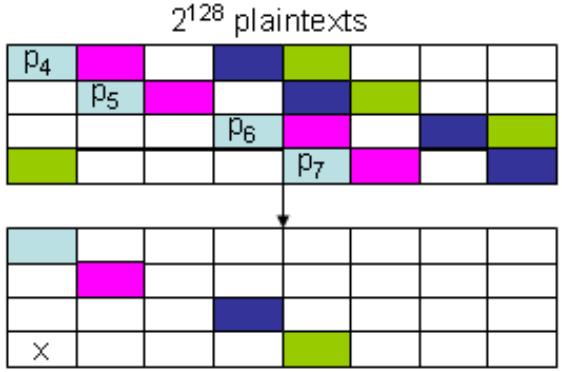
- [Ferguson and al. - 00]: one initial round



- => attack on 5 rounds with 2^{32} plaintexts

The herd technique

- One more round at the beginning:
  - Naively 2^{128} plaintexts (work, cf Nakhara and al.)
  - Fix a particular byte x => a herd: set of 2^{120} ciphertexts of 2^{88} structures
  - Test on a single herd.



- X depends on (p_4,…,p_7) and on 4 bytes of K_0
  1. Using 2^{64} counters m_y
  2. 2^{32} counters n_z
  3. Filter information on the key guess

Combine those extensions

- attack over 2+4+2=8 rounds (for Rijndael-256)
  1. Increment the 64 bits (c_0,…,c_3,p_4,…, p_7)
  2. Guess the 4 bytes of K_0, compute x, separate counters into herds.
  3. Choose a single herd, n_z en ajoutant (c_0,…,c_3) pour chaque y correct
  4. Guess the 5 bytes of K_7 and of K_6 of the two last rounds to decipher each z on one byte. Sum this value over the 2^{32} values of z and look at the 0s.
  5. Repeat this point for each value of theK_0 bytes.

- => The 4 bytes (p_4,…, p_7) and the 4 bytes of K_0 give 4 bytes
- => 2^{24} smaller herds => reduce the exhaustive search to 2^{128}-2^{119} plaintexts.

Complexity and attacks on 9 rounds

- Total cost:
  - 2^{128}-2^{119} plaintexts
  - 2^{120} cipher operations

- => Add one round at the end using a complete exhaustive search on the subkey K_9

Summary of the attacks

Rijndael-256	6	(all)	2^{32} CP	2^{72}	[4] (Integral)
	7	(all)	$2^{128} - 2^{119}$ CP	$2^{128} - 2^{119}$	[6] (Part. Sum)
	7	(all)	6×2^{32} CP	2^{44}	this paper
	8	(all)	$2^{128} - 2^{119}$ CP	$2^{128} - 2^{119}$	this paper
	9	(192)	$2^{128} - 2^{119}$ CP	2^{188}	this paper
	9	(256)	$2^{128} - 2^{119}$ CP	2^{204}	this paper

Cipher	nb rounds	Key sizes	Data	Time Complexity	Memory	Attack
Rijndael-256	6	(all)	$2 \cdot 2^{16}$ CP	2^{32}	2^{16}	2th-order integral
	7	(all)	$6 \cdot 2^{64}$ CP	2^{80}	2^{64}	8th-order integral
	8	> 192	$6 \cdot 2^{192}$ CP	2^{208}	2^{192}	24th-order integral
	8	(192)	$19 \cdot 2^{64}$ CP	2^{191}	2^{64}	8th-order integral
	8	(256)	$21 \cdot 2^{64}$ CP	2^{207}	2^{64}	8th-order integral
Rijndael-224	6	(all)	$2 \cdot 2^{16}$ CP	2^{32}	2^{16}	2th-order integral
	7	(all)	$6 \cdot 2^{64}$ CP	2^{80}	2^{64}	8th-order integral
	8	> 192	$6 \cdot 2^{192}$ CP	2^{208}	2^{192}	24th-order integral
	8	(192)	$19 \cdot 2^{64}$ CP	2^{191}	2^{64}	8th-order integral
	8	(256)	$21 \cdot 2^{64}$ CP	2^{207}	2^{64}	8th-order integral
Rijndael-192	6	(all)	$2 \cdot 2^{16}$ CP	2^{33}	2^{16}	2th-order integral
	7	(all)	$6 \cdot 2^{64}$ CP	2^{81}	2^{104}	8th-order integral
	6	(all)	$2 \cdot 2^{24}$ CP	2^{40}	2^{24}	3th-order integral
	7	(all)	$6 \cdot 2^{92}$ CP	2^{108}	2^{92}	12th-order integral
	8	(256)	$21 \cdot 2^{64}$ CP	2^{208}	2^{104}	8th-order integral
Rijndael-160	6	(all)	$2 \cdot 2^{24}$ CP	2^{40}	2^{24}	3th-order integral
	7	(all)	$6 \cdot 2^{92}$ CP	2^{108}	2^{92}	12th-order integral

Table 2. Summary of Attacks on Rijndael-b using the partial sums technique

Known Keys Distinguishers

[Knudsen – Rijmen 07]

- Notion of Known Key Distinguisher
  - Principle: create a distinguisher beginning at the middle of the cipher
  - Then, determine a particular property linking plaintexts and ciphertexts
  - Comparison withe the complexity required to find such a structure for a random permutation
- Interest: create distinguishers when block ciphers are used as hash functions

Theoritical model [Africacrypt 09]

- Advantage of Distinguishers [Vaudenay 97]: $Adv_E(A)$

$$Adv_E^{PRP}(\mathcal{A}) = \Pr\left[K \xleftarrow{\$} \mathcal{K} : A^{E_K(\cdot)} = 1\right] - \Pr\left[G \xleftarrow{\$} \mathrm{Perm}(\mathcal{M}) : A^{G(\cdot)} = 1\right]$$

$$Adv_E^{SPRP}(\mathcal{A}) = \Pr\left[K \xleftarrow{\$} \mathcal{K} : A^{E_K(\cdot),E_K^{-1}(\cdot)} = 1\right] - \Pr\left[G \xleftarrow{\$} \mathrm{Perm}(\mathcal{M}) : A^{G(\cdot),G^{-1}(\cdot)} = 1\right]$$

- Two more cases: non-adaptative, adaptative

Case of an adaptative SPRP Distinguisher

Algorithm 2 An n-limited generic adaptive distinguisher with chosen input plaintexts or output ciphertexts

Parameters: functions $g_1, \cdots, g_n$, a set $A^{(n)}$
Oracle: an oracle $\mathcal{O}$ implementing permutations c and c^{-1}
Select a fixed direction and message $(B_1, Z_1^0) = g_1()$ and get $Z_1^1 = c(Z_1^0)$ if $B_1 = 0$ or $Z_1^1 = c^{-1}(Z_1^0)$ otherwise
Calculate a direction and a message $(B_2, Z_2^0) = g_2(Z_1^1)$ and get $Z_2^1 = c(Z_2^0)$ if $B_2 = 0$ or $Z_2^1 = c^{-1}(Z_2^0)$ otherwise
...
Calculate a direction and a message $(B_n, Z_n^0) = g_n(Z_1^1, \cdots, Z_{n-1}^1)$ and get $Z_n^1 = c(Z_n^0)$ if $B_n = 0$ or $Z_n^1 = c^{-1}(Z_n^0)$ otherwise
if $(Z_1^1, \cdots, Z_n^1) \in A^{(n)}$ **then**
  Output 1
else
  Output 0
end if

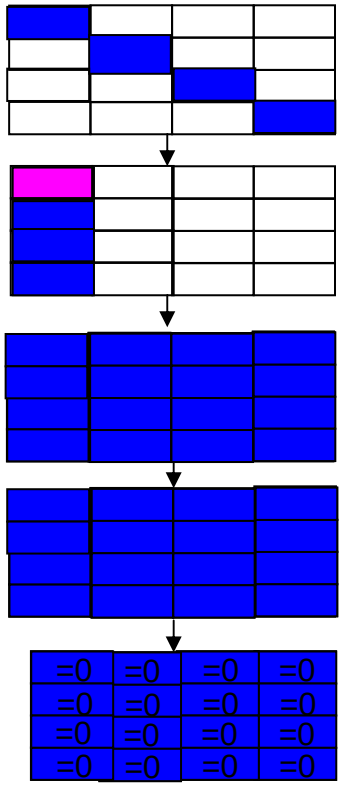
Case of a non-adaptative Known Key Distinguisher

Algorithm 3 An n-limited generic non-adaptive chosen middletexts distinguisher (NA-CMA)

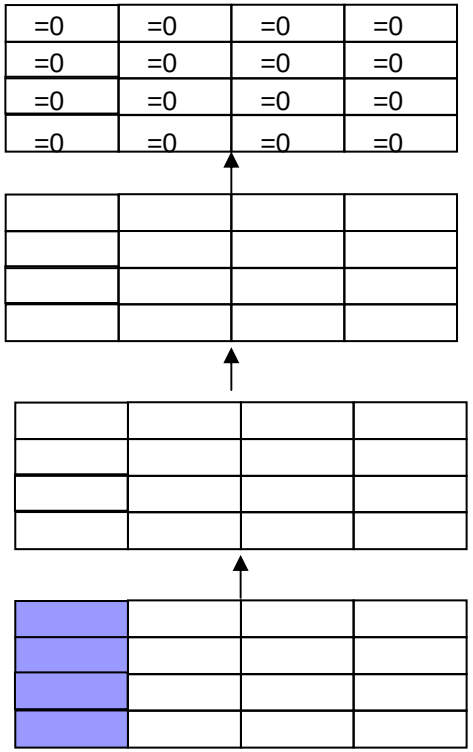
Parameters: a complexity n, an acceptance set $A^{(n)}$
Oracle: an oracle $\mathcal{O}$ implementing internal functions f_1 (resp. f_2) of permutation c that process input middletexts to the plaintext (resp. ciphertext) end
Compute some middletexts $\mathbf{M} = (M_1, \cdots, M_n)$
Query $\mathbf{P} = (P_1, \cdots, P_n) = (f_1(M_1), \cdots, f_1(M_n))$ and $\mathbf{C} = (C_1, \cdots, C_n) = (f_2(M_1), \cdots, f_2(M_n))$ to $\mathcal{O}$
if $(\mathbf{P}, \mathbf{C}) \in A^{(n)}$ **then**
  Output 1
else
  Output 0
end if

Case of study: the AES [Knu-Rij 07]

- Forward sense

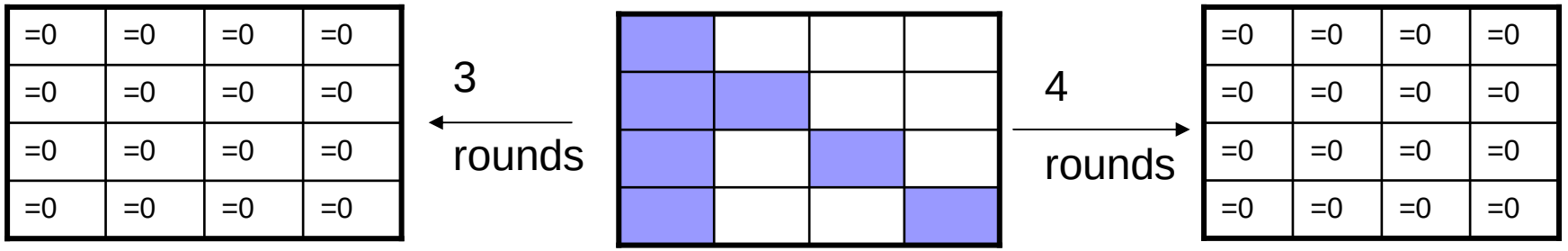


- Backward sense



KK distinguisher for the AES

- KK distinguisher on 7 rounds
  - 3 in backward, 4 in forward



  - Requires 2^{56} middletexts and 2^{56} cipher operations
  - For a random permutation => k-sum problem, Complexity: 2^{58} operations
- => KK distinguisher for the AES

KK distinguisher for Rijndael

- Same kind of properties in the backward sense
- Summary of the KK distinguishers for Rijndael [Africacrypt 2009]:

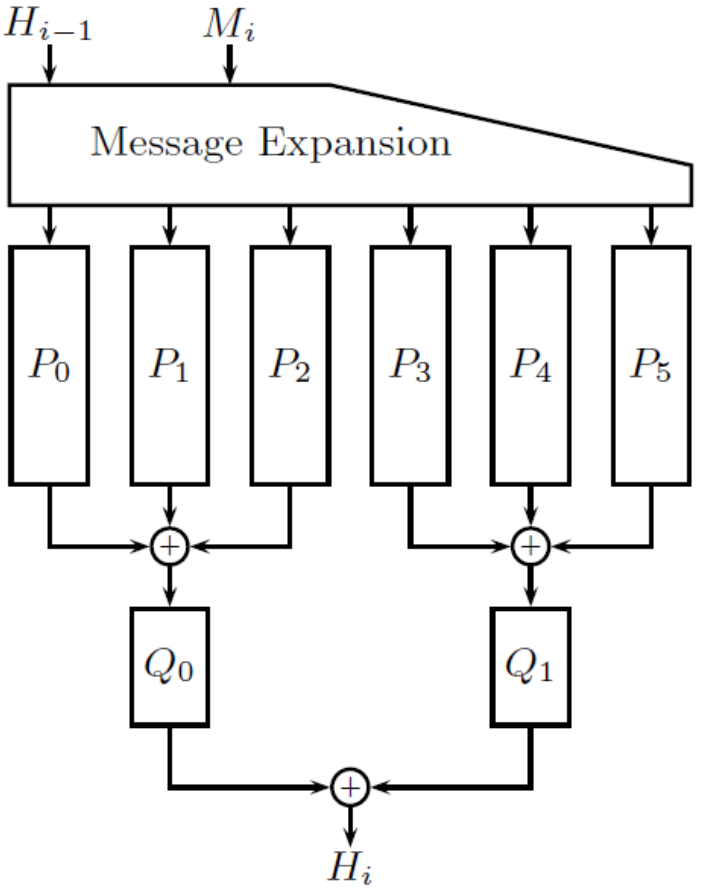
Cipher	nb rounds	Key sizes	Data	Time Complexity	Memory	Source
AES	7	(all)	2^{56} CM	2^{56}	small	[12]
Rijndael-256	8	(all)	2^{40} CM	2^{40}	small	this paper
Rijndael-224	8	(all)	2^{72} CM	2^{72}	small	this paper
Rijndael-192	7	(all)	2^{32} CM	2^{32}	small	this paper
Rijndael-160	7	(all)	2^{40} CM	2^{40}	small	this paper

Table 2. Summary of known-key distinguishers on Rijndael-b. CM means Chosen Middle-texts.

A last idea…

LANE: SHA 3 hash function

- $H_i = h_0 \| h_1 = 256$ bits
- $M_i = m_0 \| m_1 \| m_2 \| m_3 = 512$ bits
- P_i = 6 modified AES rounds
- Q_i = 3 modified AES rounds

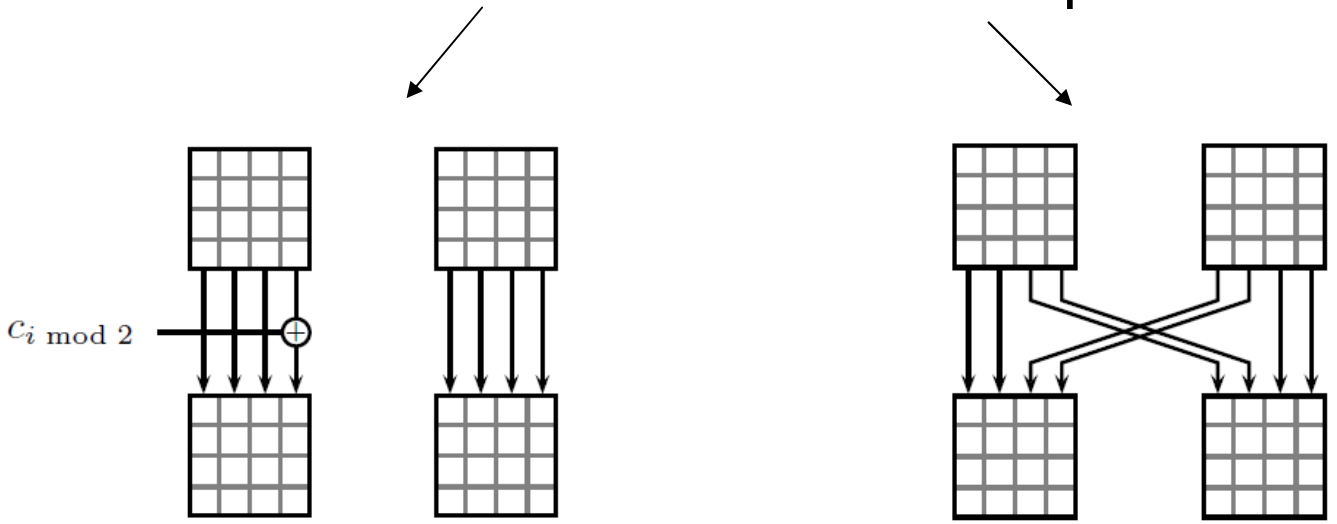


the P_i inputs

$$
\begin{array}{ll|l}
W_0 = h_0 \oplus m_0 \oplus m_1 \oplus m_2 \oplus m_3 & || & h_1 \oplus m_0 \oplus m_2 \\
W_1 = h_0 \oplus h_1 \oplus m_0 \oplus m_2 \oplus m_3 & || & h_0 \oplus m_1 \oplus m_2 \\
W_2 = h_0 \oplus h_1 \oplus m_0 \oplus m_1 \oplus m_2 & || & h_0 \oplus m_0 \oplus m_3 \\
W_3 = h_0 & || & h_1 \\
W_4 = m_0 & || & m_1 \\
W_5 = m_2 & || & m_3
\end{array}
$$

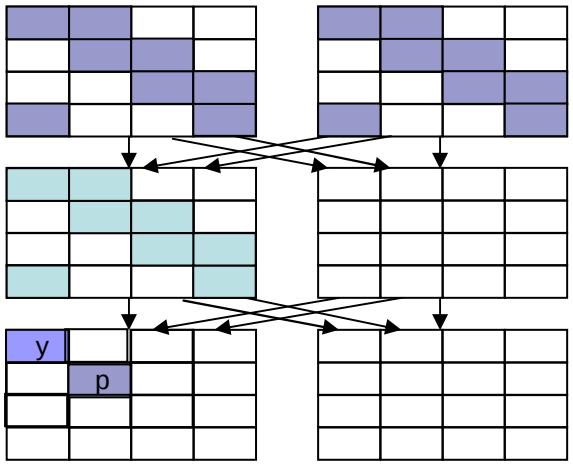
P_is and Q_is (LANE 256)

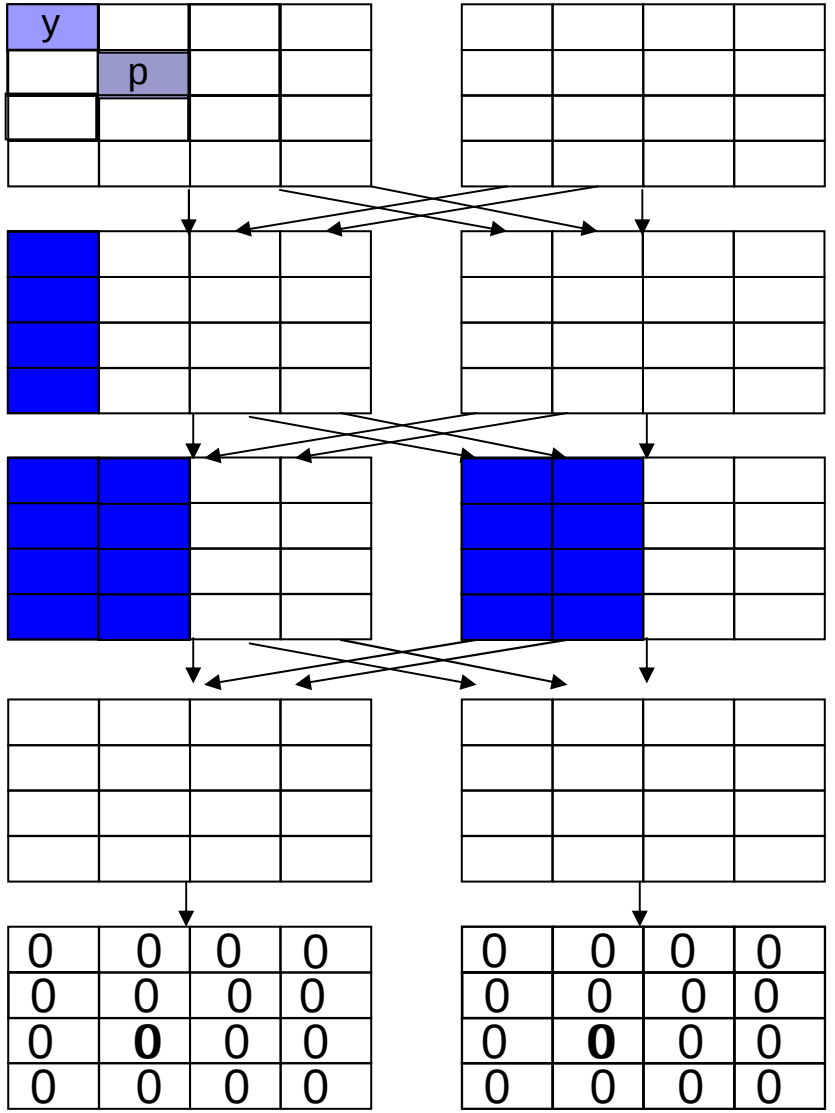
- The same operations than the ones of the AES
  - SubBytes, ShiftRows, MixColumns, KeyAdd (with constants)
  - Two more: AddConstants and SwapColumns



Integral properties of LANE-256

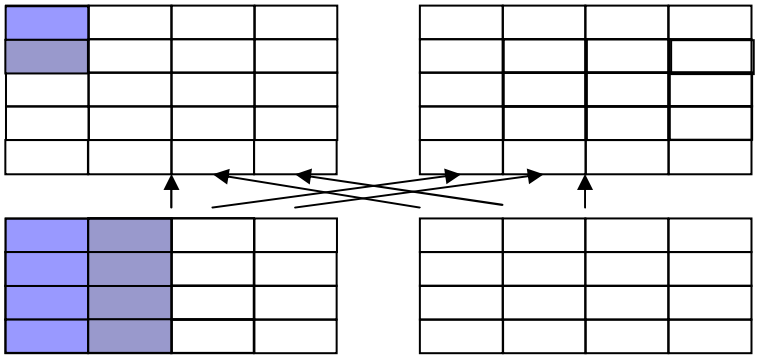
- 4 rounds + extension at the beginning:

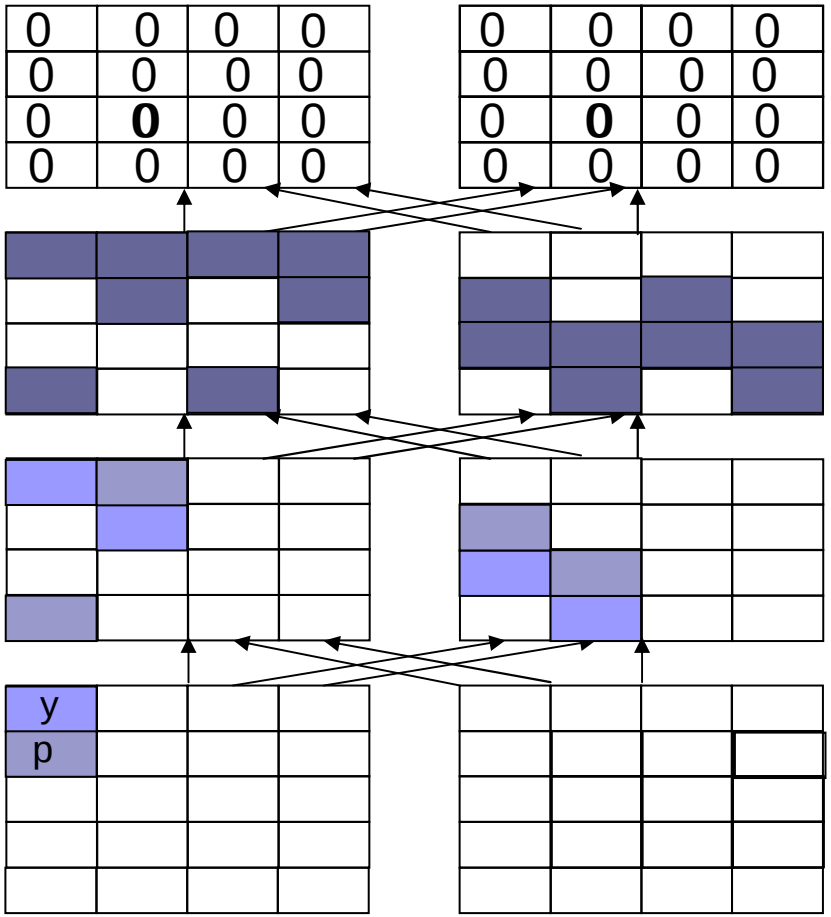




Integral property of LANE-256 backward sense

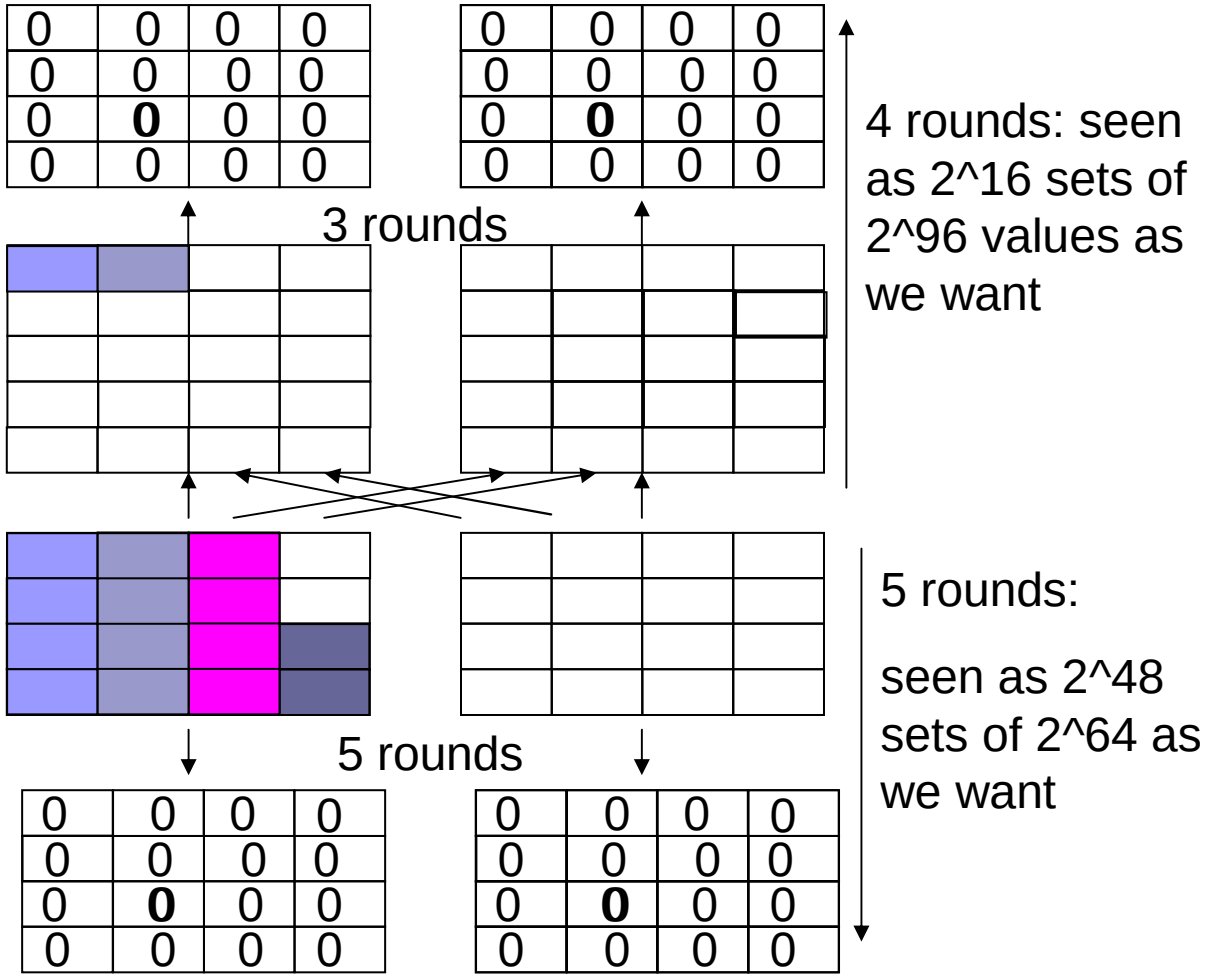
- Integral property on 3 rounds + extension at the beginning:





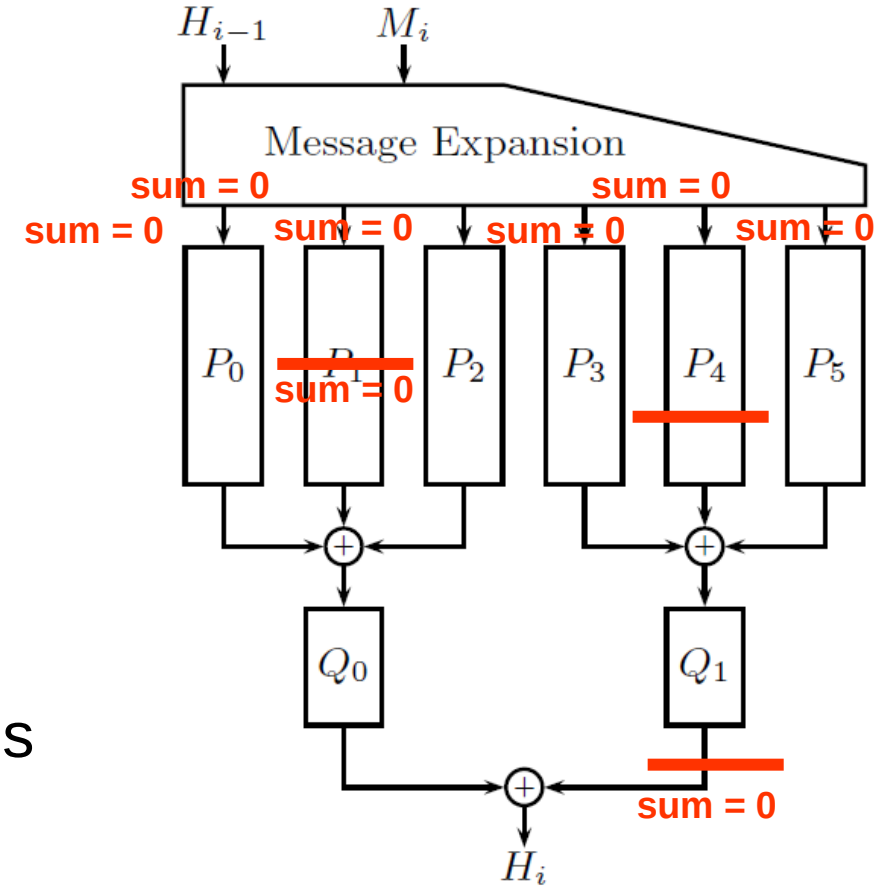
Combine the two properties

- Distinguisher in 2^{112} on the right part of LANE-256



Why only one half ?

- If $h_0=h_1=m_2=m_3$ = cte:
  - W0 = m0 + m1 || m0
  - W1 = m0 || m1
  - W2 = m0 + m1 || m0
  - W3 = 0 || 0
  - W4 = m0 || m1
  - W5 = 0 || 0
- Then:
  - over 2^{112} messages, a certain number of sums is equal to 0



Conclusion

- Integral properties of Rijndael were not well studied
  - Unknown Keys Distinguishers
  - Known Keys Distinguishers

- The last model is really useful to create distinguishers for the SHA-3 competition (cf: LANE)