Leveraging the Exchange Distinguisher in a 6-Round Key Recovery Attack on AES

Henri Gilbert[1,3], Rachelle Heim Boissier[2], Jean-René Reinhard[1]

(1) ANSSI, FR

(2) Université Libre de Bruxelles (ULB), BE

(3) Université de Versailles Saint-Quentin-en-Yvelines (UVSQ), FR

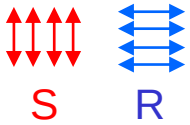
Outline

- Motivation: assess the potential of the *exchange distinguisher* of [Rønjom-Bardeh-Helleseth17, Bardeh-Rønjom19] to serve as a basis for a key recovery attack on AES.
- Our attack was developed in 2021. It was never published: we hoped to find 7-round extension before submitting and this did eventually not happen. It was presented by Rachelle at *Journées C2*, Hendaye, 10-15 April 2022

This presentation will consist of:

❶ A brief reminder of *the untwisted representation of AES* [G14]

- i.e. a further simplification of the *SuperSbox representation* of [DR06]

10 full round $\cong$ 5 ``super-rounds'' 

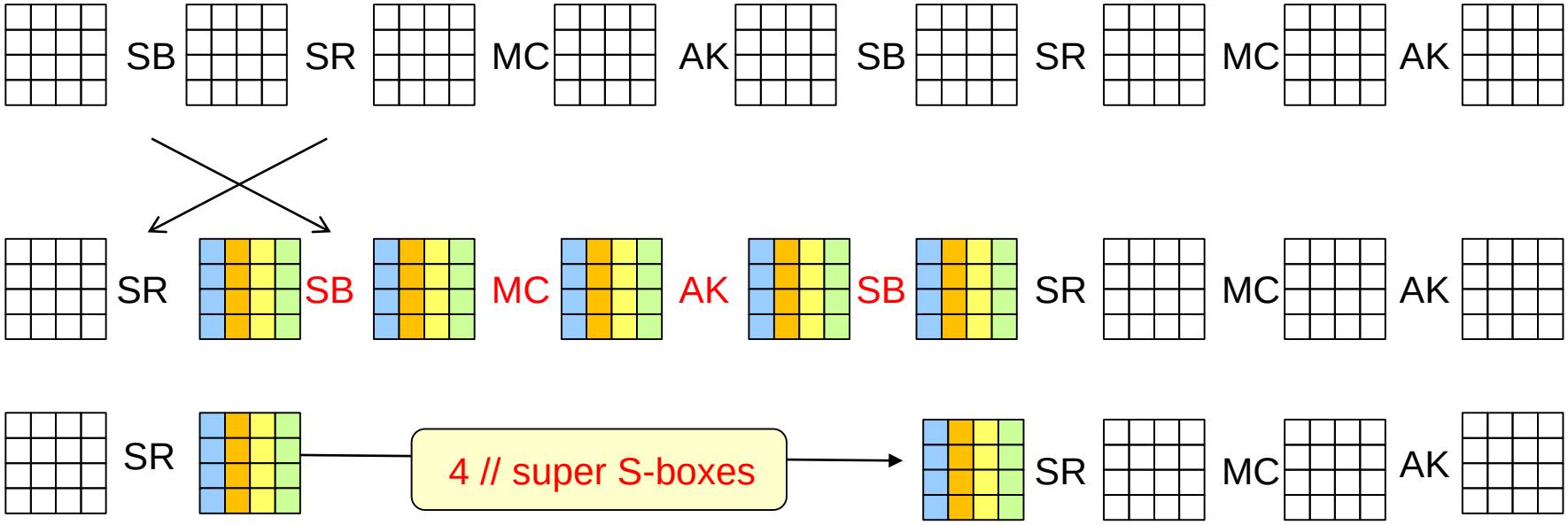
❷ An outline of a simple key recovery attack on a 6-round version of AES-128 that leverages the 4-round version of the exchange distinguisher

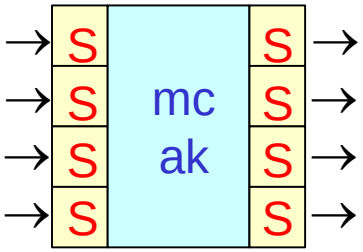
- the untwisted representation of AES will be used to simplify the attack description

❸ Some insight into the feasability of merging the exchange distinguisher and the « collision of partial functions » distinguisher of [GM00] into a stronger combined distinguisher.

❶ Starting point: super S-box notion [DR06]

- equivalent representation of 2 consecutive AES rounds





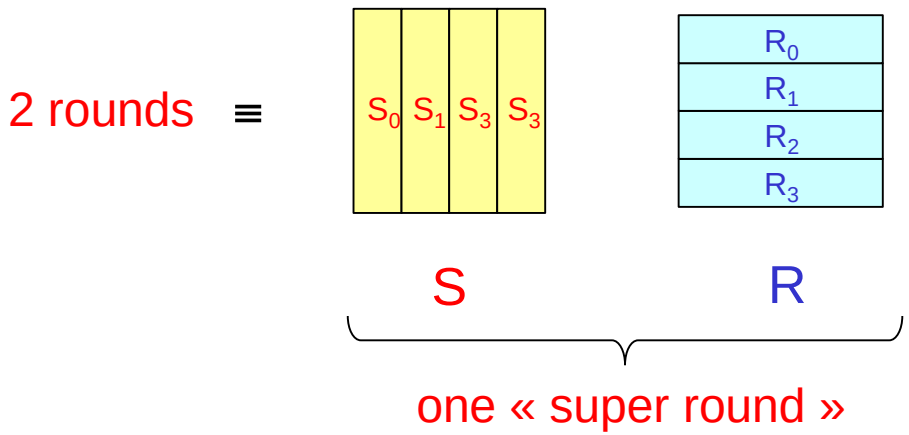
one super S-box

Untwisted representation: objective

- start from the super S-box representation of 2 rounds

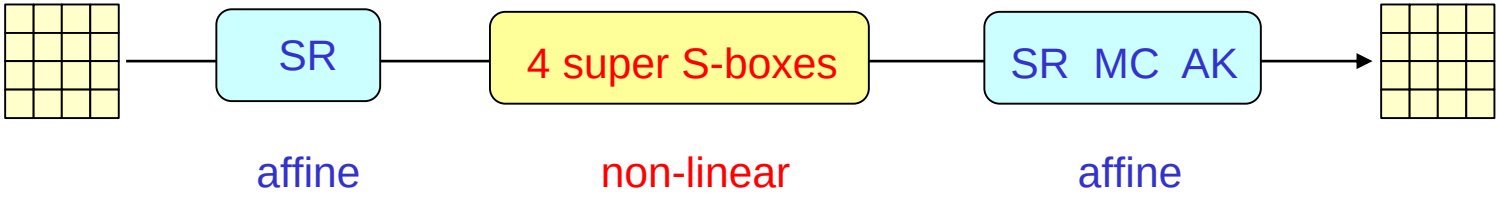
  but eliminate « ShiftRows » to get a simplified view

- equivalently represent 2 consecutive AES rounds as the composition of
  - a nonlinear transformation: essentially 4 parallel « super S-boxes »
  - an affine transformation: essentially 4 parallel « MixRows »



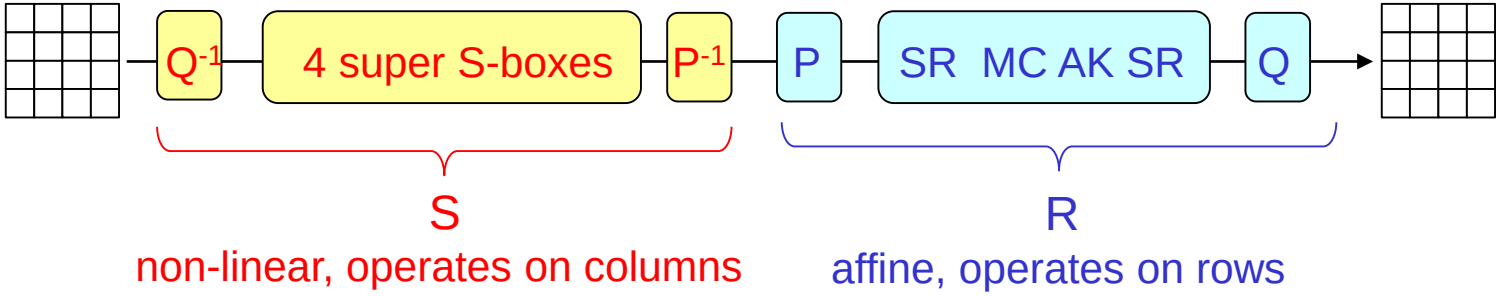
How to move to the untwisted representation (1/2)

- step 1: super S-boxes representation of 2 rounds



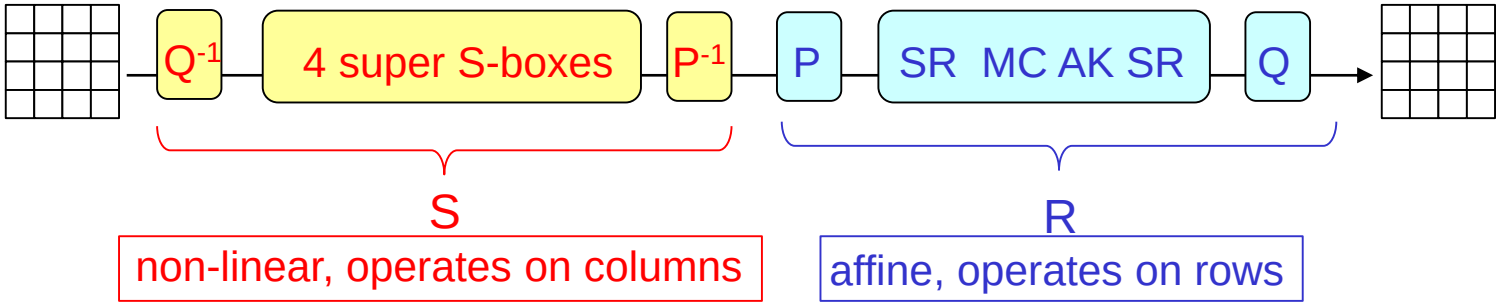
or equivalently (up to a cyclic shift of the periodic 2-round pattern)

- step 2: composition with well chosen byte permutations P and Q and their inverses



How to move to the new representation (2/2)

- problem: find suitable byte permutations P and Q at step 2

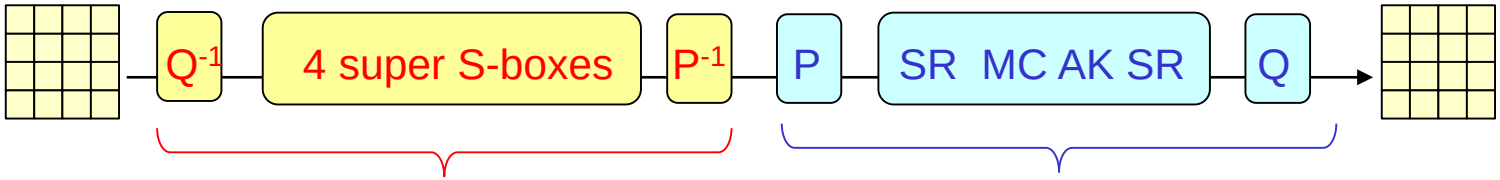


- solution: $P = SR\ T\ SR^{-1}$ and $Q = SR^{-1}\ T\ SR\ SC$

  where: T = matrix transposition ;

  SC = swapping of columns 2 et 4

Detail of P, Q and R



→ we will denote R's linear part by R_l

- more explicitly:

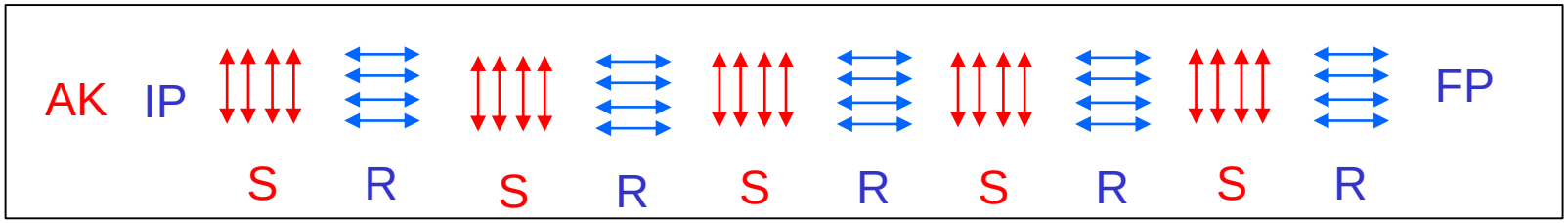
$$P:\begin{pmatrix} a_0 & a_4 & a_8 & a_{12} \\ a_1 & a_5 & a_9 & a_{13} \\ a_2 & a_6 & a_{10} & a_{14} \\ a_3 & a_7 & a_{11} & a_{15} \end{pmatrix} \mapsto \begin{pmatrix} a_0 & a_5 & a_{10} & a_{15} \\ a_3 & a_4 & a_9 & a_{14} \\ a_2 & a_7 & a_8 & a_{13} \\ a_1 & a_6 & a_{11} & a_{12} \end{pmatrix} \qquad Q:\begin{pmatrix} a_0 & a_4 & a_8 & a_{12} \\ a_1 & a_5 & a_9 & a_{13} \\ a_2 & a_6 & a_{10} & a_{14} \\ a_3 & a_7 & a_{11} & a_{15} \end{pmatrix} \mapsto \begin{pmatrix} a_0 & a_7 & a_{10} & a_{13} \\ a_1 & a_4 & a_{11} & a_{14} \\ a_2 & a_5 & a_8 & a_{15} \\ a_3 & a_6 & a_9 & a_{12} \end{pmatrix}$$

$$R_0 = R_2 = \begin{pmatrix} 2 & 3 & 1 & 1 \\ 3 & 1 & 1 & 2 \\ 1 & 1 & 2 & 3 \\ 1 & 2 & 3 & 1 \end{pmatrix} \qquad R_1 = R_3 = \begin{pmatrix} 1 & 1 & 2 & 3 \\ 1 & 2 & 3 & 1 \\ 2 & 3 & 1 & 1 \\ 3 & 1 & 1 & 2 \end{pmatrix} \qquad \text{MixColumns matrix}: M = \begin{pmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{pmatrix}$$

- note: other pairs air of byte permutations (P,Q) provide an equivalent representation. It can be shown that the group $S_4 \times S_4$ operates on the set of solutions.

Untwisted representation of AES_{10+} and AES_{10}

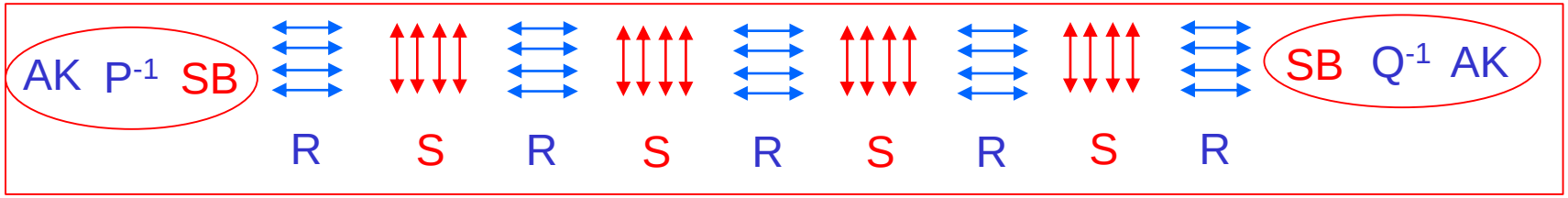
AES_{10+} = 10 full AES-128 rounds with MC in last round



more generally: $AES_{2r+} = AK \,.\, IP \,.\, (S \,.\, R)^r \,.\, FP$

- where: IP = SR Q et $FP = IP^{-1}$: initial and final byte permutations (no or little influence on security)

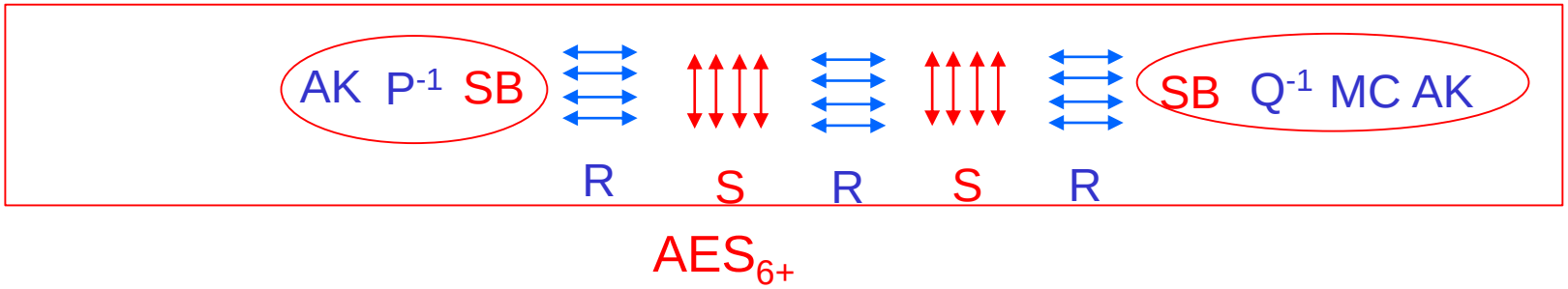
AES_{10} ≡ full AES-128 (w/o MC in last round)



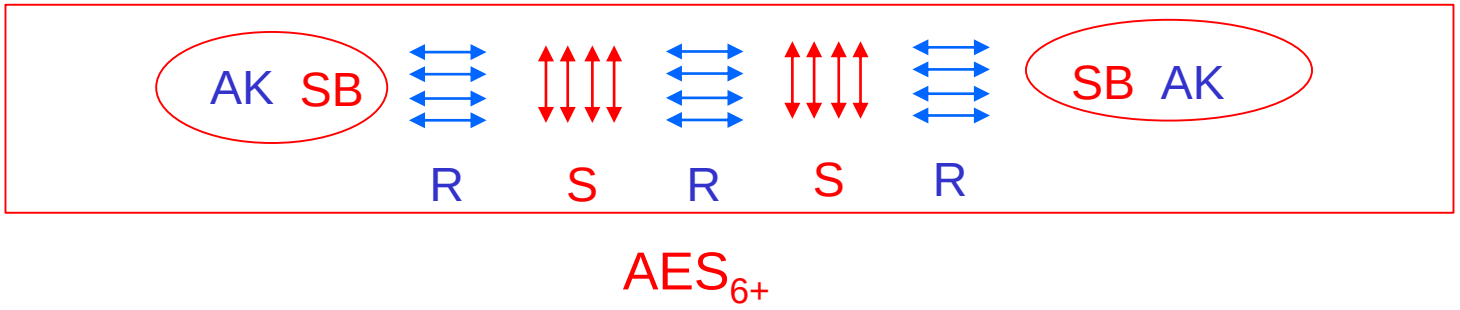
more generally: $AES_{2r} = AK \,.\, P^{-1} \,.\, SB \,.\, R \,.\, (S \,.\, R)^{r-1} \,.\, SB \,.\, Q^{-1} \,.\, AK$

$AES_{2r+} = AK \,.\, P^{-1} \,.\, SB \,.\, R \,.\, (S \,.\, R)^{r-1} \,.\, SB \,.\, Q\text{-}1 \,.\, MC \,.\, AK$

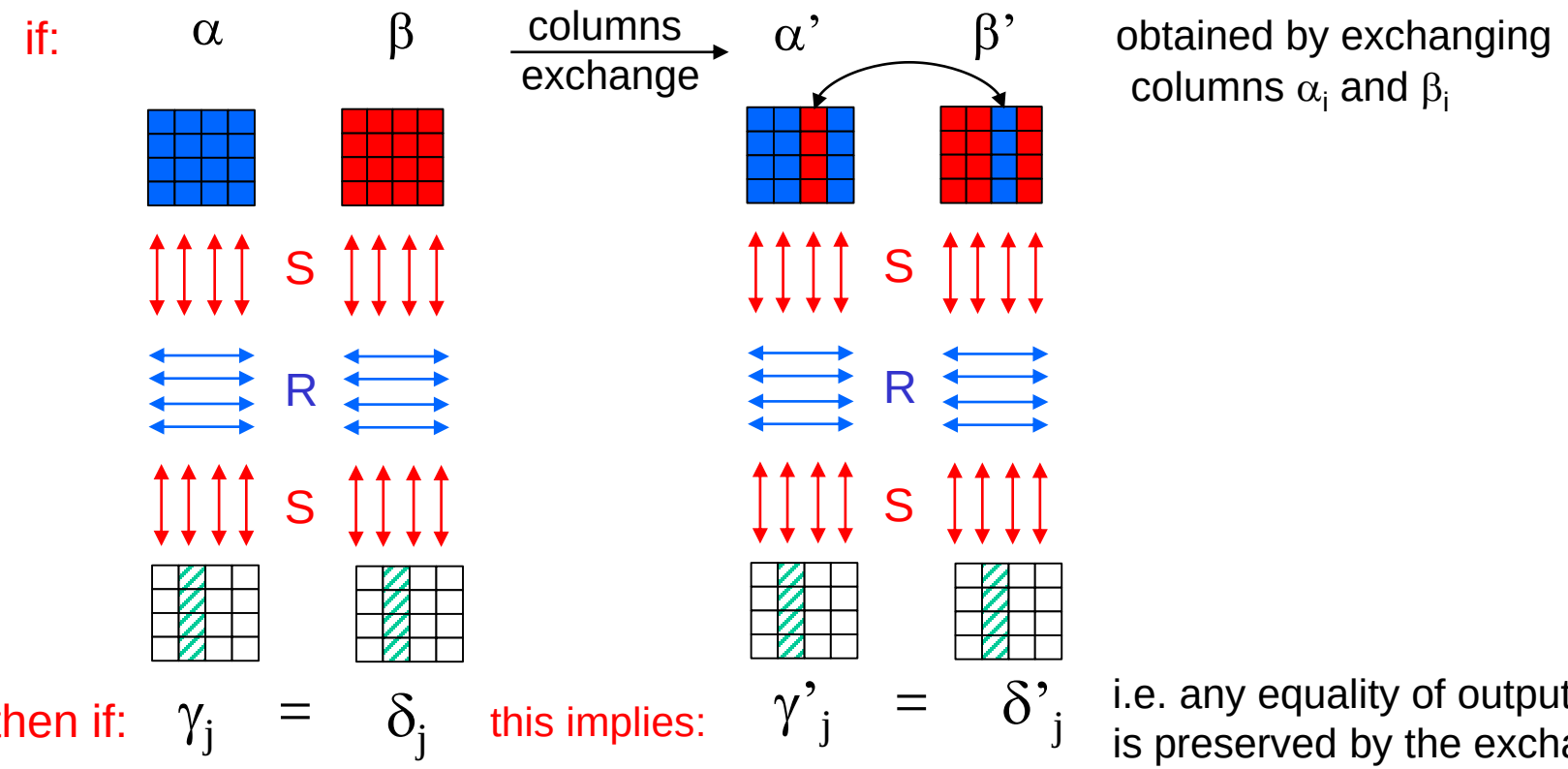
Untwisted representation of AES_{6+} used in the sequel



Since in the sequel we do not attempt to leverage the properties of the key schedule (e.g. key bridging, etc.), AES_{6+} can equivalently be represented (up to linear changes of representation of the paintexts, the ciphertexts and the subkeys K^0 to K^6) by:

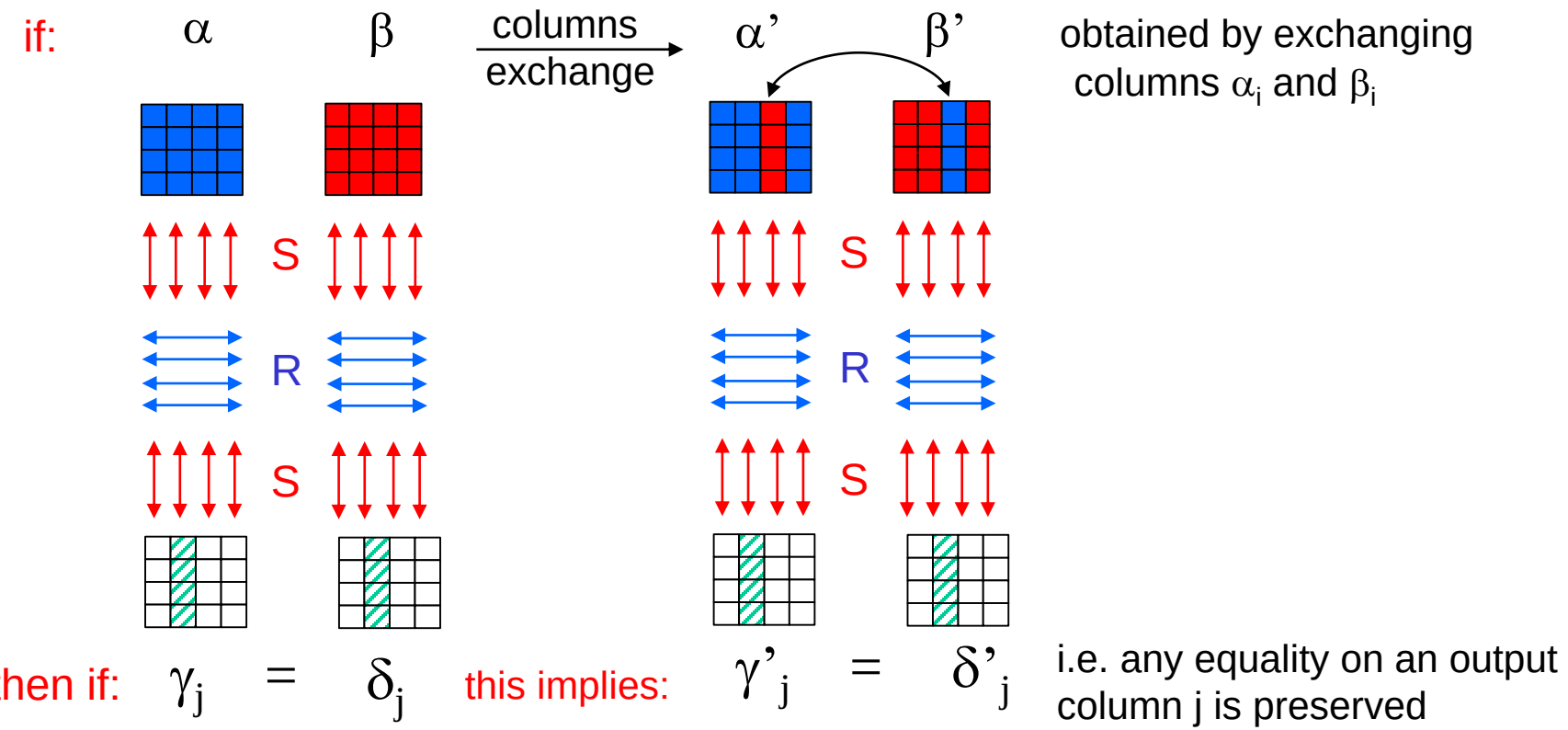


❷ The 4-round exchange distinguisher of [BR19]



- Exchange property: if S ∘ R ∘ S (α) and S ∘ R ∘ S (β) collide on any output column $j \in \{0,1,2,3\}$, then for any input column position $i \in \{0,1,2,3\}$ this equality is preserved if one exchanges columns i of α and β.

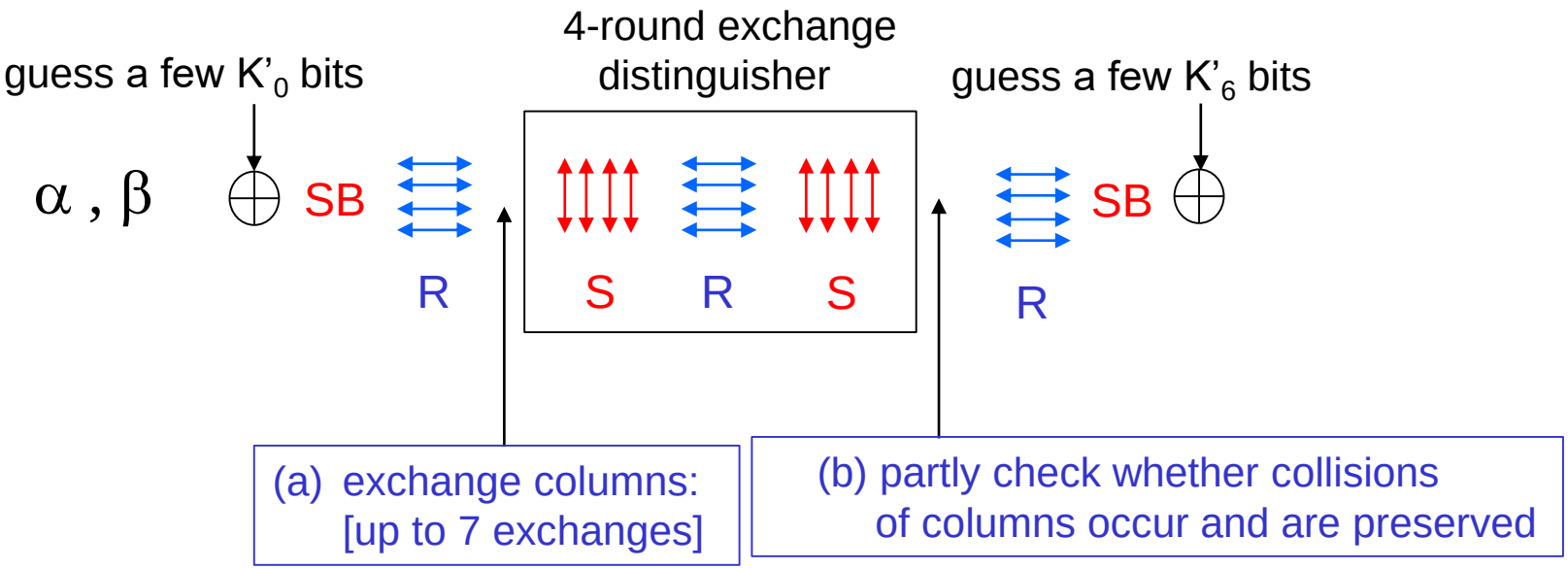
❷ The 4-round exchange distinguisher of [BR19]



- Exchange property: if S ∘ R ∘ S (α) and S ∘ R ∘ S (β) collide on any output column j∈{0,1,2,3}, then for any input column position i∈{0,1,2,3} this equality is preserved by exchanging columns i of α and β.

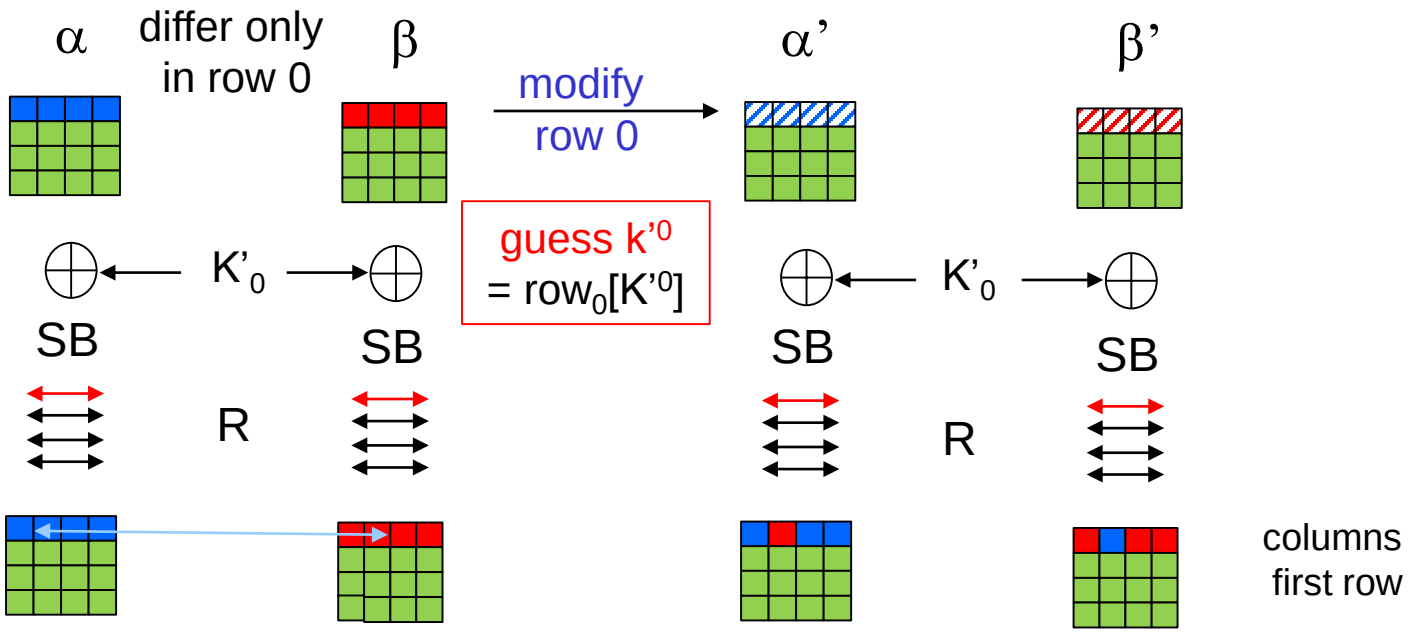
- Proof: swapping columns i neither affects the unordered pairs $\{\alpha_i , \beta_i\}$ of columns nor the output difference Δ of the first S mapping. This implies that $\gamma_j = \delta_j$ and $\gamma'_j = \delta'_j$ are equivalent since they both occur iff the j-th column of $R_I(\Delta)$ is 0.

Resulting key recovery attack (roadmap)



- Note: while there exist 5-round and 6-round extensions of the 4-round exchange distinguisher of [RBH17, BR19], we will not consider them here because they appear to be difficult to leverage for recovering the subkeys of extra external rounds.

(a) Column exchange after ~ 1 round

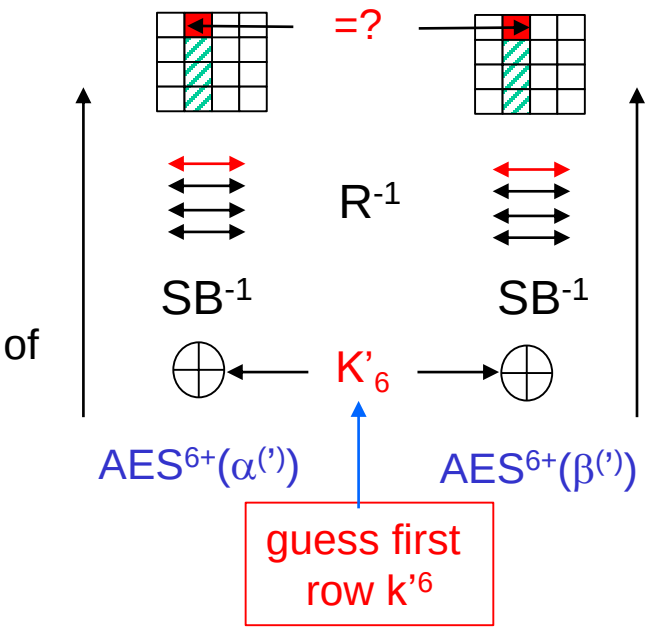


Observation: given a right 32-bit guess on the first row k'_0 of K'_0,
one can control values injected in the first row at the output of $R_l \circ SB \circ AK$
and thus exchange the bytes j of the first output row.
Since the other output rows are unaffected, the output columns j are swapped.

→ One can compute up to 7 new unordered plaintext pairs {α',β'}, that after ~1 round realize an exchange of the following subsets of columns: {0}; {1}; {2}; {3}; {0,1}; {0,2}; {0,3}

(b) Partly detecting collisions on a column before ~1 last round

- To partly test (on one red row 0 byte) whether a collision on a column occurs at the output of the distinguisher,

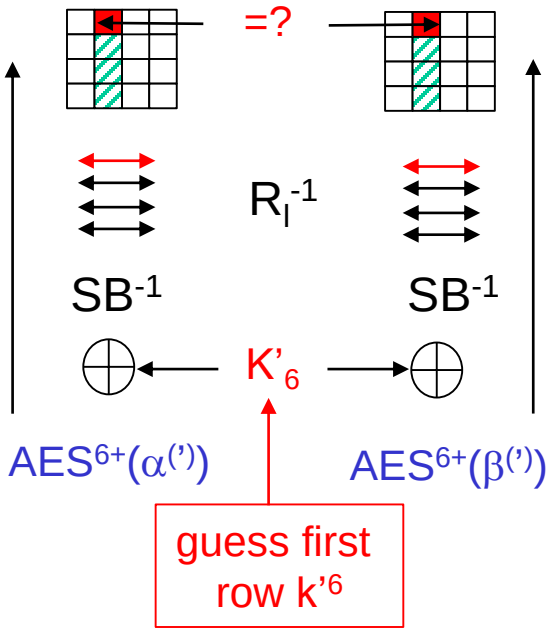


it suffices to guess the first row k'^{6} of K'^{6} (32 bits)

‘

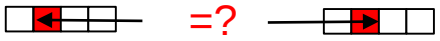
(b) Partly detecting collisions on a column before ~1 last round

- To partly test (on one first row byte) whether a collision on a column occurs at the output of the distinguisher,



it suffices to guess the first row k'^6 of K'^6 (32 bits)

variable separation: the left and right 16-bit halve $k'^{6,l}$ and $k'^{6,r}$ of k'^6 are involved separately in the upward computation of the up to 8 equations :

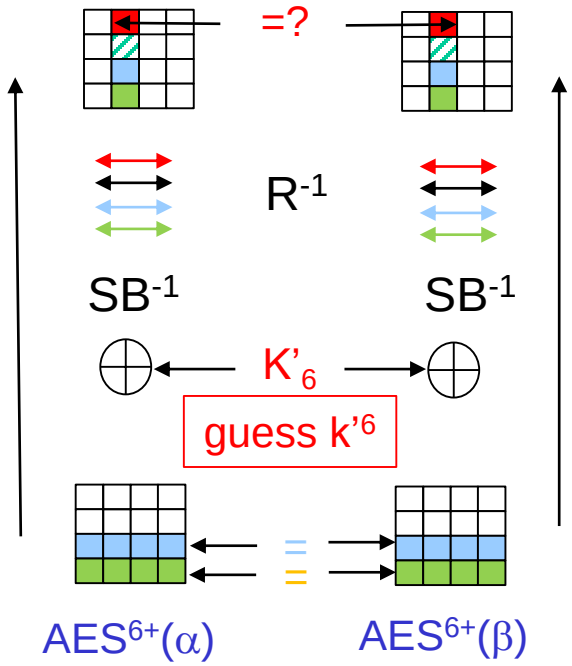


for all $\{\alpha^{(')},\beta^{(')}\}$ pairs.

→ square root savings [2^{17} instead of 2^{32}] in the complexity factor associated with guessing k'^6 (using a collision search between the contributions of both halves)

(b) Filtering trick for the pair {α, β}

- encrypt N structures of 2^{32} plaintext blocks that differ only in row 0
- choose α and β candidate blocks that collide on 2 AES^{6+} output rows (e.g. rows 2 and 3)



→ [since α and β are selected as to ensure that collisions hold on top blue and green bytes]

if for a right guess on k'^6 a collision holds on top red bytes

then the probability that both top columns collide at the output of the distinguisher is as high as 2^{-8}

Key recovery: putting everything together

- encrypt N structures of 2^{32} plaintext blocks that differ only in row 0. This provides a set S of $\approx$ N $2^{2x32-1}/ 2^{64}$ = ½ N unordered « reference pairs » $\{\alpha, \beta\}$ of input blocks s.t.
  - α, β differ in all their 4 row 0 bytes
  - $AES^{6+}(\alpha)$ and $AES^{6+}(\beta)$ collide on two rows, e.g. rows 2 and 3

- for each of the N unordered pairs $\{\alpha, \beta\}$ of S
  for each first-round key guess on k'^0 (32 bits)
  - compute 7 new unordered pairs $\{\alpha', \beta'\}$ and encrypt them under AES^{6+}
  - test if there exists a pair of $(k'^{6,l}\ k'^{6,r})$ of 16-bit guesses for the left and right halves of the last key row k'^6 that leads to 8 red byte matchings for all the 8 $\{\alpha^{(')}, \beta^{(')}\}$ pairs
    if $(k'^{6,l}\ k'^{6,r})$ passes this test, then the subkey row candidates k'^0 and k'^6 are retained

- rough heuristic reasoning for determining N and rough complexity assessment
  - right guess on (k'^0,k'^6): about N/2.2^{-8} $\{\alpha, \beta\}$ pairs of S pass the test
  - wrong guess on (k'^0,k'^6): about N/2.2^{--64} $\{\alpha, \beta\}$ pairs of S pass the test

  → $N \approx 2^{10}$ suffices in order for the right (k'^0,k'^6) to be retained with overwhelming proba and only a small fraction of false alarms to survive.

→ resulting complexity (example of trade-off): $T \approx 8 \times N \times 2^{32} \times 2^{17} \approx 2^{62}$, $D \approx 2^{42}$, $M \approx 2^{32}$

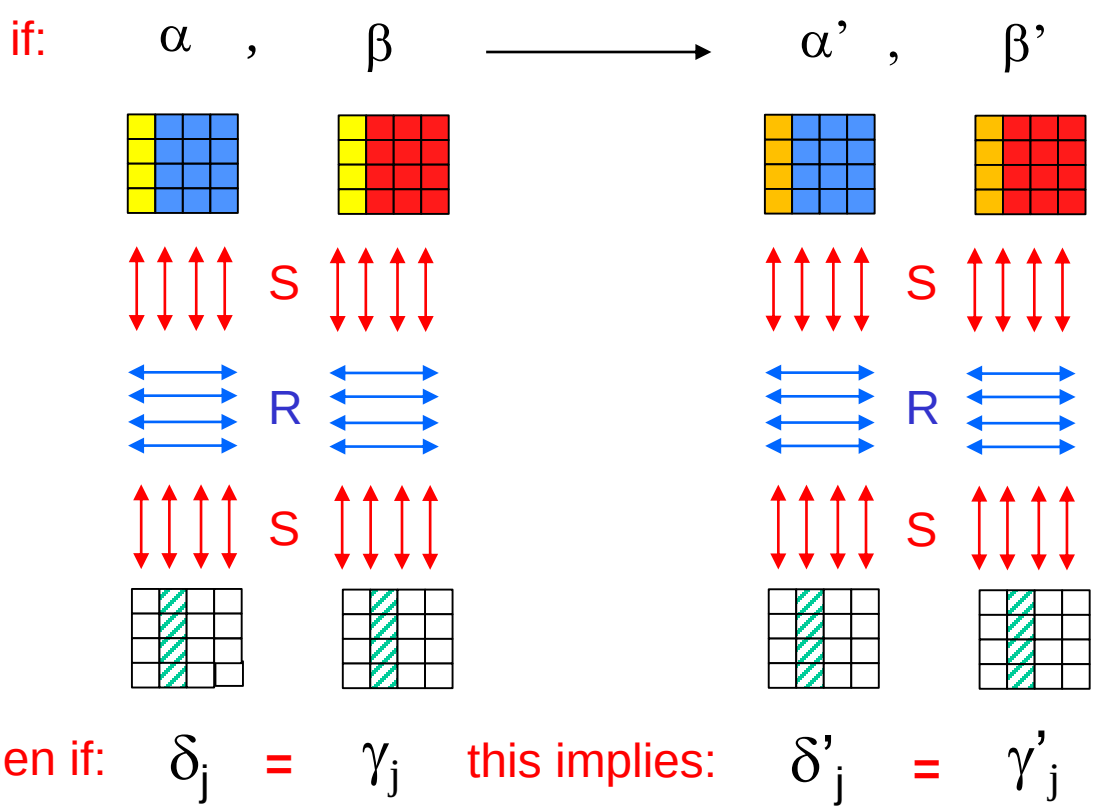
[Note: we will not detail how to derive candidates for the other rows of K'^6, how to test a whole candidate for K'^6, etc.]

Comparison with selected families of key rec. attacks against AES-128 reduced to 6 or 7 rounds

Family	Refs.	Rounds	Time	Data	Memory
Integral	[DKR97] [FKL+00] [DGKL+24]	6 6 6	2^{72} E 2^{45} E 2^{40} E	2^{33} CP 2^{35} CP 2^{33} CP	2^{32} 2^{32} 2^{32}
Boomerang	[B04] [BLU [BDKL+24]	6 6 6	2^{71} E 2^{61} E 2^{61} E	2^{71} CP/CC 2^{59} CP/CC 2^{57} CP/CC	2^{33} 2^{59} 2^{33}
Exchange*	[BDK+20] [our attack]	6 6	2^{80} E 2^{62} E	2^{26} CP 2^{42} CP	2^{28} 2^{32}
Zero-Difference	[BR22]	7	2^{110} E	2^{110} CP	2^{110}
Impossible Difff.	[BLNS18] [LP21]	7 7	2^{113} E 2^{111} E	2^{105} CP 2^{105} CP	2^{74} 2^{72}
Improved MiTM	[DKS10] [DFJ13]	7 7	2^{116} E 2^{99} E	2^{116} CP 2^{97} CP	2^{116} 2^{98}

* some related families are missing, e.g. mixture differential attacks [G17, BDK+18]

❸ Untwisted view of the 4-round distinguisher of [GM00]

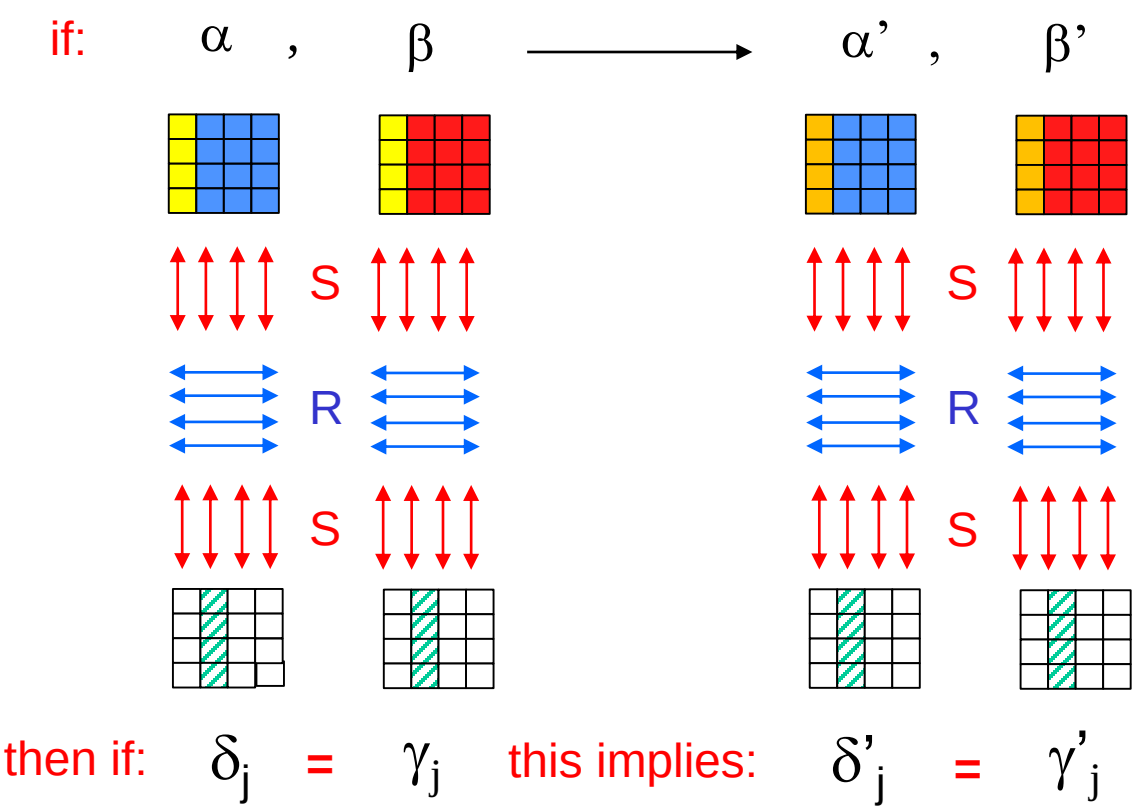


obtained by replacing col. i of α and β (represented in yellow) by any other col. i value (represented in orange)

in other words, the *4-byte to 4-byte partial mappings* $\alpha'_i \mapsto \delta'_j$ collide.

- Proof: replacing the yellow columns of α and β (that are assumed to be equal) does not affect the output difference Δ of the first S mapping. This implies that $\gamma_j = \delta_j$ and $\gamma'_j = \delta'_j$ are equivalent since they both occur iff the j-th column of $R_l(\Delta)$ is 0.

Untwisted view of the 4-round distinguisher of [GM00]

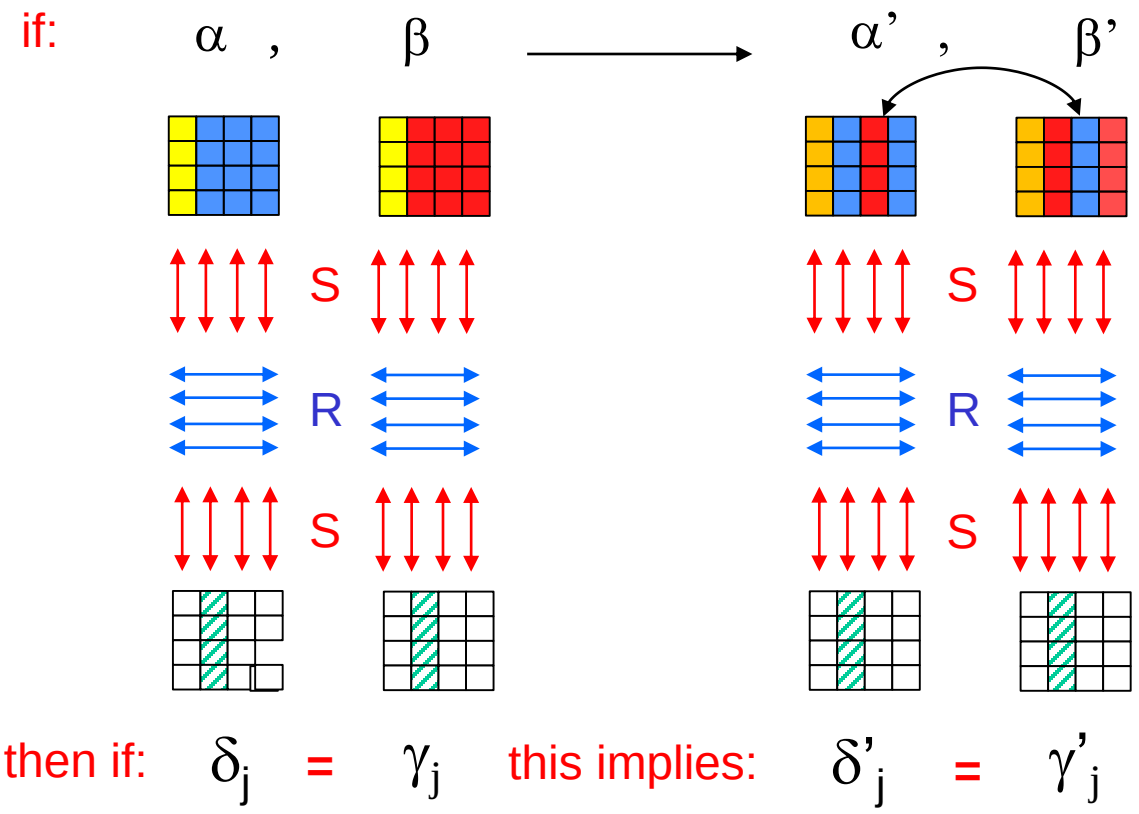


obtained by replacing col. i of α and β (represented in yellow) by any other col. i value (represented in orange)

in other words, the *4-byte to 4-byte partial mappings* $\alpha'_i \mapsto \delta'_j$ collide.

- Note: [GM00] observed that there are relatively few possible *1-byte to 1-byte partial mappings* $\alpha'_{i,k} \mapsto \delta'_{j,l}$ induced by 4 rounds. While it leveraged resulting collisions among such mappings to attack 7 AES-128 rounds [with complexity nearly 2^{128}], subsequent AES attacks from the same family [DS08, DKS10, DFJ13] leveraged an enumeration of such mappings.

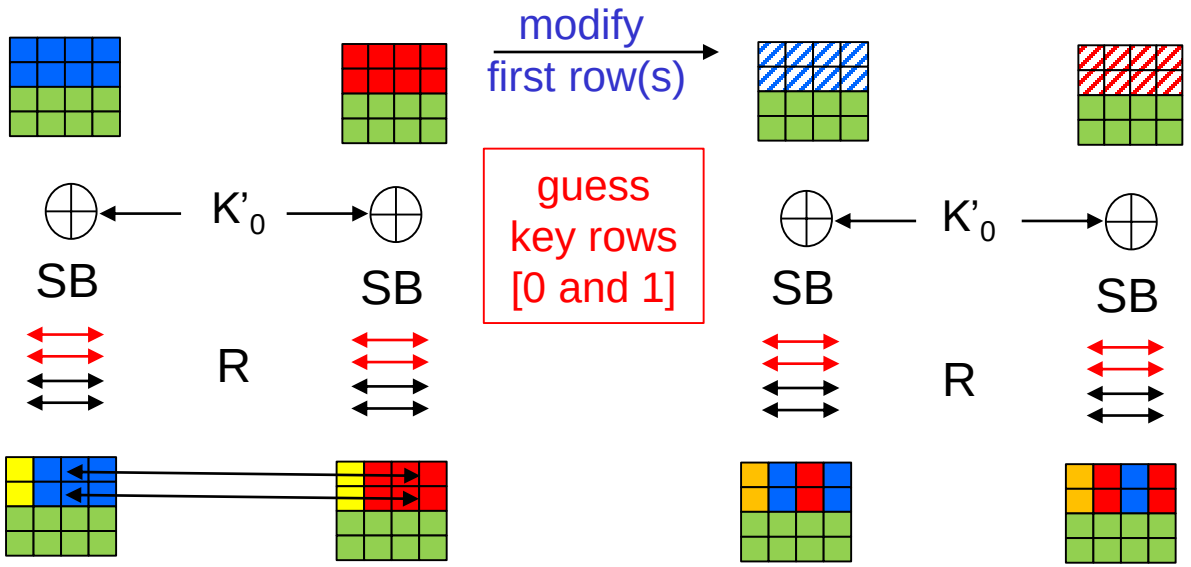
Merging the exchange and [GM00] distinguishers



tweaked (α', β') obtained by:
- replacing the value of (an) equal (yellow) column(s) of α and β
- **and/or** exchanging pairs of distinct (blue and red) columns

Remark: a strong connection appears to exist between the behaviour of the $\{\alpha, \beta, \alpha', \beta'\}$ quartets considered in the merged distinguisher above and the quartets « with related differences » considered by Bardeh and Rijmen in their 7-round zero-difference attack on AES-128 [BR22].

Building « tweaked pairs » after ~1 external round



→The control after 1 round given by guessing the 2 first rows of K'^0 allows to produce structures of [2^{16} x 5] plaintext pairs that result in a column collision at the output of round 5.

Questions: can the above merged 4-round distinguisher be leveraged differently from [GM00] and [BR22] to mount a 7-round attack on AES-128 with1 top external round and 2 bottom external rounds?
Does the large number of equalities above allow to develop a linearisation or polynomial solving technique for recovering subkeys parts of the 2 lasts rounds?