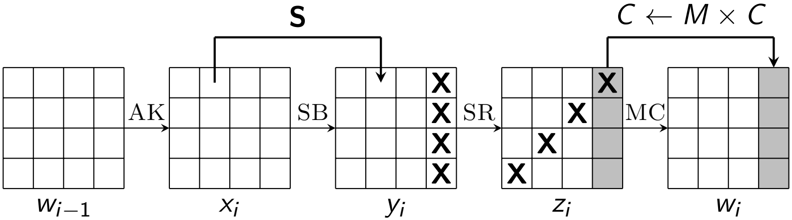
Alternative Key Schedules for the AES

Christina Boura[1], **Patrick Derbez**[2], Margot Funk[1]

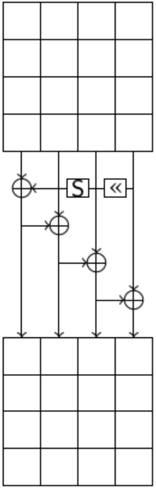
[1] Université Paris-Saclay, UVSQ, CNRS, Laboratoire de mathématiques de Versailles
[2] Univ Rennes, Inria, CNRS, IRISA

The Advanced Standard Encryption

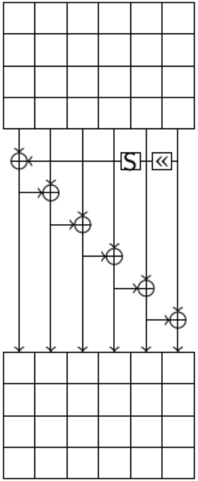


- **Standardized** in 2001 for 3 key lengths: 128, 192 and 256 bits
- Block size of 128 bits: 4×4 matrix of bytes
- An AES round applies $\mathrm{MC} \circ \mathrm{SR} \circ \mathrm{SB} \circ \mathrm{AK}$ to the state
- No MixColumns in the last round

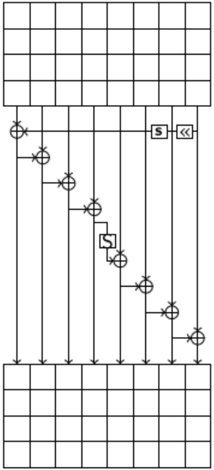
AES Key Schedules



(a) AES-128



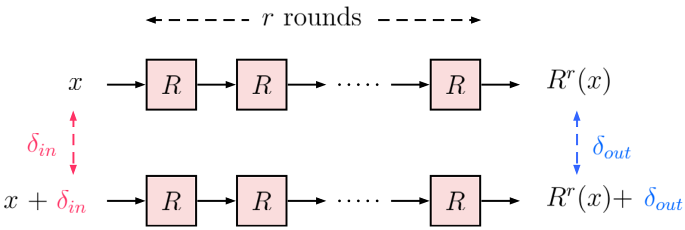
(b) AES-192



(c) AES-256

Differential cryptanalysis

- Cryptanalysis technique introduced by Biham and Shamir in **1990**.
- Based on the existence of a high-probability **differential** $(\delta_{in}, \delta_{out})$.



- If the probability of $(\delta_{in}, \delta_{out})$ is (much) higher than 2^{-n}, where n is the block size, then we have a differential distinguisher.

AES differential trails

$\#$ active S-boxes, max DP of the AES S-box $= 2^{-6}$

$\hookrightarrow$ bound on the differential probability

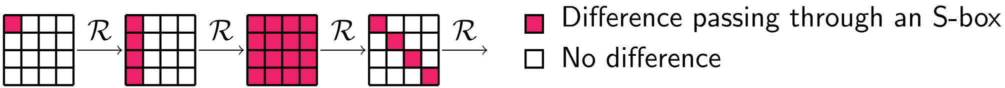


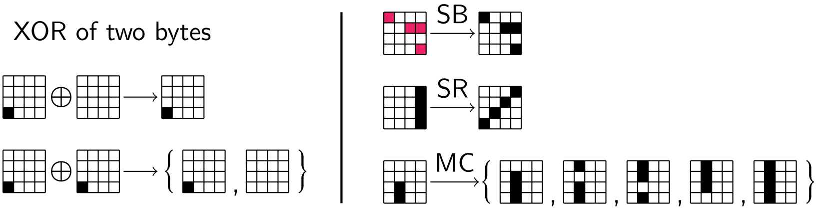
Figure: 4-round truncated differential trail of AES with 25 active S-boxes: $\boldsymbol{p \leq 2^{-25 \times 6}}$

Single-key model VS Related-key model

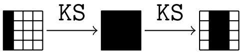
- **Single-key**: **simple** and **powerful** security proofs.
- **Related-key**: much **weaker**.
  Related-key attacks on the full AES-192 and AES-256, Biryukov et al., 2009

Modeling the AES truncated trails

Basic propagation rules ...



... do not necessarily lead to valid truncated trails.

Ex: 

is not instantiable.

Changing the key schedule for a permutation

Using a **permutation** as key schedule:

- Efficient in both hardware and software
- Easier to analyze
- Better security with simpler design?

Previous results:

- Khoo et al. (FSE 2018): permutation for AES-128
- Derbez et al. (SAC 2018): better permutations for AES-128 + bounds

Changing the key schedule for a permutation

Using a **permutation** as key schedule:

- Efficient in both hardware and software
- Easier to analyze
- Better security with simpler design?

Previous results:

- Khoo et al. (FSE 2018): permutation for AES-128
  - easy to generate similar ones at random
- Derbez et al. (SAC 2018): better permutations for AES-128 + bounds

Changing the key schedule for a permutation

Using a **permutation** as key schedule:

- Efficient in both hardware and software
- Easier to analyze
- Better security with simpler design?

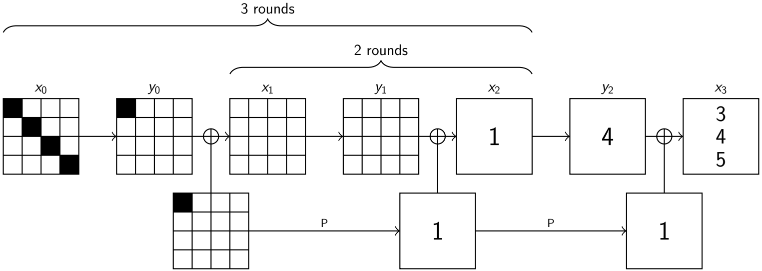
Previous results:

- Khoo et al. (FSE 2018): permutation for AES-128
  - easy to generate similar ones at random
- Derbez et al. (SAC 2018): better permutations for AES-128 + bounds
  - Issue with the model: permutations are much **worst** than expected!

Generic Bounds on 2, 3 and 4 rounds

Formally proven [DFJL18]

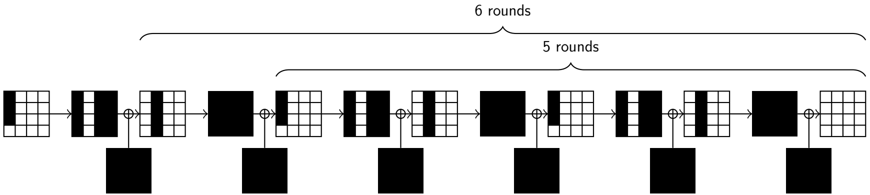
The optimal bounds for 2, 3 and 4 rounds are respectively 1, 5 and 10 active S-boxes, even when considering induced equations.



Generic Bounds on 5, 6 and 7 rounds

Formally proven [DFJL18]

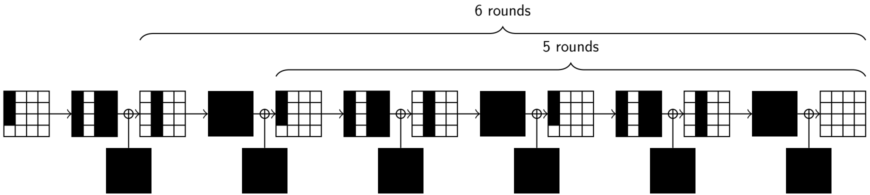
The optimal bounds for 5, 6 and 7 rounds are respectively 14, 18 and 21 active S-boxes, **without considering equations**.



Generic Bounds on 5, 6 and 7 rounds

Formally proven [DFJL18]

The optimal bounds for 5, 6 and 7 rounds are respectively 14, 18 and 21 active S-boxes, **without considering equations**.



What are the bounds when considering equations?

A Definition

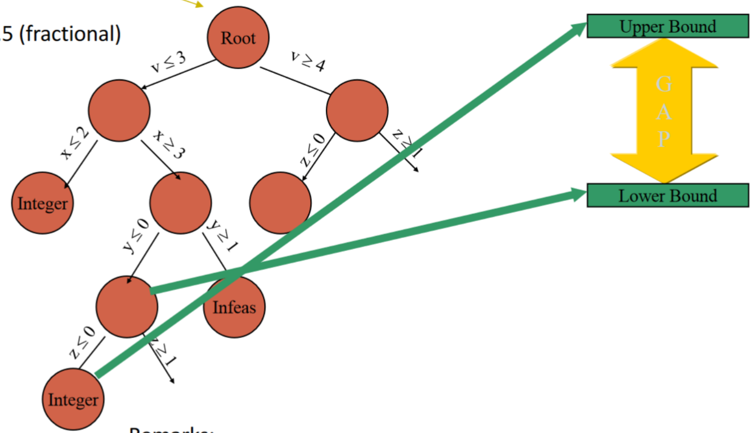
- A **mixed-integer program** (MIP) is an optimization problem of the form:

$$\boxed{\begin{aligned} &\textit{Minimize} && c^T x \\ &\textit{Subject to} && Ax = b \\ & && l \leq x \leq u \\ &\text{some or all } x_j \text{ integer} \end{aligned}}$$

MIP Solution Framework

Solve LP relaxation:

v=3.5 (fractional)

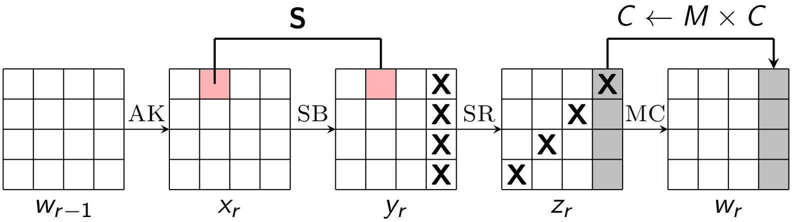


Remarks:

(1) GAP = 0 $\Rightarrow$ Proof of optimality

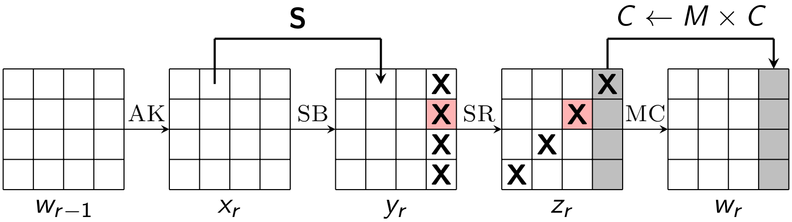
(2) In practice: Often good enough to have good Solution

Application to AES



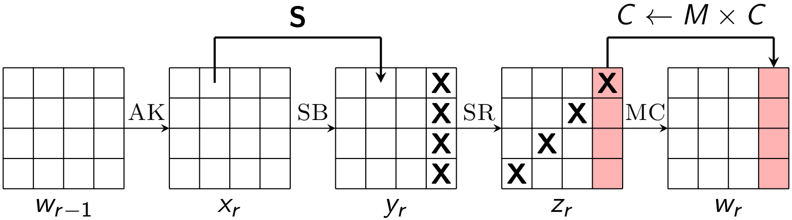
- $x_r[i] = y_r[i]$

Application to AES



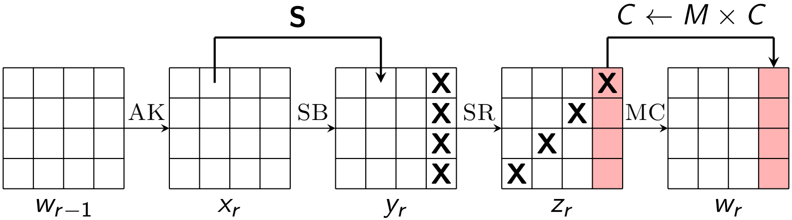
- $x_r[i] = y_r[i]$, $y_r[i] = z_r[\mathtt{SR}[i]]$

Application to AES



- $x_r[i] = y_r[i]$, $y_r[i] = z_r[\mathtt{SR}[i]]$
- $\sum_{i \in C} z_r[i] + w_r[i] = 0$ or ≥ 5

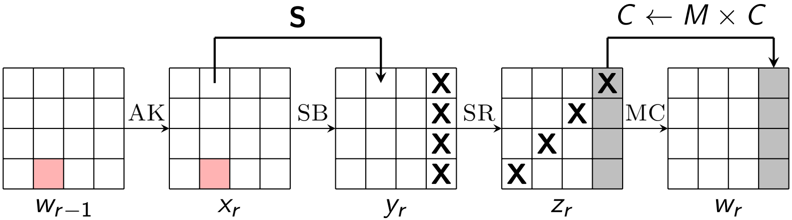
Application to AES



- $x_r[i] = y_r[i]$, $y_r[i] = z_r[\mathbb{SR}[i]]$
- $\sum_{i \in C} z_r[i] + w_r[i] = 0$ or ≥ 5
- Introduce an extra binary variable e

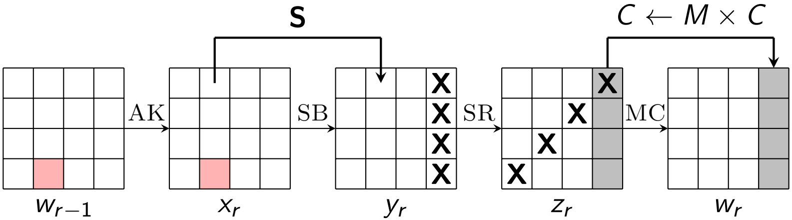
$$\sum_{i \in C} z_r[i] + w_r[i] \geq 5e \text{ and } \sum_{i \in C} z_r[i] + w_r[i] \leq 8e$$

Application to AES



- **No difference in key:** $w_{r-1}[i] = x_r[i]$

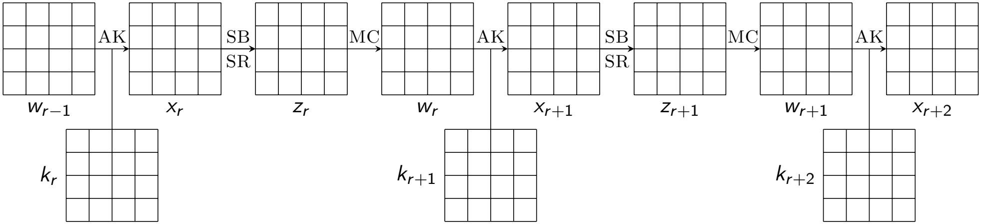
Application to AES



- **No difference in key:** $w_{r-1}[i] = x_r[i]$
- **Difference in key:** $w_{r-1}[i] + k_r[i] + x_r[i] \neq 1$

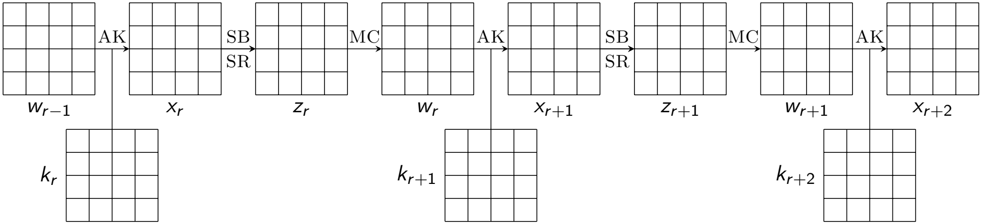
$$\begin{cases} 1 - w_{r-1}[i] + k_r[i] + x_r[i] & \geq & 1 \\ w_{r-1}[i] + 1 - k_r[i] + x_r[i] & \geq & 1 \\ w_{r-1}[i] + k_r[i] + 1 - x_r[i] & \geq & 1 \end{cases}$$

Correctness of the model



Is this model correct?

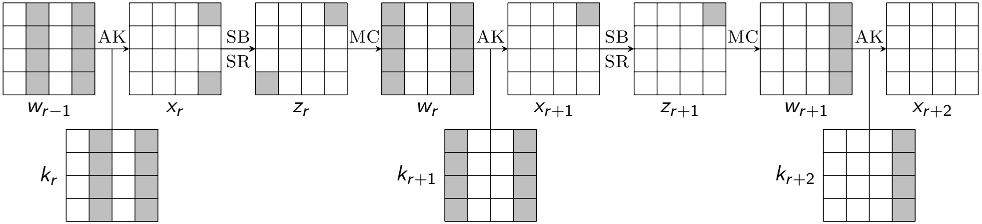
Correctness of the model



Is this model correct?

- Yes, if there is no difference in the key

Correctness of the model



Is this model correct?

- Yes, if there is no difference in the key
- No otherwise!

$w_r \oplus w_{r+1} = \texttt{MC}(z_r \oplus z_{r+1})$ does not satisfy MDS property!

Linear Algebra

How to solve this issue?

- Compute all linear combinations of the original system and add corresponding constraints?
  - too many constraints $\rightarrow$ model very slow to solve

Linear Algebra

How to solve this issue?

- Compute all linear combinations of the original system and add corresponding constraints?
  - too many constraints $\rightarrow$ model very slow to solve
- Use a callback: check validity of solutions a posteriori
  - Depend on the problem

How to solve this issue?

- Compute all linear combinations of the original system and add corresponding constraints?
  - too many constraints $\rightarrow$ model very slow to solve
- Use a callback: check validity of solutions a posteriori
  - Depend on the problem
- Better solutions?

Double MILP model

Goal: find a permutation ensuring ***b* active S-boxes**.

Generate P

Evaluate P

Ensure that P is a permutation.

Double MILP model

Goal: find a permutation ensuring ***b* active S-boxes**.

Generate P $\xrightarrow{\quad P \quad}$ Evaluate P

Ensure that P is a permutation.

Double MILP model

Université de Rennes

Goal: find a permutation ensuring ***b*** **active S-boxes**.

Generate P $\xrightarrow{P}$ Evaluate P

Ensure that P is a permutation.

No solution with less than b active S-boxes

Double MILP model

Goal: find a permutation ensuring ***b* active S-boxes**.

Generate P $\xrightarrow{P}$ Evaluate P

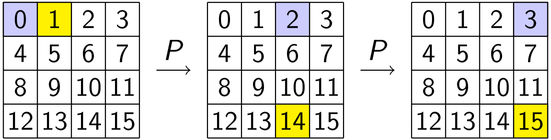
Ensure that P is a permutation.

Remove the bad subkeys pattern $(K_1, \ldots, K_n)$.

A trail with less than b active S-boxes.

Removing a bad subkeys pattern

- 1^{rst} idea: forbid the exact trail.



At most 3 of these equalities should be true.

$$P(0) = 2 \qquad P(1) = 14$$

$$P(2) = 3 \qquad P(14) = 15$$

Removing a bad subkeys pattern

- 2nd idea: forbid the subkeys pattern.

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

$\xrightarrow{P}$

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

$\xrightarrow{P}$

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

At most 1 of these equalities should be true.

$$P(\{0,1\}) = \{2,14\}$$

$$P(\{2,14\}) = \{3,15\}$$

- Possible if and only if the differences can all be equal!

Results on AES-128

Université de Rennes

Rounds	3	4	5	6	7
`AES-128`	5	12	17	21	27
`Khoo et al.`	5	10	14	19	23
P_{128}	5	10	14	20	22
	5	9	15	20	23

- Not able to strictly improve Khoo et al. bounds
- Permutations seem weaker than original key-schedule ...
- ... but all active S-boxes are located in the internal states

AES-192 and AES-256

These versions are much weaker against differential cryptanalysis

- Boomerang attacks on the **full version** against both of them!

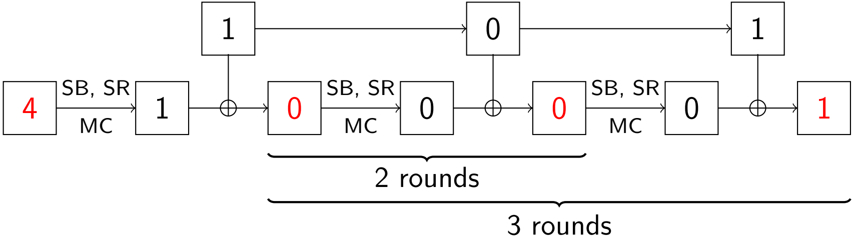
AES-192 and AES-256

These versions are much weaker against differential cryptanalysis

- Boomerang attacks on the **full version** against both of them!

AES with permutation-based key-schedule

The optimal bounds for 2, 3 and 4 rounds for `AES-192` (resp. `AES-256`) are 0, 1 (resp. 2) and 5 active S-boxes.



Results

Rounds	3	4	5	6	7	8	9	10
`AES-192`	1	4	5	10	14	18	24	29
P_{192}	1	5	10	13	17	22	25	28
`AES-256`	1	3	3	5	5	10	15	16
P_{256}	1	2	5	10	14	16	22	26

- Improve the resistance against differential cryptanalysis
- Secure against boomerang attacks!

Conclusion

- The key schedule is one of the less understood components in block ciphers.
- Simple key-schedules are easier to study and can provide good resistance against differential cryptanalysis.

Open problems:

- How to reduce the search space?
- Optimize against other types of attacks: meet-in-the-middle attacks, key-recoveries, ...

Conclusion

- The key schedule is one of the less understood components in block ciphers.
- Simple key-schedules are easier to study and can provide good resistance against differential cryptanalysis.

Open problems:

- How to reduce the search space?
- Optimize against other types of attacks: meet-in-the-middle attacks, key-recoveries, ...

Thank you for your attention!